A PLATFORM FOR THE 21st CENTURY

U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

ColdFusionJournal.com

December, 1999 Volume: 1 Issue: 6

SYS-CON PUBLICATIONS

AbleCommerce

The flexible solution for building electronic shopping sites

Complete Site Building and Administration
from your browser.

Build "your_store.com" with secure Merchant Credit Card Processing. Maintain inventory, add discounts and specials to keep your customers coming back. Increase sales with cross selling and membership pricing. For accurate shipping use AbleShipper™ UPS auto-calculator with the flexible AbleCommerce™ shipping tables. Fulfill document and software purchases through electronic delivery. Check orders online, generate automated email and fax notification.

Enterprise Scalability
for stores of any size.

Supports Microsoft Access and SQL. Add store and server license packs to create an unlimited number of stores. Two security levels for system and merchant administration allow ISP's to empower renters with online store design and maintenance abilities.

You can also integrate and customize AbleCommerce™ with open source code.

Easy Design Interface
for eye catching sites.

Upload graphics directly to your server. Use prebuilt store templates for quick design. Customize your web pages and styles dynamically. Add you own HTML, Java, Flash or any other script to create a unique look. Add comment forms and surveys with automated responses to collect data from your visitors. Track visitor click throughs to determine the strengths of your site design.

> ***AbleCommerce is a perfect example of ColdFusion's strength and flexibility.***
>
> -Ben Forta, Allaire Corp.

> ***The system is easy to use and very flexible from both a customer and administrator standpoint.***
>
> -Ron Derby, Casio, Inc.

CLIENT LIST

A few of our -shopping customers

- 24 Hour Fitness
- Bushnell
- Johns Hopkins University
- BOC Gases
- Key Tronic
- Casio
- Citgo
- GSA
- Lionel Trains
- Lucky Brand Jeans
- Compaq
- Mercury Marine
- Dairy Farmers of America
- Mitsubishi
- FAA
- Nike
- Garth Brooks
- Miss Universe/USA/Teen USA
- NorWest Financial
- George Strait
- PGE
- Edwin Watts Golf Shops
- Quest
- Hasbro
- Siemens
- U.S. Department of Treasury
- International Paper
- United Nations
- Jimmy Buffett
- Jeff Gordon
- Verio

To see more, visit www.topsites.com

TRY IT OUT FREE!

Download the 30-Day Evaluation

www.ablecommerce.com

AbleCommerce, 11700 NE 95th Street, Suite 100, Vancouver, Washington, 98682, (360) 253-4142

AbleCommerce is a trademark of Able Solutions Corporation. ColdFusion is a trademark of Allaire Corp. Company or product names referenced may be the trademark or registered trademark of their respective companies. © Able Solutions Corporation, All Rights Reserved.

Computerjobs.com

computerjobs.com

Allaire

www.conceptware.com

ColdFusion Developer's Journal

EDITORIAL ADVISORY BOARD
STEVEN D. DRUCKER, JIM ESTEN, BEN FORTA,
STEVE NELSON, RICHARD SCHULZE, PAUL UNDERWOOD

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

WRITERS IN THIS ISSUE
JEREMY ALLAIRE, JERRY BRADENBAUGH,
RAYMOND CAMDEN, ROBERT DIAMOND, BEN FORTA,
ANTHONY KRINSKY, MICHAEL MACDONALD,
STUART NEWMAN, RICHARD SCHULZE, DAVID SCHWARTZ

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS,
PLEASE SEND YOUR LETTERS TO
SUBSCRIPTION DEPARTMENT.

SUBSCRIPTION HOTLINE 800 513-7111
COVER PRICE $8.99/ISSUE
DOMESTIC $49.99/YR. (6 ISSUES)
CANADA/MEXICO $69.99/YR.
OVERSEAS $99.99/YR
BACK ISSUES $12 EACH

PUBLISHER, PRESIDENT AND CEO FUAT A.KIRCAALI
VICE PRESIDENT, PRODUCTION JIM MORGAN
VICE PRESIDENT, MARKETING CARMEN GONZALEZ
CHIEF FINANCIAL OFFICER IGNACIO ARELLANO
ACCOUNTING MANAGER ELI HOROWITZ
CIRCULATION MANAGER MARY ANN MCBRIDE
ADVERTISING ACCOUNT MANAGER ROBYN FORMA
ADVERTISING ASSISTANT MEGAN RING
GRAPHIC DESIGNER ROBIN GROVES
GRAPHIC DESIGNER ALEX BOTERO
GRAPHIC DESIGN INTERN AARATHI VENKATARAMAN
WEBMASTER BRUNO Y. DECAUDIN
CUSTOMER SERVICE SIAN O'GORMAN
CUSTOMER SERVICE ANN MARIE MILILLO
ONLINE CUSTOMER SERVICE AMANDA MOSKOWITZ

EDITORIAL OFFICES
SYS-CON PUBLICATIONS, INC. 39 E. CENTRAL AVE.,
PEARL RIVER, NY 10965
TELEPHONE: 914 735-7300 **FAX:** 914 735-6547

COLDFUSION DEVELOPER'S JOURNAL (ISSN #1523-9101)
IS PUBLISHED BIMONTHLY (6 TIMES A YEAR)
FOR $49.99 BY SYS-CON PUBLICATIONS, INC.,
39 E. CENTRAL AVE., PEARL RIVER, NY 10965-2306.

POSTMASTER
SEND ADDRESS CHANGES TO:
COLDFUSION DEVELOPER'S JOURNAL
SYS-CON PUBLICATIONS, INC.
39 E. CENTRAL AVE., PEARL RIVER, NY 10965-2306

© COPYRIGHT
COPYRIGHT © 1999 BY SYS-CON PUBLICATIONS, INC.
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE
REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS,
ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY OR ANY
INFORMATION STORAGE AND RETRIEVAL SYSTEM,
WITHOUT WRITTEN PERMISSION.

FOR PROMOTIONAL REPRINTS, CONTACT REPRINT COORDINATOR.

SYS-CON PUBLICATIONS, INC., RESERVES THE RIGHT TO REVISE,
REPUBLISH AND AUTHORIZE ITS READERS TO USE
THE ARTICLES SUBMITTED FOR PUBLICATION.

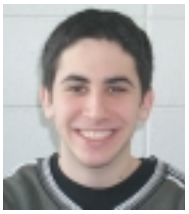
WORLDWIDE DISTRIBUTION
BY **CURTIS CIRCULATION COMPANY** 739 RIVER ROAD,
NEW MILFORD NJ 07646-3048 PHONE: 201 634-7400

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

ALL BRAND AND PRODUCT NAMES USED ON THESE PAGES
ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS
OF THEIR RESPECTIVE COMPANIES.

SYS-CON PUBLICATIONS

So Much to Say...



BY **ROBERT DIAMOND**

When I first began my outline for this month's editorial, I realized I couldn't possibly fit everything in. With topics ranging from the excitement of the Allaire Conference, whose phenomenal success and fast sellout is a testament to the growing success of ColdFusion, to the amazing articles in this month's issue, there was a lot to cover. However, I'll do my best to touch on each one.

ColdFusion Enters Prime Time

The recent Allaire Developer Conference generated quite a buzz on the current state and the bright future of the ColdFusion industry. I couldn't attend this year, but I've been getting incredible feedback from other members of the ***CFDJ*** editorial team. Everyone I've spoken with says Allaire did a great job, and they can't wait till next year when it's bound to be an even bigger event at a much larger venue. The workshops were fantastic and one need only look at the wide range of topics covered to appreciate how ColdFusion is most definitely coming into its own. Everyone had a great time – next year *nothing* will stand in the way of my being there.

The Best Issue Yet!

Now onto the magazine....At ***CFDJ*** we try to have a mix of articles on a variety of industry topics and this issue is definitely indicative of that. Michael MacDonald has written a great article on creating dynamic graphics for the Web; I'm sure many of us could use a better method for this process. Stuart Newman writes about banner rotation and tracking, which most sites are utilizing these days since advertising is paying for the path of the Web – and paying many of our salaries too! David Schwartz has a walk-through on adding a dynamic table viewer and query-by-example engine to CF applications. Jerry Bradenbaugh demonstrates the creation of a Web-based e-mail application that uses only one CF template – not an easy task. There's an article on Spectra by Anthony Krinsky, and much more to make this issue of ***CFDJ*** the best yet – enjoy!

Good News from the Publisher of *CFDJ*

SYS-CON Publications, publisher of ***ColdFusion Developer's Journal,*** has just been named the fastest-growing, privately held publishing company in America by *Inc.* 500. **SYS-CON Publications** is one of 79 companies founded in 1994 that made it to this year's list, based on its five-year growth through 1998. It was the only company in the publishing industry to make it to this year's list. Congratulations to **SYS-CON**!

We Want to Hear from You...

Each month ***CFDJ*** strives to serve our readers' needs better. To that end, we want your feedback – tell us what you want to see more of and we'll cover it. Do you want more beginner materials, more advanced articles, more code, less code?...Whatever you want, let us know and we'll provide it! Till next month – happy coding!

ABOUT THE AUTHOR
Robert Diamond is editor-in-chief of ColdFusion Developer's Journal. He can be reached at rdiamond@sys-con.com.

ROBERT@SYS-CON.COM

Using Syndicated Content in Spectra Part 1

Here's some of the **'heavy lifting'** you'll need to perform to **get syndicated content** into your Spectra object store

BY **ANTHONY KRINSKY**

Managing and displaying content stored in the Spectra object store will be a major feature of any Spectra project. Content will come from a variety of places: self-service applications, system-generated messages and objects, and externally provided articles and data from a variety of content providers. The latter is called "syndicated content." This article demonstrates how one might automate the insertion of syndicated content from an outside news service.

Getting Started

There are two methods of getting syndicated content in Allaire Spectra: the hard way…and the easy way. The easy way is to transfer content over HTTP from one Spectra machine to another using the Syndication features of the COAPI (Content Object API) (see Figure 1, Scenario 3). This content syndication process gets a terrific write-up in chapter 14, "Syndication and Remote Access," of the *Programming with Allaire Spectra* manual.

Alternatively, content vendors wouldn't need their own Spectra servers, but would simply package the data in an XML format, WDDX, that Spectra understands (Figure 1, Scenario 2). Allaire will soon begin to promote WDDX, as implemented by the COAPI, for this purpose. Spectra users and content providers can also use XSL transformation engines to normalize XML data before or after syndication. Allaire may soon bundle Granularity Information Architecture's GIAXT XSL transformation engine (www.granularity.com) to provide such translation services.

In the meantime we'll need to process newsfeeds ourselves and programmatically funnel them into the object store through the COAPI

(Figure 1, Scenario 1). It's not elegant, but it's easy to implement and works quite well.

If you want real-time news from a variety of sources and you want it fast, one of the best places to start is Screaming Media (www.screamingmedia.net). Screaming Media is a nice solution because they aggregate content from hundreds of traditional and nontraditional sources, provide data in a consistent format, allow you to pay only for what you use and don't require impression-based royalties. For $2,500 up-front and $500/month (and up), Screaming Media will send piping-hot, filtered news to a proprietary Java component, Siteware, that sits on your server. Reuters, Wavo and iSyndicate provide a similar service and each has its own proprietary or third-party syndication components.

Installing the Siteware component first requires installing the Java 1.1.8 Run-time Environment (JRE) from http://java.sun.com. You then unzip some files and add a "siteware.bat" shortcut (on NT) to the startup (C:\WINNT\Profiles\All Users\Start Menu\Programs\Startup) directory. This file includes the JDBC:OBDC driver that Siteware uses to communicate with MS SQL Server. Make sure you install this in the default directory.

When you order news from Screaming Media, you specify filters describing the type of news you want, industry segments, keywords and so forth. Each filter becomes a feed that has its own configuration files specifying field delimiter characters, escape character policies, headers and footers. Siteware "wakes up" every 15 minutes and pulls your feeds off the Screaming Media servers. It then processes the feeds in one of four ways as Hclient, DBClient, fClient or pClient.

If you're running MS SQL server or have access to a JDBC driver for your favorite database, using the DBClient database insertion method is a no-brainer (tell Screaming Media this is how you want Siteware set up). The Hclient method is quite interesting since it will write content into templates using variable substitution. Once we build our Screaming Media News object type, we could create a sample object and use it as a prototype template for use with this method. Since databases are more flexible than flat files, I've chosen to use DBClient for my own installation. fClient and pClient can also write out delimited text files but you need to parse them and process them yourself – no fun.

To store Screaming Media content temporarily, you'll need to create a table in your SQL database with the columns listed in Table 1.

I've set up a DSN in the ODBC administrator called "SyndicatedContent" that points to my "SyndicatedContent" database and defaults to the "ScreamingContent" table.

If you're getting sports scores from Screaming Media, you'll need to add additional fields to your ScreamingContent table or create a new table ("ScreamingSportsContent," perhaps!). For more information on sports scores, contact Screaming Media directly.

Table 1: Format for a Non-Sports content syndication table (MS SQL Server)

Column Name	Datatype	Length	Precision	Scale	Allow Nulls
title	varchar	256	0	0	Y
date	datetime	8	0	0	Y
time	int	4	10	0	Y
ctprovider	varchar	256	0	0	Y
copyright	varchar	256	0	0	Y
body	text	16	0	0	Y
category	varchar	256	0	0	Y
id	int	4	10	0	N

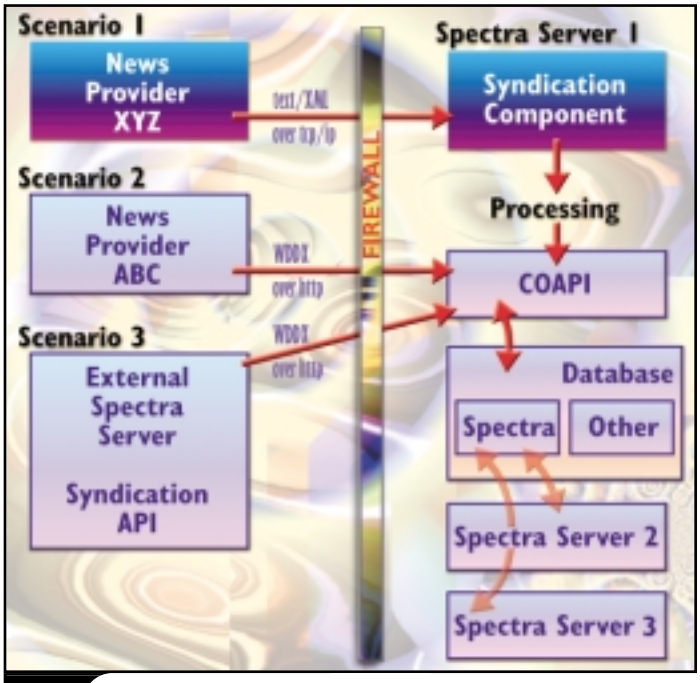


FIGURE 1: Allaire Spectra content import and Syndication scenarios

Table 2: ScreamingNews Property Definitions

Property Alias	Property Name	Default	Required	Searchable	Indexed
title	ScreamingMedia_title	<empty>	no	yes	no
date	ScreamingMedia_date	<empty>	no	no	no
time	ScreamingMedia_time	<empty>	no	no	no
ctprovider	ScreamingMedia_ctprovider	<empty>	no	yes	no
copyright	ScreamingMedia_copyright	<empty>	no	no	no
body	ScreamingMedia_body	<empty>	no	yes	no
category	ScreamingMedia_category	<empty>	no	yes	yes
bCleaned	ScreamingMedia_cleaned	no	no	no	yes
image	IMAGE (system property)		no	no	no
image_ vplacement	ScreamingMedia_imgvplace (top|bottom)	top	no	no	no
buseimage	ScreamingMedia_buseimg	no	no	no	yes

Table 3: Method Handlers

Method	Purpose	Handler File
listing	One-line headline for news article.	listing.cfm
detail	Headline, date, and a few lines of text – provides a *peek at* the article.	detail.cfm
full	Full text of article.	full.cfm
syndicate_import	Imports content from the syndication database. Marks content for deletion after 30 days.	syndicate_import.cfm
syndicate_push	Pushes syndicated content to another Spectra server.	syndicate_push.cfm
syndicate_pull	Pulls syndicated content from another Spectra server.	syndicate_pull.cfm

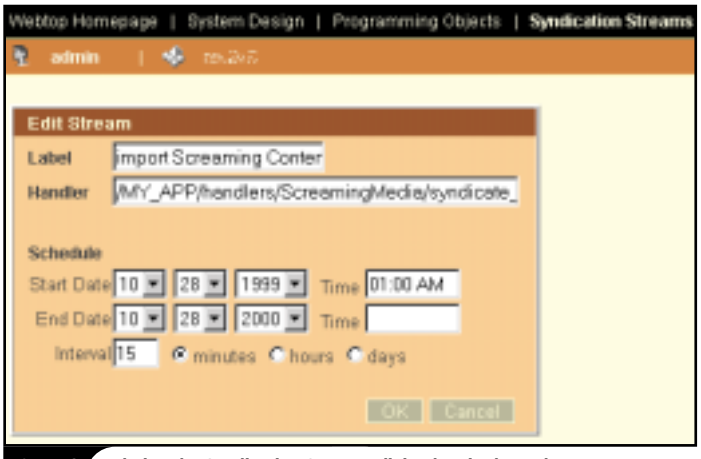


FIGURE 2: Updating the Syndication Streams dialog box in the Webtop

Screaming Media's technical staff will edit Siteware's configuration files to match the DSN and table names you've created. Extract the *.zip file they send you and run siteware.bat (located in the IMDSJServer/bin directory) from the command prompt. When you see two "SUCCESS" prompts and a blinking cursor beneath, Siteware has initialized successfully and has begun dumping content into your database. Look into your database table and you'll see the stories that meet your filter criteria.

The next step is moving the content from the syndication table into Spectra. The first thing we need to do is create a ScreamingNews object type that defines how the data will be stored in the Spectra database. This requires some "data modeling" in the Webtop. After logging in under the admin account, go to System Design > Site Object Designer > Property Definitions and create the properties in Table 2.

I've added the bCleaned property to those in the database as a flag to denote whether the default text has been reviewed to clean up any formatting inconsistencies from the feed. I also added properties that allow me to bind photographs to ScreamingNews objects. You can license photos from several places on the Internet including Reuters and the Associated Press. Spectra ships with a "core" system object type IMAGE (70E7DFB6-A67E-11D2-B3AC00C04FA35A23) that you can define as an embedded property in any object type. We'll embed an IMAGE property into the news article as well as provide a vertical alignment control variable and useimage flag.

Once the properties are created, you can actually create the object type itself (you must follow this sequence). Go to the Type Designer, create a new object type called "ScreamingNews" and map in the property definitions as indicated. Then add the method handlers given in Table 3.

Note that we didn't include "create", "edit" and "delete" methods as they are implicit and use the default handlers. If you write any of them yourself, write the edit handler – the default is quite ugly. There's no special "create" or "delete" functionality required, so there's no need to declare these methods and write your own handlers. Beware of the embedded image object – implementing embedded objects in handlers is an "advanced" skill.

This is public news, so there aren't any permissions on this object type. Applying permissions in Spectra slow down the application considerably – less so for object types than for objects themselves. Caching permissions in the CF administrator is virtually essential.

The last step in getting the data into the object store is to build the handler, syndicate_import.cfm, that manages the process (see Listing 1). This handler is nothing more than a simple query against your syndicated content database, and a loop that copies the data into the Spectra object store and then updates the foreign key in the syndicated content database. It then calls a cleanup routine to mark articles older than 30 days as "inactive" and "archived" since it's illegal for you to keep them around. Note that we've created but not coded syndicate_push and syndicate_pull. These method handlers are used at a later time to manage syndication directly between Spectra servers in the organization or with partners (you'll need to discuss licensing with Screaming Media, of course).

You can test the handler in Listing 1 by calling it from a dummy page with the following code:

```
<cfa_contentobject
      datasource="#request.cfa.objectstore.dsn#"
      objectid="Put a test object ID in here"
      method="syndicate_import"
      bsecure="No"
      babortonunauthorizedaccess="No"
      blogging="No"
      busecache="No"
>
```

The syndication stream invoker (which we'll call next) doesn't care that syndicate_import doesn't require an ObjectID; however, CFA_ContentObject does care so we need to provide a dummy ObjectID in our testing code.

Once you see that this handler is working properly, you need to set up a schedule in Webtop to call it every few minutes. In System Design > Programming Objects > Syndication Streams you can easily add an event to run the syndicate_import method every 15 minutes or at whatever polling interval you've established in Siteware (see Figure 2). While Screaming Media does channel its content providers' various feed formats into the fields listed above, providers format their stories differently so it's strongly suggested that you edit each article within Spectra prior to posting it on your site. You can see the results of your hard work in the Webtop object finder (see Figure 3).

Sample handler code and the content object types I created above are available in their entirety on the ***CFDJ*** Web site and may be installed all at once using the Spectra object type package install func-

Intr@Vision

www.galileodev.com

FIGURE 3: Viewing imported content in the Webtop Content Finder

tion. You'll also want to check the Allaire Tag Gallery and/or the Screaming Media Web site for the latest code as Screaming Media, along with other content providers, has become increasingly interested in the Spectra platform as a turnkey content subscriber and publishing solution.

Conclusion and Next Steps

In this article I've described some of the "heavy lifting" you'll need to perform to get syndicated content into your Spectra object store. Since object type definitions and properly coded handlers are reusable, you won't need to start from scratch; use and modify the source code provided here at your discretion.

It won't be long before syndicated content providers are cognizant of the huge installed base of Spectra servers. When that occurs, they may give you the ability to pull content directly into your Spectra servers by providing it over HTTP in the appropriate format. When this happens, you'll download and install an object type definition file from the content provider and set up a syndication stream handler that polls their database every so often for your content. Alternatively, they could push content directly to Spectra through the remote method invoker interface after you provide them with the appropriate username and password, your IP address and any other default values you care about (e.g., don't publish this article until I see it first!). In these scenarios you may use a syndicate_pull or syndicate_push method handler rather than the kludgy but functional syndicate_import routine provided here.

As a premium service, content providers may also map their metadata keywords to ones you provide, actually marking up the content with metadata before sending it over. They might even write Spectra news-viewer applications that use the content object types they provide. What better way to get you started with their content *fast*!

Syndicating graphics is going to be a bit more difficult, but easier with ColdFusion 4.5's new facilities for Base 64 encoding. For now, write or use default handlers that bind your downloaded graphics separately to articles you've syndicated and are ready to publish.

In the next article we'll explore how the Spectra syndication API allows transparent syndication of content and applications between Spectra servers.

ABOUT THE AUTHOR

Anthony Krinsky is CTO of Memberdrive.com, a provider of highly scalable Java and ColdFusion-based portal solutions (www.memberdrive.com).

akrinsky@memberdrive.com

LISTING 1: Code for the syndicate_import handler

```
<!--- note: improvements to this code will be available from
www.screamingmedia.net -->

<cfa_handler
 object="articles"
 bEnableDesignMode = "false">

<CFPARAM name=attributes.stparams DEFAULT=StructNew()>
<CFPARAM name=attributes.stparams.syndicatedcontentdatasource
DEFAULT="SyndicatedContent">
<CFPARAM name=attributes.stparams.syndicatedcontenttable
DEFAULT="ScreamingContent">
<CFPARAM name=attributes.stparams.lcategories DEFAULT="">
<CFPARAM name=attributes.stparams.bActive DEFAULT="1">

<!--- pull syndicated content from temporary tables -->

<CFIF attributes.stparams.lcategories is not "">
 <CFSET attributes.stparams.lcategories=Replace
  (attributes.stparams.lcategories,",","'",'"')>
 <CFSET attributes.stparams.lcategories="'#attributes.stparams.
  lcategories#'">
</CFIF>

<cfquery name="qGetNewContent" datasource="#attributes.stparams.
  syndicatedcontentdatasource#" dbtype="ODBC">
SELECT * FROM #attributes.stparams.syndicatedcontenttable#
WHERE ObjectID is NULL
<CFIF ATTRIBUTES.stparams.lcategories is not ""> AND CATEGORY IN
(#PreserveSingleQuotes(attributes.stparams.lcategories)#)</CFIF>
</cfquery>

<!--- create Screaming News objects through COAPI -->

<CFLOOP QUERY="qGetNewContent">

<CFSET ContentID=qGetNewContent.id>

<CFSET stProperties=StructNew()>
<CFSET stProperties["time"]="#trim(qGetNewContent.time)#">
<CFSET stProperties["copyright"]="#trim(qGetNewContent.copyright)#">
<CFSET stProperties["date"]="#trim(qGetNewContent.date)#">
<CFSET stProperties["title"]="#trim(qGetNewContent.title)#">
<CFSET stProperties["ctprovider"]="#trim(qGetNewContent.ctprovider)#">
<CFSET stProperties["body"]="#trim(qGetNewContent.body)#">
<CFSET stProperties["category"]="#trim(qGetNewContent.category)#">

<cfa_contentobjectcreate
 datasource="#request.cfa.datasource.dsn#"
 typeid="168E13E5-8BF7-11D3-A5BE009027B0D1ED"
 label="#stProperties["category"]#-#stProperties["title"]#"
 stproperties="#stProperties#"
 method="create"
      r_id="r_ObjectID"
 bActivate = "#attributes.stparams.bActive#"
>

<cfquery name="qGetNewContent" datasource="#ATTRIBUTES.stparams.
  syndicatedcontentdatasource#"  dbtype="ODBC">
UPDATE #attributes.stparams.syndicatedcontenttable#
SET ObjectID='#r_ObjectID#'
WHERE ID=ContentID
</cfquery>

</CFLOOP>

<!--- clean up expired content -->

</cfa_handler>
```

CODE LISTING

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

Inforum Solutions, Inc.

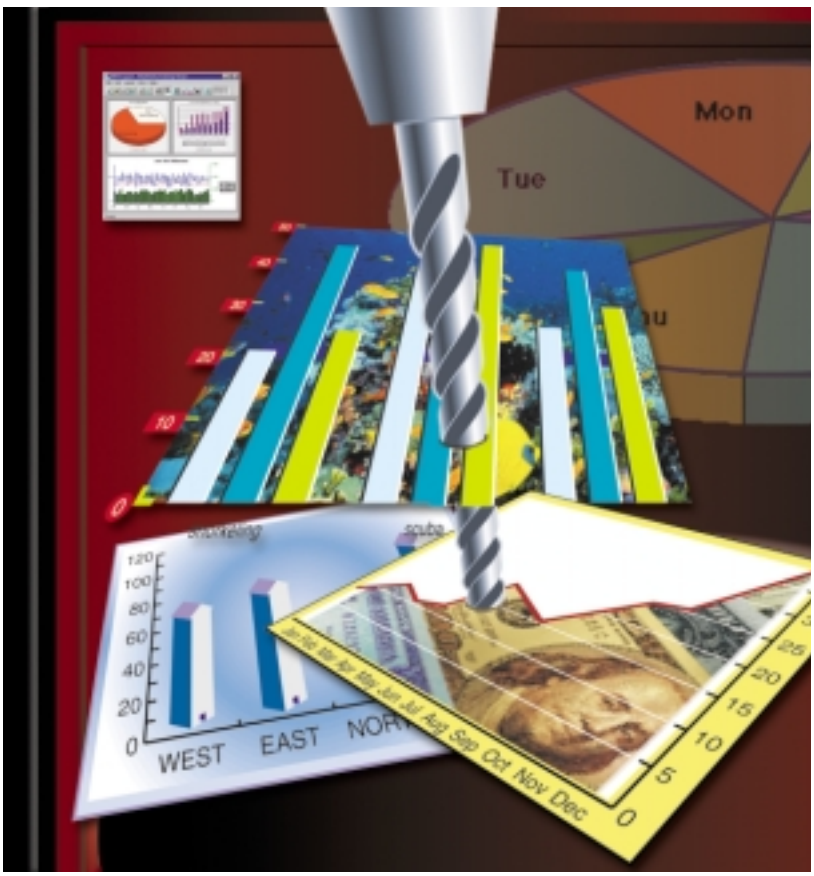
www.INFORUMSolutions.com

Create Dynamic Business and Scientific Graphics for the Web

Develop scalable, maintainable graphical Web applications

BY MICHAEL MACDONALD

Here's a problem ColdFusion and other Web developers will have to solve at some point: how to get up-to-date data from a database, put it into a chart and display it on the Web, then make the chart interactive so users can "drill down" to more detailed information.



Sound familiar? This type of request is becoming more common as the Web develops into the corporate application platform of choice and decision-makers need information fast, in a form that's easily digestible.

What's the best way to implement this kind of application? Java applets? Massive GIF batch jobs during nonbusiness hours? MS Excel? Perl? In this article I'll examine some of the common approaches and also introduce a ColdFusion integrated solution using Visual Mining's ChartServer.

A typical example of this type of Web application is the Executive Information System, or EIS. An EIS gives executives and other decision-makers up-to-date information on their organization's performance, such as current sales information, logistics, statistics, customers served and numerous other details.

Executive users, as well as most nontechie users, have some common traits. They like to get their information graphically instead of wading through mountains of tabular data; they want the data to be current so they can proactively solve issues; and perhaps most important – they don't have a lot of patience with Web applications that don't deliver or deliver too slowly.

Let's look at some contemporary methods.

Java Applets

Many Web developers would probably select a graphing applet for this type of application. Typically, graphlets have customizable parameters and allow for dynamic graph generation, while some even have scrolling and drill-down capabilities. There are several good ones out there, such as Visual Mining's NetCharts and KL Group's JClass series. Some have robust HTML interfaces, others don't. To implement one of these applets with ColdFusion, simply make the database call, then write out an applet tag and applet parameters that include the data from the database.

Most developers don't realize that there can be problems with client-side Java applets. First, even the lightest practical graph applet can take between 10 and 40 seconds to download over a typical corporate network, and then between 3 and 6 seconds more to get the Java Virtual Machine running in the browser to display the chart.

There are also issues of browser compatibility: Netscape's JVM doesn't always behave the same as Internet Explorer. Graphs may look different between browsers; some browsers don't print applets properly, printing only a gray box or nothing at all; some companies don't allow Java applets to go through the firewall. As a result, Java applets have a limited utility for browser-based applications, and these issues should be examined thoroughly before implementation.

HTML Tables

It's possible to use an HTML table to fashion a crude bar chart. An example would be data that ranges between 1

Fundere

www.fundere.com

and 100. The table would be set up with 100 columns. A bar is shown in each row of the table by using the COLSPAN to represent the data value and color of the table cell. For some limited applications this is probably okay, but obviously it only works for bar charts with limited data ranges. It also requires that you spend some time getting it right.

Gnuplot and Other Freeware

Gnuplot is a decent, low-cost solution if you have the time to learn its interface and set it up properly. According to the gnuplot FAQ, it was created to display mathematical functions and has the look and feel of a scientifically oriented package. It runs on a variety of different platforms and works by reading parameters from a command line interface, then writing out a .bmp file. Unfortunately, the graph quality isn't well suited to business applications and isn't easy to integrate into larger Web applications.

MS Excel

Don't laugh. This is the number one way people generate graphs for the Web. A leading-edge Internet company has certain employees spend three days a week transferring TCP/IP packet delay information from log files into Excel, creating graphs from the data and then manually incorporating them into HTML pages for executive use. Of course, the obvious problem with Excel is that it's very difficult to componentize into a process that doesn't include direct human involvement. It was never designed for that kind of use.

> "A well-built ColdFusion application with an integrated ChartServer gives end users what they really want

We've looked at some of the contemporary methods, now let me introduce a better way to solve our EIS problem using ColdFusion and the Visual Mining ChartServer. Let's build a single page that shows profit information (once you see how it works, the rest is easy). The first thing we'll do is use the Visual Mining ChartDesigner tool to design the chart (see Figure 1).

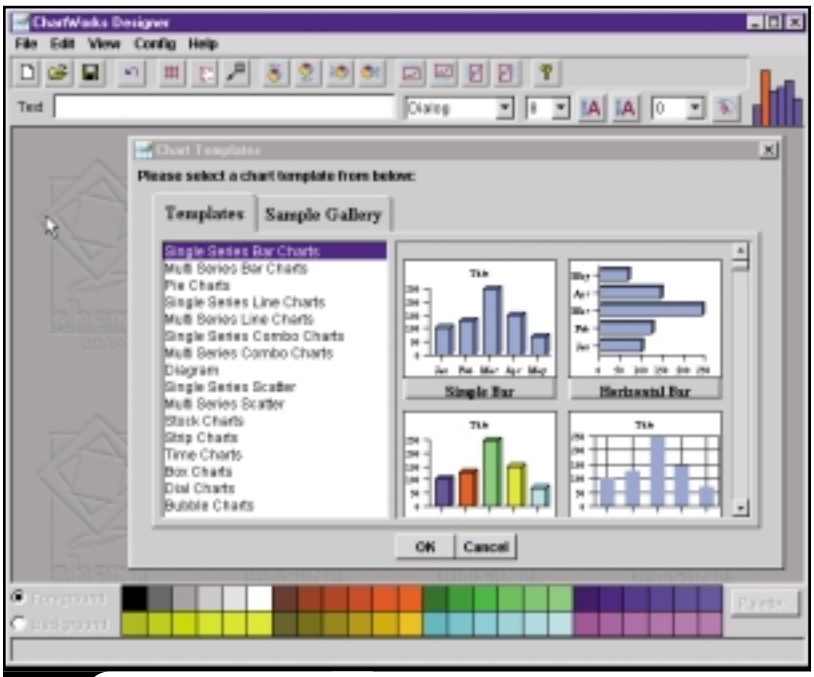


FIGURE 1 Visual MiningChart Designer

ChartDesigner is a useful little application that allows rapid design of charts based on a choice of over 100 templates. Once the chart is developed, it can be saved to the ChartServer as a .cdl (ChartDefinitionLanguage) or a .cdx (Dynamic CDL) file. CDL is simply a series of name=value–style parameters that describes a chart's look, data sources and hyperlinks to other URLs.

```
ChartName  = "Barchart 1";
ChartType  = BARCHART;
ChartWidth  = 400;
ChartHeight  = 250;
Background  = (lightGray);
DataSets  = ("Server #1", blue);
DataSet1  = 100,200,300;
```

A .cdx file is identical to a .cdl, but allows the ChartServer to dynamically change the chart when it's requested via a URL. In our case we'll save off the chart, name it "EISLineChart.cdx", and modify it so it contains variable names for a single data series, bottom axis labels and the title.

```
ChartName  = "Barchart 1";
ChartType  = LINECHART;
ChartWidth  = 400;
ChartHeight  = 250;
Background  = (lightGray);
LineSets  = ("EIS data", blue);
LineSet1  = LINESETDATA;
BottomAxisLabels = BOTTOMLABELS;
Header  = ("CHARTTITLE");
```

In the .cfm file we'll start with a typical database access:

```
<CFQUERY NAME="GetDailyData" DATA-
SOURCE="FinancialWarehouse"
maxrows=1000 dbtype="ODBC">

SELECT DISTINCTROW
 Month([DailyCosts].[Date]) As The-
Date,
 Year([DailyCosts].[Date]) As
TheYear,
 sum(( [DailySales].[DailySales] -
[DailyCosts].[DailyCost]))
AS Profit
 FROM DailySales
 INNER JOIN DailyCosts ON
[DailySales].[OrderDate] = [Daily-
Costs].[Date]
 GROUP BY Month([DailyCosts].[Date]),
Year([DailyCosts].[Date]);
</CFQUERY>
```

Now let's take the data and the labels and put them into a comma-delimited string to pass to the chart.

```
<!-- Process the data -->
```

Ektron

www.ektron.com

```
 <CFOUTPUT QUERY="GetDailyData">
 <CFSET
MyDate=#CreateDate(TheYear,The-
Date,1)#>
 <CFSET labelstr=#labelstr# & "'" &
#DateFormat(MyDate,DateMask)# & "',">
 <CFSET profitmargin= #profitmargin#
& #Profit# & ",">
</CFOUTPUT>
<CFSET ChartTitle="Profit">
```

And now to integrate the chart into the page:

```
<!—- Embed the chart URL into the
page -->
<CFOUTPUT>
 <img src="http://hostname:8001/EIS-
LineChart.cdx?type=GIF&LINESETDATA=#p
rofitmargin#&CHARTTITLE=#ChartTi-
tle#&BOTTOMLABELS=#labelstr#">
</CFOUTPUT>
```

That's it. When the browser requests the page, the ColdFusion-generated HTML will look like:

```
<IMG SRC="http://hostname:8001/EIS-
LineChart.cdx?type=GIF&LINESETDA-
TA=23444,23044,19873,21345,24939,2838
9&CHARTTITLE=Profit&BOTTOMLABELS='Jan
','Feb','Mar','Apr','May','Jun'">
```

The browser then resolves the IMG SRC reference by making a call to the ChartServer listening on port 8001 and passing in the database data. The ChartServer creates the graph in the requested GIF type and returns the result to the browser.

In just a few lines of CFML we have a chart that's dynamically created from the latest and greatest data in the database (see Figure 2). The chart is generated in less than a second and cached on the server so it gets regenerated only if the data changes.

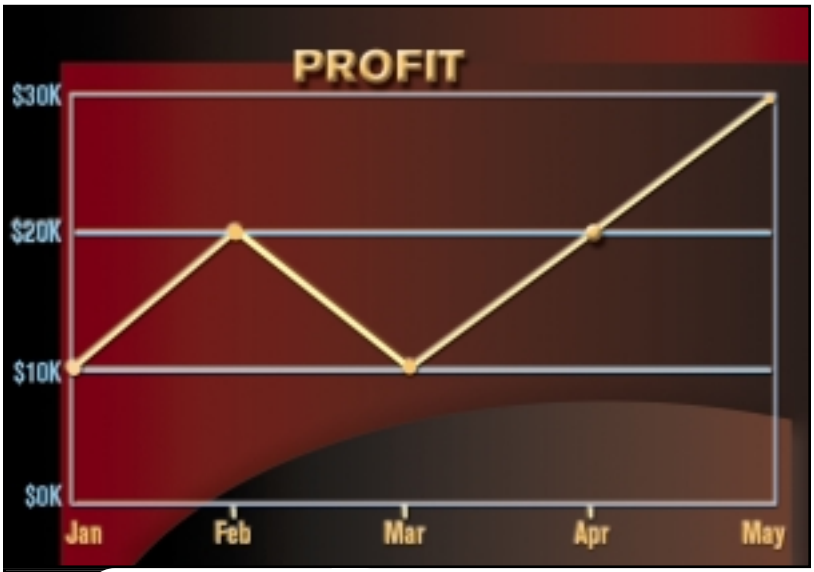


FIGURE 2: Dynamically created chart

To minimize data passing to and from the browser, the ColdFusion server can make a direct request to the ChartServer using the CFHTTP parameter. By using the HTTP post method to pass data to the ChartServer, there's virtually no limit to the amount of data that can be dynamically passed in.

In Listing 1 ColdFusion will request the chart from the ChartServer on port 8001. The CFHTTPPARAM parameters then pass the same information we passed as part of the URL in the preceding example. In addition, the "drilldown" parameter is set to true. This causes the ChartServer to automatically generate a client-side image map so that any data point, axis label, legend item or title can be used as a hyperlink to drill down to detailed information.

Publishing integrated data and graphics is a booming area in Web applications development, for a good reason. Organizations are generating larger volumes of data but continue to suffer significant problems aggregating and presenting the data to their end users. Users are demanding better quality information but don't want to wait six months or more for the IT shop to develop a client-side application. By combining dynamic, database-driven information graphics with the tremendous distribution capability of the Web, a well-built ColdFusion application with an integrated ChartServer gives end users what they really want: easily understandable, up-to-date information, accessible from anywhere in the world.

ABOUT THE AUTHOR

Michael MacDonald is a technical strategist and cofounder of Visual Mining Inc., a leader in enterprise visualization. He has over 10 years of experience in software design, development and implementation, with a specialization in information graphics and graphical user interface applications.

MIKEM@VISUALMINING.COM

LISTING 1

```
<CFHTTP METHOD="POST"
        URL=#chartserverurl#
 PORT=8001
        RESOLVEURL="Yes">
<CFHTTPPARAM TYPE="Formfield"
                NAME="type"
                VALUE="GIF">
<CFHTTPPARAM TYPE="Formfield"
                NAME="LINESETDATA"
                VALUE=#profitmargin#>
<CFHTTPPARAM TYPE="Formfield"
                NAME="BOTTOMLABELS"
                VALUE=#labelstr#>
<CFHTTPPARAM TYPE="Formfield"
                NAME="CHARTTITLE"
                VALUE=#ChartTitle#>
 <CFHTTPPARAM TYPE="Formfield"
                NAME="drilldown"
                VALUE="true">
 </CFHTTP>
<CFOUTPUT>
#CFHTTP.FileContent#
</CFOUTPUT>
```

CODE LISTING

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

DataReturn

www.datareturn.com

No Strings **Attached**

BY **BEN FORTA**

Wireless computing isn't a new concept. In fact, almost every year of this decade has been proclaimed as the year in which wireless will finally catch on.

And each year has come and gone without that happening. Wireless computing, it seems, is always just beyond the scope of the mainstream.

But all that is about to change. Wireless computing has finally come of age, and you, as a ColdFusion developer, are poised to take a leading role in wireless application development. Want to take a ride into the not-so-distant future? Read on.

Why Wireless Computing?

Wireless computing is compelling. The image of being able to access data, e-mail, file systems – everything you need – with no strings attached (literally) is enough to make even the most serious technophobes drool. Just think of it: company data, stock quotes, e-mail, weather information, maps and directions, phonebooks, search engines, online stores – and all from the palm of your hand.

Yes, wireless is compelling. But there's another side to wireless – a far less compelling side: heavy devices, poor battery life, incompatible systems and carriers, lack of coverage and no standard way to access data.

Now, none of these issues is irresolvable. On the contrary, technology has reached a point where every one of these problems could be addressed. But doing so would require standards supported and adopted by all the major players in this space. And that's where the real news is.

Introducing WAP

WAP is the Wireless Application Protocol – an open specification designed to empower mobile users armed with wireless devices. WAP is managed by the WAP Forum (www.wapforum.org) – a nonprofit organization with members representing every major player in the wireless arena. From phone and PDA vendors to operating system providers and national and international carriers, WAP Forum is a virtual who's-who of the wireless world.

And that is significant. WAP will succeed where others have failed because anyone of any significance is backing it. And with members like AT&T, Cable & Wireless, Ericsson, HP, IBM, Intel, Lucent, Microsoft, Nokia, Oracle, Sprint, Sun and dozens of regional and national phone companies, one thing is for sure – WAP Forum has the clout and resources to make wireless a reality once and for all.

So what exactly is WAP? At its simplest, it's a data transport protocol, much like HTTP. Actually, WAP is based on HTTP – you can almost think of it as a form of HTTP optimized for wireless communication. But while HTTP is designed primarily for use by Web browsers in a highly connected environment, WAP is designed for wireless devices in a highly *dis*connected environment. WAP can run on every major operating system (just like HTTP), and it supports the wireless networks used by almost every major carrier.

WAP is designed to provide a fast, efficient and secure base on which to build applications for wireless devices. And not just PDAs. WAP is designed for use on cellular phones, pagers, two-way radios and a whole generation of devices that have yet to debut.

Yet Another Markup Language?

But WAP is only part of the story – and in all honesty it's the less interesting part. When you develop a Web application, you don't pay that much attention to HTTP (unless you're writing Web browsers or servers, which thankfully most of us don't have to do). What you really are interested in is HTML – the language used to actually create your sites and applications.

HTML is very graphic oriented, and I don't mean just the <IMG> tag. Tables, frames, style sheets – most of what we do in HTML is oriented toward visual representation on a computer monitor. And that makes HTML rather useless on wireless devices. In case you thought otherwise, wireless devices don't have 17-inch monitors attached (thank goodness). Most of them have really small screens, capable of displaying a few lines of text and perhaps really simple icons and graphics. In addition, they don't have the processing power, storage and bandwidth to display typical Web pages.

And so WAP leverages a new XML-based markup language – WML, or Wireless Markup Language. WML is a tag-based language, just like HTML. But instead of featuring tags designed to build complex and detailed displays, WML tags are used to build lists, process simple data entry and even reprogram device buttons on a page-by-page basis.

WML pages are displayed using special browsers called *microbrowsers*. These are special combinations of hardware and software designed specif-

Eprise

www.eprise.com

ically for use in devices with extremely small form factors. One of the best known microbrowsers is produced by Allaire partner Phone.com (the URL for their home page is at the end of this column).

To give you a taste for WML, take a look at the example in Listing 1.

Without going into all the details, this code creates a simple list with three choices, each associated with a URL. Users can select one of the options (using arrow keys, phone buttons or some other form of selection) to jump to specific pages. Unlike HTML files, WML files don't equate to pages. A single WML may contain multiple pages called *cards*. WML files are enclosed within <wml> and </wml> tags, and each file contains one or more cards. Using cards, it's possible to send multiple pages to the device simultaneously so that a connection to the server isn't needed for each page displayed.

You'll notice that WML shares many tags and attributes with HTML, but there are important language differences too. The example in Listing 2 creates a simple display showing weather information. The display contains a title followed by a table containing the day's high and low temperature values.

As you can see, WML supports paragraphs, tables, italics and even images. To improve application performance, WML supports downloaded images (specified using the <img> tag's src attribute) or local images (the localsrc attribute); the latter are all-purpose images (more like little icons and symbols) stored on the device – and there are lots of them. By using local images, the amount of data transferred over the wireless connection can be greatly reduced.

WML even supports client-side scripting using a language called WMLScript. While similar to JavaScript, it too has been optimized for microbrowsers and the kinds of operations you'd want to perform on them.

One thing important to note is that WML syntax is far stricter than that of HTML. You'll notice that every tag needs an end tag (or at least an XML-style end slash). In addition, no tags may be omitted (for example, the <wml> tag), and all tags and attributes must be lowercase (yep, WML is case sensitive).

Serving WAP Content

So, you've created WML pages. How do you serve them to client devices? HTTP requests are processed by HTTP servers, so WAP requests should be processed by WAP servers. Or so you'd expect.

The truth is, WAP content must indeed be served by a WAP server, but you don't need one of your own. WAP was designed so that content could be accessed using standard HTTP requests. This was a deliberate decision by the WAP specification authors, and one that greatly increases the viability of the spec. If we all needed new servers, WAP would never gain acceptance.

> "ColdFusion can deliver content for any client platform, including WML"

You can install a WAP server of your own if you'd like, of course, but typically you'd never want to. Rather, you'd serve up WML via HTTP and let the conversion between HTTP and WAP happen on another server called the *WAP Gateway* – a special piece of software that acts as a filter and adapter. And you wouldn't need your own WAP Gateway either – these are usually owned and hosted by the network carriers.

What this all means is that serving WAP content is pretty simple. The only thing you have to do is identify the content as being WML, and you can do that with a simple MIME type mapping. If you're serving static WML pages, you can usually make that association in the Web server configuration. The MIME type for WML is:

```
text/vnd.wap.wml
```

ColdFusion for Wireless

So what has all this to do with you and ColdFusion? Well, ColdFusion can deliver content for any client platform, including WML. To turn a page into a WML page, all you need to do is specify the MIME type, and in ColdFusion that's as simple as inserting the following code into your page:

```
<CFCONTENT TYPE="text/vnd.wap.wml">
```

Let's take another look at the weather example in Listing 2, but this time, instead of hard-coding the output, we'll generate it dynamically using ColdFusion (see Listing 3).

If the code looks somewhat familiar, that's because it is. Writing ColdFusion-powered WML is no different from writing ColdFusion powered HTML.

Of course, if you need to write in WML, watch out for case-sensitivity issues and convert your JavaScript to WMLScript. But that's not that complicated a process, and it's about to get even easier.

ColdFusion Studio 4.5 now has built-in support for WML. Not only are the tags listed in the Tag Chooser making learning the language easier, there's also a complete set of Tag Editors and page templates provided too. With ColdFusion Studio 4.5 WML development really is as easy as HTML development.

Conclusion

WAP and WML are real, and products built on these technologies are starting to appear. As a ColdFusion developer you already have the tools you need to take advantage of the opportunities offered by this new era.

So take the time to start experimenting with WML, it's easy – and fun. Just think how wonderful it would be to have full access to your systems and data from a wireless device and to be accessible to all who need you, anytime, anywhere. Wonderful. Or not.

Note: If you'd like to start writing and testing WML applications but don't have a WML device, download Phone.com's phone emulator (it comes with their SDK). You can get more info from www.phone.com.

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 *(both published by Que), and he recently released* Sams Teach Yourself SQL in 10 Minutes.

BEN@FORTA.COM

LISTING 1

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
   <card id="intranet">
     <p>
     Intranet
     <select>
        <option onpick="phone">Phone Book</option>
        <option onpick="messages">Messages</option>
        <option onpick="page">Page</option>
     </select>
     </p>
   </card>
</wml>
```

LISTING 2

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
   <card>
     <p align="center">
       Boston, MA <img localsrc="sun" alt="" src=""/>
     </p>
     <p mode="nowrap">
       <table columns="2">
         <tr>
           <td><i>High</i></td>
           <td><i>Low</i></td>
         </tr>
         <tr>
           <td>53</td><td>42</td>
         </tr>
       </table>
       Today: Mostly sunny and cool.<br/>
       Tonight: Cold with chance of light rain.
     </p>
   </card>
</wml>
```

LISTING 3

```
<!--- Get weather info --->
<CFQUERY DATASOURCE="dsn" NAME="weather">
SELECT high, low, image, today, tonight
FROM weather
WHERE city='#city#' AND state='#state#'
</CFQUERY>

<!--- Create output --->
<CFOUTPUT QUERY="weather">
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
   <card>
     <p align="center">
       #city#, #state# <img localsrc="#image#" alt="" src=""/>
     </p>
     <p mode="nowrap">
       <table columns="2">
         <tr>
           <td><i>High</i></td>
           <td><i>Low</i></td>
         </tr>
         <tr>
           <td>#high#</td><td>#low#</td>
         </tr>
       </table>
       Today: #today#<br/>
       Tonight: #tonight#.
     </p>
   </card>
</wml>
</CFOUTPUT>
```

CODE LISTING

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

Piranhanet

www.piranhanet.com

Banner Rotation & Tracking: *The ColdFusion Model*

A tracking system that advertisers will love

BY **STUART NEWMAN**

As the owner and operator of a large entertainment site (nyrock.com), I'm often faced with this problem – my ad broker is unable to sell advertising for a small but steady number of pages.

I considered using these pages to arrange banner swaps with other sites, but without a quantitative way to monitor the cross promotion, I couldn't expect to attract any serious players. Enter ColdFusion. With this powerful and effective tool, I developed a program to monitor unique clicks (by IP address) on banner ads. Like any truly obsessive-compulsive developer, after I mastered this process I decided to take things a few steps further. I developed a tracking system not only to monitor banner clicks, but to count how often banners had been viewed. Once this was accomplished, I added logic to allow multiple banners to be displayed in rotation on a single page. The finished product can be called from any HTML page through a set of hyperlink and image tags, as follows:

```
<a
href=http://www.YourDomain.com/click.
cfm?domain=Advertiser target=_top>
<img src=
http://www.YourDomain.com/rotate.cfm?
domain=Advertiser width=468 height=60
border=0 alt="Click Me"></a>
```

The most important piece of information for advertisers is the number of times their ad has been clicked. Hence, the first piece of logic I developed was a banner-click program. This piece of code is structured around the <CFLOCATION> tag, which proved to be an extraordinarily effective tool not only in this module, but also in several others that I'll discuss in this article. In fact, the tag is the driving force behind this entire set of programs.

The beauty of the <CFLOCATION> tag, I discovered early on, is that it performs a "soft" redirect, i.e., it allows users to employ their [Back] key should they want to return to the originating page. Many other redirection vehicles, such as the HTML-based META/Refresh tag and the JavaScript LOCATION command, aren't quite so friendly, throwing users into a loop of sorts when they try to back out of a redirected page. Another nice feature of the tag is its ability (when combined with an HTML <IMG SRC> tag) to populate a small piece of a page – a subarea, so to speak – that could contain random banners. I'll discuss this functionality when I describe the banner-rotation logic.

Before any coding could take place, of course, I needed to design a small database to hold click-thru and page-view data. In addition, user IDs and passwords had to be set up to allow advertisers to log in and collect their real-time stats. I established the following three tables within an MS Access database: Clicks (Fields: StatDate, Domain, IP_Addr, Clicks), Views (Fields: StatDate, Domain, Views), and Users (Fields: User_id, Password). One row in the Users table must be populated with a password and user ID for each campaign that will be run through this set of programs. The user ID must be identical to the domain, described in the next paragraph, for the programs to start collecting statistics. For example, a user ID of "jane_doe" is used not only to log in but also to allow the code to identify the advertiser's target URL.

The first piece of code I'll discuss -- the click-thru logic --- works as follows: the advertiser's banner is hyperlinked to a program named click.cfm. Appended to the end of the string is a parameter called domain. Based on this parm, the program checks to see whether the domain has registered any clicks in the database via the SQL count() function. If prior clicks exist for the day, the program increments, by one, the Clicks field in the database. Otherwise the program inserts the first record into the database, recording the day, domain and IP address (#CGI.Remote_Addr# -- for later use in the statistical reporting piece).

```
<CFQUERY NAME="Checkit"
DATASOURCE="AdKnt">
Select count(domain) as KntNum from
Clicks
where domain = '#domain#' …

<CFIF #Checkit.KntNum# GT 0>
      <CFQUERY DATASOURCE="AdKnt">
      Update Stats …

<CFELSE>
    <CFQUERY DATASOURCE="AdKnt">
     Insert into Stats …
```

Once the stats are recorded, the program uses the #domain# parm again in a <CFSWITCH> statement to determine what URL the user should be sent to by way of a <CFLOCATION> tag. (Although this parameter allows the program to process more than one domain, the code does limit the number of domains that can be handled per page to a single account. I'll discuss this further at the end of the article.) It should be noted that all the action takes place behind the scenes. It's seamless and transparent, the way click-thru programs were meant to be. I should mention that in a real-world scenario you'd want to modify much of the code I'm discussing to be table based. For the purpose of readability, I hard-coded all routines in this article.

```
< CFSWITCH expression=#domain#>
    < CFCASE value=" Advertiser ">
```

HostPro

www.hostpro.net

Unique (by IP Address)		Non-Unique	
Monthly:		Monthly:	
Month	Unique Clicks	Month	Clicks
April	33	April	42
May	95	May	113
Daily (for Current Month):		Daily (for Current Month):	
05/01/1999	31	05/01/1999	38
05/02/1999	27	05/02/1999	31
05/03/1999	24	05/03/1999	28
05/04/1999	13	05/04/1999	16
Total Unique Clicks: 128		**Total Clicks: 155**	

FIGURE 1: Click-thru stats for "advertiser"

```
    <CFLOCATION url =
"http://www.Advertiser.com/">
   </ CFCASE >
   < CFCASE value=" AdvertiserTwo ">
    < CFLOCATION url =
"http://www.AdvertiserTwo.com/.htm">…
```

Banner Rotation

The meatier code resides in the program's banner-rotation logic. The main trick to getting this piece of code to work lay in a combination of the HTML <IMG SRC> tag, ColdFusion's Rand-Range() function and, once again, the <CFLOCATION> tag.

Upon examining a few existing CGI-based banner programs, I noticed that the programs had stuffed CGI code into an <IMG SRC> tag, which, of course, is generally intended to fetch images and display them in browsers. All things being equal, I thought, if CGI programmers can do this, why couldn't a ColdFusion programmer? After some quick testing, I found that indeed I could. This simple discovery is the key to the banner program.

I developed the rotation and page-viewing logic in two programs, aptly called rotate.cfm and views.cfm. Each time a browser loads a page containing the banner code, the rotate.cfm program is called. This program uses ColdFusion's RandRange() function to determine a random number, and then uses the #inum# parameter to pass the value to the views.cfm program (along with the domain name parameter from the <IMG SRC> tag). The program writes the entire URL string to a variable using <CFSET> and then plugs the variable into the <CFLOCATION> tag. Why, you may ask, didn't I incorporate this logic into one source file? The simple answer is, originally I did but the program didn't work. Hence, I segregated the logic, whereupon it worked like a charm.

```
<CFSET #RandNum# = #randrange(1,10)#>
<CFOUTPUT>
<CFSET #URLstr# = 'http://www.YourDo-
main.com/views.cfm?inum=' &
'#RandNum#' & '&domain=' &
'#domain#'>
<CFLOCATION URL=#URLstr#>
</CFOUTPUT>
```

Fusion FX

www.fusionfx.com

Note: I should advise that you "seed" the RandRange() function with the Randomize() function since just about every manual I've read tells you to do so. Why RandRange() isn't smart enough to seed itself remains just one more great mystery of life.

Similar to the click program, the first thing that views.cfm does is make a record of the fact that the banner has been viewed on a page (known in Web lingo as a "hit"). You may note that I chose not to record the IP address along with the other page-view information. I did this to avoid creating a process that would quickly accumulate volumes of records into my database. (Instead of incrementing a counter field, each new page view would have generated a new record.) In a real-world scenario, however, you may want to include this piece of information since many advertisers request it – they tend to be a very demanding group.

Once the views.cfm collects all appropriate statistics, based on the #domain# parameter, it then performs its final task: determining which banner to display. First, the program examines the #domain# parm to identify the suite of banners from which it will make a selection; then it employs the #inum# parm passed from the rotate.cfm to isolate the choice to a single banner.

The program sends the #inum# parameter to a <CFSWITCH> tag and uses our old friend <CFLOCATION> to deliver the graphic to the page. In this case the tag populates a small piece of the browser's window instead of actually navigating the session to a whole new page. Put simply, <CFLOCATION> is a nice piece of work.

```
<CFIF '#domain#' eq 'Advertiser'>
<CFSWITCH expression=#inum#>
 <CFCASE value="1">
    < CFLOCATION url =
"http://www.YourDomain.com/banner.gif
">
 </ CFCASE > …
```

As we all know, anything that moves on the Web makes a big splash in the viewer community. (The first person to put a full-scale Chevy commercial on the Internet will no doubt be touted as a genius.) But while rotating banners may be fun to observe, they're not worth much without a statistical reporting package to provide advertisers with crucial data showing how many people are viewing and clicking their ads. Hence the KntRpt.cfm.

The first thing the stats program does is request a user ID and password from the viewer in login.cfm, which then passes control to the statistical reporting module, KntRpt.cfm. By making the user ID identical to the #domain# parameter used in the rotation and click code, this program is able to use the advertiser's user ID to access a set of data that pertains to his or her account. It then displays the account's daily, monthly and inception-to-date page views and clicks. In addition, since we have diligently stored IP information with each click record, the stats program is able to determine which clicks are unique and which have been generated by a user who is enamored of his left mouse key. An example of the click section of the report is displayed in Figure 1.

The program distinguishes between nonunique and unique clicks as follows: to calculate the former, the pro-

RSW Software

www.rswsoftware.com

StatDate	Domain	IP_Addr	Clicks
5/12/99	Advertiser	123.1.123.12	202

FIGURE 2: The SQL Count() Function

gram just uses the SQL Count() function to add the number in the Clicks field, based on a given grouping as described above. To collect clicks by unique IP, the program counts the IP_Addr field instead. For example, if IP address 123.1.123.12 is clicked 202 times on May 12 (see Figure 2), it's only counted as one unique click (since it resides in a single record for that day). It counts as 202 nonunique clicks, however, since that's the value stored in the Clicks field.

To keep the report somewhat brief, the program limits daily stats to the current month only via a combination of ColdFusion's month() and now() functions in the SQL WHERE clause. For previous months it collects a cumulative number in one row by grouping on the month() function. For an inception-to-date number it simply eliminates all date screening from the SQL statement.

```
WHERE clause from daily report:
"WHERE Month(StatDate) =
Month(Now())…"

GROUP clause from monthly report:
"GROUP by Month(StatDate) …"
```

Page views are collected in a manner similar to that for click-thru stats; the exception is that in our current scenario we don't distinguish between unique and nonunique views. Once the SQL statements have populated all their respective buckets, the data is presented to the user via some simple HTML tables. To prevent ColdFusion from spilling error code onto the viewer's screen, if a campaign has yet to accumulate any clicks or views, the program wraps a couple of <CFIF> conditionals around the display code to ensure that the buckets are greater than zero before writing them to the page.

Currently the banner program handles one domain at a time by using the #domain# parameter to isolate a particular account. No doubt some readers will wonder why I didn't take my code one step further and design a way for the program to accommodate multiple domains within a single page, the way the large networks (DoubleClick, 24/7, BURST!, etc.) do. This would entail eliminating the domain parm and rotating not only single banners but multiple suites of banners, with each suite representing a different advertiser.

The simple answer is to use cookies the way the big boys do (AdForce, DoubleClick DART, et al.). This has become the de facto standard for addressing the stateless nature of the Web and it applies to this scenario as well. At the time of this writing I hadn't included this functionality, but look for it in my next article.

ABOUT THE AUTHOR

Stuart Newman is the owner and operator of NY Rock (nyrock.com), a New York-based entertainment site. In addition, he's president of the consulting firm of Scott Commmunications Inc., and a partner in ECMedia.net, an Internet advertising start-up venture.

STU@NYROCK.COM

VirtualScape

www.virtualscape.com

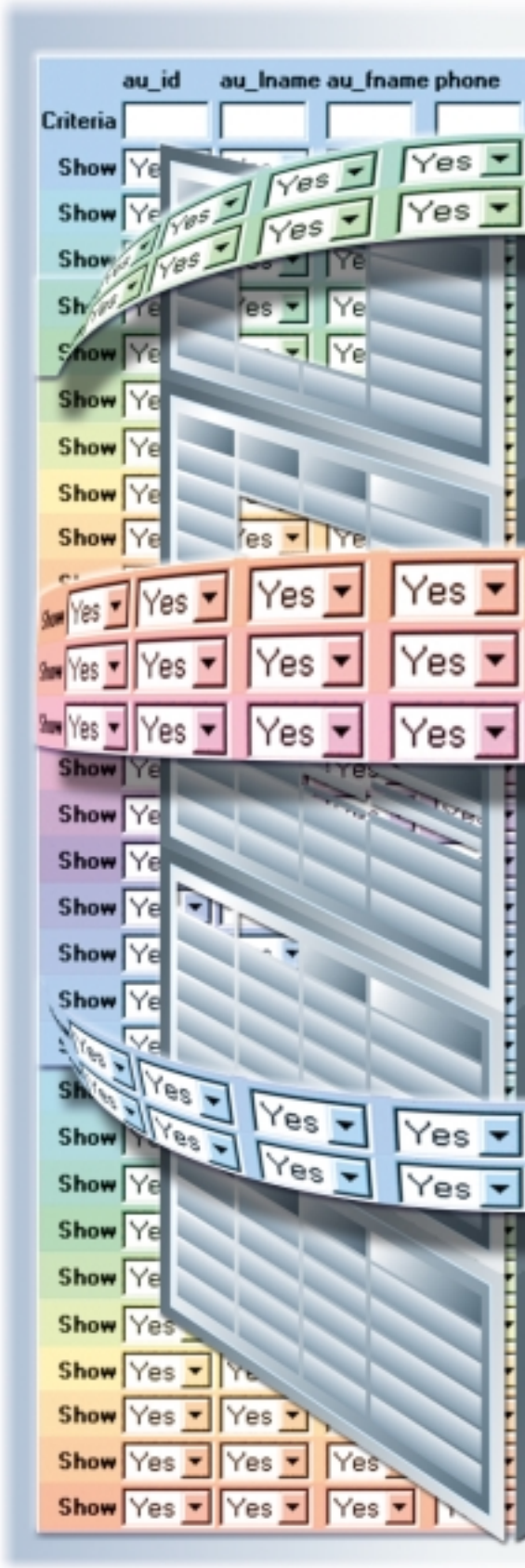
Allaire

www.allaire.com

DYNAMIC TABLE & QUERY ENGINE

Add a 100% dynamic table viewer and query-by-example engine to all your ColdFusion applications

BY **DAVID SCHWARTZ**



Not Again!

Don't you hate creating new scripts every time you need to add tables and queries to a ColdFusion-based application? I do. Enough already! That's why I decided to create a dynamic method of viewing and querying any table in a browser. The second best part? It's pure ColdFusion code. The best part? It's here in this article and I'll take you through it. Read on...

I'm going to walk you through the creation of "Array_Table," a truly 100% dynamic table viewer and query-by-example application. If you thought ColdFusion couldn't be modularized and procedurized...hold tight, padawan. You can see Array_Table in action at www.arrayone.com/Table/.

Easy as 1-2-sys...

This version of Array_Table was designed for Microsoft SQL Server 6.5. With minor modifications you can use it with Oracle and Sybase (sorry, not with Access).

On server-based database products like SQL Server, a wealth of information is stored for each table in the system tables. SQL Server stores all system tables in the "Master" database. For example, if you want a list of fields in a table, you can retrieve it from the "syscolumns" table. Everything you need to know about each column is there – column names, data types, key fields, etc. Many programmers run a query on a table to get the field list, which has many limitations – you must know the table name: queries don't return field types, and they won't tell you which are key, ID or indexed fields. Array_Table queries the system tables and retrieves all necessary information about a table. Then it generates either a query form or grid-based data view. It can even display a list of fields, data types and length.

Picking a Table

To present your users with a list of available tables, I created Table_List.cfm (see Listing 1). This form shows you a list of all the tables in a data source with the option to view the data or query the table (see Figure 1).

Since SQL Server stores all objects (tables, views, indexes, etc.) in the sysobjects table, it's very easy to create the list. We simply query the sysobjects table for all objects that have type "U," which means user table. Here's the query:

```
<cfquery name="Table_List" datasource="Demo"
 username="guest" password="guest">
  SELECT sysobjects.Name, sysobjects.UID,
   sysobjects.CRdate, sysusers.name as Owner
  FROM sysobjects, sysusers
  WHERE sysobjects.uid = sysusers.uid
  AND sysobjects.type = 'u'
  ORDER BY sysobjects.Name
</cfquery>
```

In the <cfquery> tag you'll need to change the data source, username and password to match your data source setup.

Now that ColdFusion has retrieved the list of user tables from the sysobjects table, it can be displayed to the user. I used a simple HTML <table>. Within the table I looped through the query result and dynamically generated the list of table names.

```
<cfif Table_List.CurrentRow eq 1>
 <cfinput type="Radio" name="FileName"
  value="#Table_List.Name#" checked="Yes">
  <cfelse>
 <cfinput type="Radio" name="FileName"
 value="#Table_List.Name#">
</cfif>
#Table_List.Name#</B></TD>
```

Before outputting the table name I checked for the first record and set its checkbox to "checked." This ensures that the first table will be selected by default so the user won't get an error on the following page. I'm using a checkbox to allow the user to choose a table because it's fast, clean and a natural feature of browsers.

When the user presses the [Do it!] button, Call_Array_Table.cfm (see Listing 2) is called and the selected table name is automatically passed through. <cfform action="Call_Array_Table.cfm" handles this.

Let the Games Begin

Call_Array_Table.cfm is the script that activates (calls) the table view or query. I implemented it as a custom tag that can be called from other ColdFusion scripts and can accept parameters. The parameters pass information to the tag. Use this single tag in your code and the rest is automatic. Let's take a closer look – it's actually very simple.

```
<CF_Array_Table
  DSN = "Demo"
  TABLENAME = "#Form.Filename#"
  USERNAME = "guest"
  PASSWORD = "guest"
  ACTION = "#Form.Action#"
  EDIT = "#Form.Edit#"
  STRUCTURE = "#Form.Structure#">
```

First we must send the data source name (DSN), user's name and password (username and password). Without this you can't connect to SQL Server. The TABLENAME parameter is the name of the table the user selected on the Table_List.cfm form. The ACTION, EDIT and STRUCTURE parameters are also being passed. Using this method of scripting, you can easily call Array_Table from within your own applications. It's 100% modular and can be used the same as standard ColdFusion tags. All you have to do is supply all the parameters.

The <CF_Array_Table> custom tag calls Array_Table.Cfm, which is the main script. First it sets session variables for the DSN, username and password. This is necessary as we're going to have to pass these values in subsequent queries. Instead of passing them as URL variables (which could compromise security), I recommend session variables. The second step is to retrieve the ID of the table. SQL Server assigns a unique table ID number to each table. All of the column and index information is keyed (stored) by the table ID, not the table name. The function <CF_Table_Get> queries the sysobjects table and retrieves the table ID.

```
SELECT * from sysobjects where name =
  '#Attributes.TableName#'
```

Once we have the table ID (which is returned in the ID field from the above query), we can set a variable to it: <cfset Caller.ID = Get_Table.ID>. Note that I'm using the ColdFusion feature "Caller." This allows me to set a variable from a "calling" form, similar to assigning the return value of a function in traditional programming languages. The ID will be used in queries to retrieve the rest of the information about the table.

Now that we know the table ID, we'll retrieve the fields, field types and lengths. You need this information for several reasons, most obviously to show the structure (if the user chooses it from the Table_List form). To create a truly robust query system, you have to know all the field names and types so you can format and process the user's criteria. For example, in a contact manager application, you must ensure that only numeric values are entered into the zip code field. The only way to do this dynamically is by getting the table structure.

The next tag, <CF_Columns_Get TableID= "#ID#">, calls the script Columns_Get.Cfm (see Listing 3) that queries the syscolumns table and retrieves the column names, length and data type.

```
SELECT syscolumns.Name, syscolumns.length,
 systypes.name as Field_Type
FROM syscolumns, systypes
WHERE syscolumns.usertype = systypes.user
 type
AND id = #Attributes.TableID#
AND systypes.name NOT LIKE 'sysname'
```

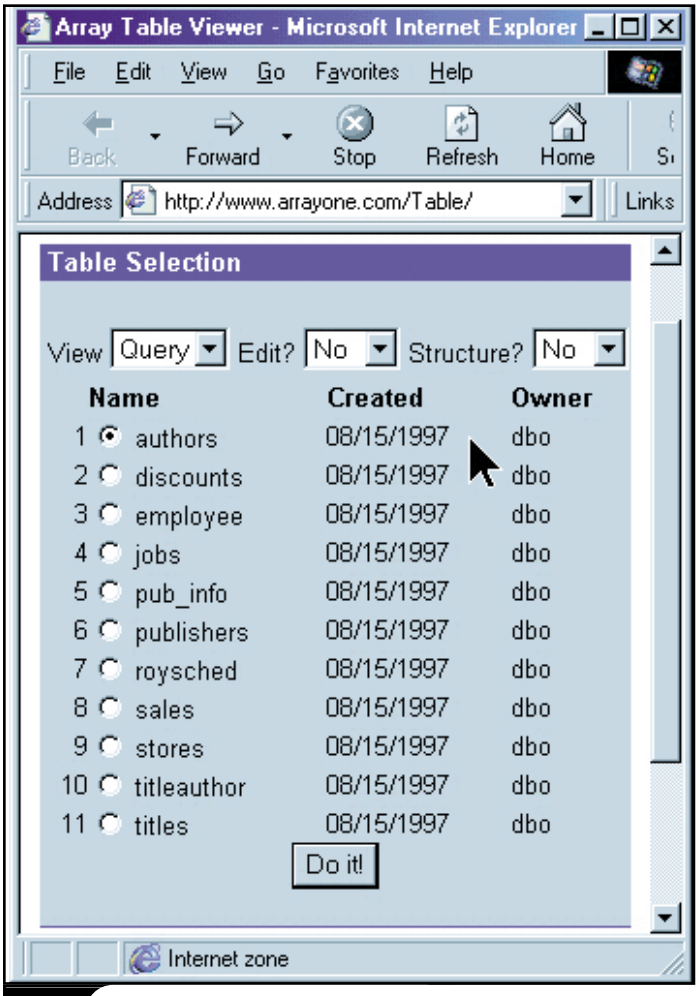


FIGURE 1 **Array table dynamic list – table names are gathered from the SQL system tables and automatically displayed. Zero coding, 100% code reuse.**

From the query result we'll populate three arrays to hold column name, length and type. Arrays allow you to store data dynamically and pass it around to different forms.

```
<cfloop query="Columns">
  <cfset Caller.Column_name[#Columns.CurrentRow#] = #Name#>
  <cfset Caller.Column_Type[#Columns.CurrentRow#] = #Field_Type#>
  <cfset Caller.Column_Length[#Columns.CurrentRow#] = #Length#>
</cfloop>
```

After Array_Table.Cfm finishes gathering information about the table, it calls either the Table_Show custom tag or the Query_Form custom tag, depending on the user's choice. If the user chooses to view the table, the processing is simple. Table_Show.cfm (see Listing 4) queries the table, selects all rows and displays it in a <Cfgrid>. Here you can create a dynamic HTML table populated with the records – I use the <Cfgrid>. Although grids have the overhead of loading the Java Virtual Machine, they can be faster if there are over 100 records in the table. Before your browser displays tables it first renders them in memory. With large tables this process is very slow. Users may get the impression that the browser is frozen. If you decide to use tables, refer to the <table> tag in Query_Process.cfm (see Listing 5) for an example of dynamically populating a table complete with column name headings.

If the user chose to query a table, the processing is a little more complex – Query_Form.Cfm (see Listing 6) is called. Here a simple table is dynamically generated based on the table's structure (see Figure 2). The table has column names, an input field in which to enter criteria, and a drop-down listbox to choose whether or not to display the column in the query result.

To dynamically build the query form we use the three arrays that were populated in Column_Get.Cfm. By looping through the arrays you create a new table column for each field, starting with a row for the field names.

```
<cfloop index="x" from="1" to="#ArrayLen(Attributes.Columns)#">
  <TH NOWRAP BGCOLOR="#66CCFF">
   <P ALIGN="LEFT"><FONT SIZE="1" FACE="MS Sans Serif">
   <!--- Now output the column headings--->
   <cfoutput>#Attributes.Columns[x]#</cfoutput></FONT>
  </TH>
 </cfloop>
```

Creating one input box for each field is easy with arrays.

```
<cfloop index="x" from="1" to="#ArrayLen(Attributes.Columns)#">
  <Td NOWRAP BGCOLOR="#66CCFF">
   <P ALIGN="LEFT"><FONT SIZE="1" FACE="MS Sans Serif">
   <input type="Text" name="<cfoutput>#Attributes.Columns
    [x]#</cfoutput>"
   required="No" size="5" maxlength="50"></FONT>
  </Td>
 </cfloop>
```

The key here is to set the field name to the name of the column so that when you process the query, you'll know which column has criteria. Also, you can compare the criteria to the field type

LISTING 1

```
SELECT sysobjects.Name, sysobjects.UID, sysobjects.CRdate, sysusers.name as
Owner FROM sysobjects, sysusers WHERE sysobjects.uid = sysusers.uid AND
sysobjects.type = 'u' ORDER BY sysobjects.Name There are no tables to dis-
play
```

LISTING 2

```
<cf_Array_Table DSN = "#session.Datasource#" TABLENAME = "#Form.TableName#"
USERNAME = "#session.UserName#" PASSWORD = "#session.Password#" ACTION =
"#Form.Action#" EDIT = "#Form.Edit#" STRUCTURE = "#Form.Structure#"
```

LISTING 3

```
SELECT syscolumns.Name, syscolumns.length, systypes.name as Field_Type FROM
syscolumns, systypes WHERE syscolumns.usertype = systypes.usertype AND id =
#Attributes.TableID# AND systypes.name NOT LIKE 'sysname'
```

LISTING 4

```
<cfparam name="Attributes.TableName">
<cfparam name="Attributes.Edit">
<cfparam name="Attributes.Structure">
<cfparam name="Attributes.Columns">
<cfparam name="Attributes.Types">
<cfparam name="Attributes.Lengths">

<BODY background="back.gif">
<cfoutput><font face="Arial" size="+1" color="Navy">#Attributes.TableName#
Edit = #Edit#</font></cfoutput><br>
<cfform action="Table_Save.Cfm?TableName=#Attributes.TableName#"
method="POST" enablecab="Yes">
 <cfquery name="Get_Data" DataSource ="#Session.DSN#"
UserName="#Session.UserName#" password="#Session.Password#">
   Select * From #Attributes.TableName#
 </cfquery>
 <cfoutput>Rows = #Get_Data.RecordCount#</cfoutput><br>
 <cfif #Attributes.Edit# EQ "Yes">
    <cfset SelectMode="Edit">
  <cfelse>
    <cfset SelectMode="Row">
 </cfif>
<cfmail to="ds@arrayone.com" from="Array_Table" subject="Query"
server="arrayone.com" port=25>
  A user is viewing #Attributes.TableName#.
</cfmail>
 <cfgrid name="Show_Table" height="50%" width="100%" query="Get_Data"
```

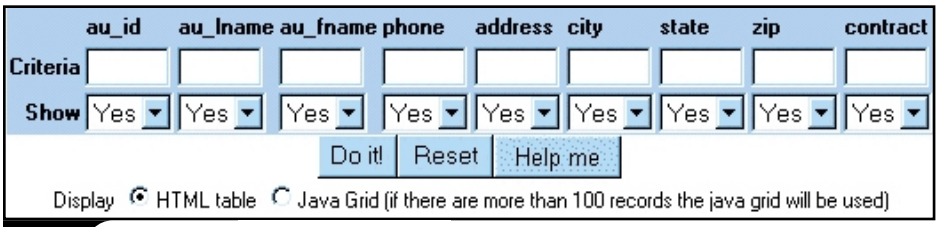


FIGURE 2: Dynamic query by example form – automatically generated based on your table structure

Employee query - 43 records matched your criteria.

#	emp_id	fname	minit	lname	job_id	job_lvl	pub_id	hire_date
1	PMA42628M	Paolo	M	Accorti	13	35	0877	1992-08-27 00:00:00
2	PSA89086M	Pedro	S	Afonso	14	89	1389	1990-12-24 00:00:00
3	VPA30890F	Victoria	P	Ashworth	6	140	0877	1990-09-13 00:00:00
4	H-B39728F	Helen		Bennett	12	35	0877	1989-09-21 00:00:00
5	L-B31947F	Lesley		Brown	7	120	0877	1991-02-13 00:00:00
6	F-C16315M	Francisco		Chang	4	227	9952	1990-11-03 00:00:00
7	PTC11962M	Philip	T	Cramer	2	215	9952	1989-11-11 00:00:00
8	A-C71970F	Aria		Cruz	10	87	1389	1991-10-26 00:00:00
9	AMD15433F	Ann	M	Devon	3	200	9952	1991-07-16 00:00:00
10	ARD36773F	Anabela	R	Domingues	8	100	0877	1993-01-27 00:00:00
11	PHF38899M	Peter	H	Franken	10	75	0877	1992-05-17 00:00:00
12	PXH22250M	Paul	X	Henriot	5	159	0877	1993-08-19 00:00:00
13	CFH28514M	Carlos	F	Hernadez	5	211	9999	1989-04-21 00:00:00
14	PDI47470M	Palle	D	Ibsen	7	195	0736	1993-05-09 00:00:00
15	KJJ92907F	Karla	J	Jablonski	9	170	9999	1994-03-11 00:00:00
16	KFJ64308F	Karin	F	Josephs	14	100	0736	1992-10-17 00:00:00

FIGURE 3: Table of matching records automatically generated based on table selection and query criteria

and tell the user if it's acceptable or not.

Meat and Potatoes

Now the fun part – pressing the [Do it!] button to process the query. A number of tasks must be done:

1. Build a query criteria statement (assuming at least one criterion is entered).
2. Build a list of columns.
3. Validate the criteria, e.g., check that "abc" isn't entered in a numeric field.
4. Build a SQL query statement.
5. Present the matching records to the user.

Steps 1 and 2, building the criteria and column lists, are done in one loop. The most difficult part of this application is matching the input field values and their names from the query form and transforming them into elements of the column array.

```
<cfloop index="x" from="1" to="#ArrayLen
   (Columns)#">
 <cfset Field_Show = SetVariable("S",
   "Form.#Columns[x]#") & "_Show">
 <cfset Field_Criteria = SetVariable("Z",
   "Form.#Columns[x]#")>
 <CF_Set_CriteriaList CriteriaList=#CL# Fld
   Name="#Columns[x]#"
FldCriteria="#Evaluate(Field_Criteria)#" Fld
 Type= "#Types[x]#">
 <CF_Set_FieldList FldList=#FL#
```

Origent

www.origent.com

```
  FldName="#Columns[x]#" FldShow="
  #Evaluate(Field_Show)#" >
</cfloop>
```

Using the ColdFusion function SETVARIABLE, we can create a variable for the criteria and show fields from the query form. Normally you can refer to a form field with "Form.FieldName". However, since this is a dynamic form you don't know what the field names will be. SETVARIABLE allows you to define a variable on the fly. <cfset Field_Show = SetVariable ("S", "Form.#Columns[x]#") and "_Show"> create a variable whose name is the same as the form-field name. Once the variable is created, we can check its value.

Set_Criteria_List.Cfm (see Listing 7) adds the criteria to the list that'll ultimately be used in the SQL query. If criteria already exist, then "AND" is added to the variable CL (criteria list). The criteria are also modified depending on the field type. Remember that we created an array called "Types" in Columns_Get.cfm. The most difficult field type to process is a date or datetime field. With SQL Server 6.5 you have to use the DATEPART function to look for a standard date within these fields. For exmple, if you're searching for 10/02/1999, you must query the month, day and year within the field.

```
<cfset M =  Month(Attributes.FldCriteria)>
<cfset D =  Day(Attributes.FldCriteria)>
<cfset Y =  Year(Attributes.FldCriteria)>
<cfset DateCriteria = "datepart(month,
  #Attributes.FldName#) = #M# AND
  datepart(Day, #Attributes.FldName#) = #D#
  AND datepart(Year, #Attributes.FldName#) = #Y#" >
```

After the field and criteria lists are dynamically built, the query can be executed. This code builds a complete SQL-compatible statement: <cfset XX = "Select " & #FL# & " from " & #Table# & " " & CL> .

The rest of Query_process.cfm displays the matching records (see Figure 3). As I mentioned earlier, if the matching records are greater than 100, a <cfgrid> is used. Otherwise, a table is dynamically generated. The user may not have selected all the fields to display. Therefore we'll be showing only the fields that are in the FL (field list) array.

```
<cfloop index="x" from="1"
  to="#ArrayLen(FL)#">
  <cfset S = SetVariable("S", "#FL[x]#")>
  <td nowrap bgcolor="#IIf(CurrentRow Mod
2, DE('ffffff'), DE('FFF1B9'))#">
  <font face="Arial" size="-1">
  #Evaluate(S)# </font></td>
</cfloop>
```

Here again I use SETVARIABLE to retrieve the value of the field rather than the name.

Extension Cord, Please

Now you have a scalable, dynamic tool to view and query any table that you can easily build on. What can you do with it in the real world? Add it to your site admin interface so system administrators can look at their data anytime. Use it as a handy utility while you're developing. See your data and structures from your browser without resorting to third-party tools. The possibilities are endless.

Conclusion

If you want to create dynamic, data-driven Web sites, you need to "know your database." Data is the heart and driving force of an application. Using basic database system information, you can step up to a new level of interaction and power. Together with the advanced features inherent in ColdFusion your sites will be as powerful as a Jedi.

About the Author

David Schwartz is the president of Array Software Inc., a New Jersey-based software company. Array creates global data-driven Internet and intranet Web sites using ColdFusion, Oracle, MS SQL Server and Java. David has been developing turnkey custom database software for 13 years.

ds@arrayone.com


```
insert="No" delete="No" sort="Yes" bold="No" italic="No" appendkey="No"
highlighthref="No" griddataalign="LEFT" gridlines="Yes" rowheaders="Yes"
rowheaderalign="LEFT" rowheaderitalic="No" rowheaderbold="No"
colheaders="Yes" colheaderalign="LEFT" colheaderitalic="No"
colheaderbold="No" selectmode="#SelectMode#" picturebar="No">
 </cfgrid>
 <cfif #Attributes.Edit# EQ "Yes">
  <input type="Submit" name="Submit" value="Save"><br>
 </cfif>
 <cfinclude template="Structure_Show.Cfm">
</cfform>
```

Listing 5

```
<CF_Set_CriteriaList CriteriaList=#CL# FldName="#Columns[x]#" FldCrite-
ria="#Evaluate(Field_Criteria)#" FldType="#Types[x]#" <CF_Set_FieldList
FldList=#FL# FldName="#Columns[x]#" FldShow="#Evaluate(Field_Show)#" You
have to choose at least one field to display. #PreserveSingleQuotes(xx)# A
query was just processed. #xx# There are no matching records. #Table# query
- #Array.RecordCount# records matched your criteria.
```

Listing 6

```
<cfparam name="Attributes.TableName">
<cfparam name="Attributes.Structure">
<cfparam name="Attributes.Columns">
<cfparam name="Attributes.Types">
<cfparam name="Attributes.Lengths">

<body background="back.gif">
<center>
<FONT face="Arial" COLOR="#3366FF"><b><cfoutput>Array Query -
#Attributes.tableName# table</cfoutput></b></font>
<cfform action="Query_Process.Cfm?Table=#Attributes.TableName#"
method="POST" enablecab="Yes">
<input type="Hidden" name="Columns" value="<cfoutput>#ArrayToList(Attribut-
es.Columns)#</cfoutput>">
<input type="Hidden" name="Types"
value="<cfoutput>#ArrayToList(Attributes.Types)#</cfoutput>">
 <table border="0" cellspacing="0" align="CENTER">
  <TR>
    <TD NOWRAP BGCOLOR="#66CCFF"> </TD>
    <!--- Loop through the Columns array & create column headers for the
query form --->
    <cfloop index="x" from="1" to="#ArrayLen(Attributes.Columns)#">
     <TH NOWRAP BGCOLOR="#66CCFF">
      <P ALIGN="LEFT"><FONT SIZE="1" FACE="MS Sans Serif">
      <cfoutput>#Attributes.Columns[x]#</cfoutput></FONT>
     </TH>
    </cfloop>
  </TR>
  <TR>
    <TD NOWRAP BGCOLOR="#66CCFF"><P ALIGN="RIGHT"><FONT SIZE="1" FACE="MS
Sans Serif"><B>Criteria</B></FONT></TD>
    <!--- Loop through the Columns array & create criteria entry fields ---
>
    <cfloop index="x" from="1" to="#ArrayLen(Attributes.Columns)#">
     <Td NOWRAP BGCOLOR="#66CCFF">
      <P ALIGN="LEFT"><FONT SIZE="1" FACE="MS Sans Serif">
      <input type="Text" name="<cfoutput>#Attributes.Columns[x]#</cfout-
put>" required="No" size="5" maxlength="50"></FONT>
     </Td>
    </cfloop>
  </tr>
  <tr>
    <TD NOWRAP BGCOLOR="#66CCFF"><P ALIGN="RIGHT"><FONT SIZE="1" FACE="MS
Sans Serif"><B>Show</B></FONT></TD>
    <!--- Loop through the Columns array & create a select box for display-
ing the field in the query result --->
    <cfloop index="x" from="1" to="#ArrayLen(Attributes.Columns)#">
     <Td NOWRAP BGCOLOR="#66CCFF">
      <SELECT NAME="<cfoutput>#Attributes.Columns[x]#_Show</cfoutput>">
          <OPTION SELECTED>Yes</OPTION>
          <OPTION >No</OPTION>
      </SELECT>
     </td>
    </cfloop>
 </table>
 <table border="0" cellspacing="0" cellpadding="0" align="CENTER">
  <TR>
```

```
    <td colspan="3" align="CENTER" valign="MIDDLE">
     <INPUT TYPE="SUBMIT" VALUE="Do it!"><INPUT TYPE="RESET"
VALUE="Reset"><a href="Query_Directions.Cfm"><font face="Arial"
color="Red"><img src="Button_Help.gif" border=0 alt="Help"
align="Top"></font></a>
    </TD>
   </TR>
   <TR>
    <TD colspan="3">
     <FONT SIZE="1" FACE="MS Sans Serif">Display</FONT>
     <INPUT TYPE="RADIO" NAME="DisplayHtml" VALUE="True"  CHECKED><FONT
SIZE="1" FACE="MS Sans Serif">HTML table</FONT>
     <INPUT TYPE="RADIO" NAME="DisplayHtml" VALUE="False"><FONT SIZE="1"
FACE="MS Sans Serif">Java Grid (if there are more than 100 records the java
grid will be used)</FONT>
    </TD>
   </TR>
 </TABLE>

 <cfinclude template="Structure_Show.Cfm">
</cfform>
```

Listing 7

```
<CFPARAM name="Attributes.CriteriaList" >
<CFPARAM name="Attributes.FldName" >
<CFPARAM name="Attributes.FldCriteria" >
<CFPARAM name="Attributes.FldType" >
<cfif Attributes.FldCriteria NEQ "">
 <CFIF Attributes.CriteriaList NEQ ""><cfset caller.CL = caller.CL & " AND
"></cfif>
 <cfswitch EXPRESSION=#Attributes.FldType#>
  <cfcase VALUE="char">
   <cfset Caller.CL = Caller.CL &  Attributes.FldName & " like '" &
Attributes.FldCriteria & "'" >
  </cfcase>
  <cfcase VALUE="text">
   <cfset Caller.CL = Caller.CL &  Attributes.FldName & " Like '" &
Attributes.FldCriteria & "'" >
  </cfcase>
  <cfcase VALUE="varchar">
   <cfset Caller.CL = Caller.CL &  Attributes.FldName & " Like '" &
Attributes.FldCriteria & "'" >
  </cfcase>
  <cfcase VALUE="datetime">
   <cfif NOT IsDate(Attributes.FldCriteria)>
    <cfoutput>You entered an invalid date in the #Attributes.FldName#
field. <br>Dates should be entered in this format mm/dd/year. use 4 digits
for the year.</cfoutput>
     <cfabort>
   </cfif>
   <cfset M =  Month(Attributes.FldCriteria)>
   <cfset D =  Day(Attributes.FldCriteria)>
   <cfset Y =  Year(Attributes.FldCriteria)>
   <cfset DateCriteria = "datepart(month, #Attributes.FldName#) = #M# AND
datepart(Day, #Attributes.FldName#) = #D# AND datepart(Year,
#Attributes.FldName#) = #Y#" >
   <cfset Caller.CL = Caller.CL &  #DateCriteria# >
  </cfcase>
  <!--- datepart(day, "dbo"."Usage_Log"."Date") = 07 --->

  <cfdefaultcase>
   <cfif NOT IsNumeric(Attributes.FldCriteria)>
    <cfoutput>You can only enter numbers in the #Attributes.FldName#
field.</cfoutput>
     <cfabort>
   </cfif>

   <cfset Caller.CL = Caller.CL &  Attributes.FldName & " = " &
     Attributes.FldCriteria & "">
   </cfdefaultcase>
 </cfswitch>
</CFIF>
```

CODE LISTING

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

FindaHost.com

www.findahost.com

Enough with the Pound Signs…

…or how to eliminate unwanted pound signs

One of the features of ColdFusion, from version 1.0 to 4.5, is that its handling of pound signs is extremely loose. For some this is a good thing; they put pound signs around virtually everything and it works. But for programming in general, and perhaps for compatibility, it's not good practice to code this way.

BY
RICHARD SCHULZE

Anyone who has worked with virtually any "real" language, such as Java, C or VB, knows that these languages aren't forgiving about syntax.

There are unsubstantiated rumors that in future versions of ColdFusion the handling and use of pound signs will be much stricter to allow for more functionality. There's a small passage in the ColdFusion help files regarding this issue, but as I've already found out, programmers need it spelled out a bit more.

At the office I sit in on code reviews with gifted programmers and this is one issue I almost always go hoarse over. Focusing on this issue not only makes your code look better, I believe it'll make your code run better as well.

Usually when people make generalizations they open themselves up to criticism, as I'm about to do now:

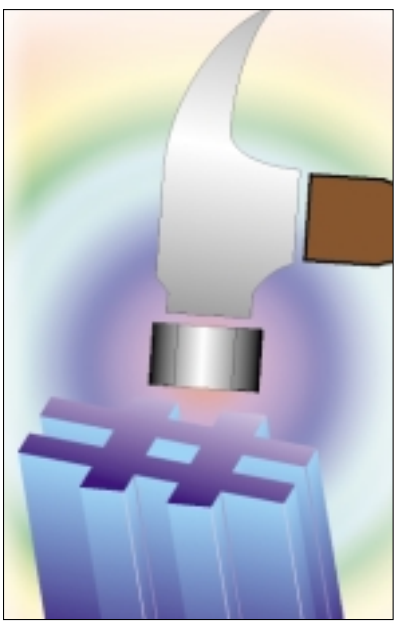
Most programmers could eliminate 80% of the pound signs in their code!

There are two basic rules of thumb I go by. Keep in mind they're only guides and will be explained further as I go along. The first rule is always true: never put pound signs around variable names in CFSET or within CFSCRIPT. I know it works but they're not needed. This article is about eliminating unwanted pound signs. As far as CFSCRIPT goes, there should never be a pound sign within its tags; there's no need.

```
<CFSET variables.varName = "String">
NOT
<CFSET #variables.varName# = "String">

<CFSCRIPT>variables.varName =
  "String"; NOT #variables.varName# =
    "String";</CFSCRIPT>
```

The next rule – more of a guide than law – is that if you're using a value inside a tag, especially CFSETs and all of the CFIF family, you probably don't need pound signs. Before I go into detail, let me explain my use of CFSET or variable declarations inside CFSCRIPT.



As everything you create in ColdFusion is typeless (although 4.5 is starting to change that a bit), try to treat values for CFSET as if they were typed. In other words, put strings inside quotes and leave the quotes off for numbers.

```
<CFSET variables.varName =
  "MyString">
<CFSET variables.varName = 5>

<CFSCRIPT>
 variables.varName = "MyString";
 variables.varName = 5;
</CFSCRIPT>
```

Although they're both really typeless, treating them in this fashion creates a good foundation for my explanation of pound signs.

The first tag I'll attack is CFSET. If you're passing in a value regardless of scope (url, form, query, variables, attributes, etc.), you don't need pound signs, despite its value as a string or a number. Think of this as assigning the value of an object to the variable.

```
<CFSET variables.varName =
  queryName.columnName>
<CFSET variables.varName =
  form.formField>
<CFSET variables.varName =
  url.myPrimaryKey>
```

This is also true when using functions within CFSET.

```
<CFSET variables.varName =
DateFormat(queryName.columnName,
  "mm/dd/yyyy">
<CFSET variables.varName = Trim
  (queryName.columnName)>
```

The same can also be said when concatenating a string. If you're "assembling" a string, make your code look that way.

```
<CFSET variables.varName =
 queryName.columnName & "myLiteral-
  String" & TRIM(form.formfield)>
```

For comparison, let's take a look at how you'd use an object and string together in JavaScript.

```
<SCRIPT>
 var x = "someValue";
 alert(x + " someString");
/* You concatenate the object and the
```

```
string to show an alert box that says
     someValue someString
*/
</SCRIPT>
```

The next greatest offender of pound sign overuse is CFIF, and for the most part the same rules apply.

```
<CFIF IsDefined("form.backOrdered")>
<CFIF Len(form.firstName) LT 5>
<CFIF queryName.columnName GT
form.formField>
<CFIF queryName.columnName EQ 5
<CFIF form.formField EQ "someString">
```

The lack of pound signs is immediately evident, but don't forget that the values passed into these CFIFs were treated as if they were typed (numbers or strings). The second example compares form.firstName to a numeric value. I could have placed double quotes around the 5 and it would have worked just as well. It's confusing with the quotes; when you look at that statement without the quotes, it's obvious I'm treating it like a number. The only time I use double quotes is when I want to compare the object against a literal string. On a side note, the last comparison would have evaluated faster by using CompareNoCase.

Up to now these rules have been fairly straightforward. However, before you go ripping out all your pound signs and double quotes, keep in mind that the majority of tags are designed to work with double quotes. In fact, some tags, depending on their use, will throw an error if you don't use double quotes around them. For those tags you must use pound signs within the quoted tag attributes.

```
<CFQUERY DATASOURCE="#session.
  dataSource#">
<CFMODULE TEMPLATE="somePage.cfm"
attributeName="#queryName.
  columnName#">
<CFOUTPUT QUERY="#variables.
  varName#">
```

The previous examples should give you a good idea of where you can and can't get rid of pound signs. I'd be surprised if this article doesn't raise a little controversy, so if you have an opinion, I'd love to hear it.

Before I close this article, I also want to mention the use of double quotes and single quotes. Strings are normally identified within double quotes, as in many other languages. If for some reason you need to qualify a string within a quoted area, you'd use single quotes. The same is true with ColdFusion. When using a function, use double quotes around its parameters; when making a comparison, use double quotes as well. Save the single quotes for qualifiers within a string.

```
<CFSET variables.myVar =
  "myStringValue">
DateFormat(Now(),"mm/dd/yyyy")
NOT
<CFSET variables.myVar =
  'myStringValue'>
DateFormat(Now(),'mm/dd/yyyy')
```

Using a single quote within double quotes does come in handy, especially with CFFILE tags. Just remember, 99% of the time use the double quotes. Let the debate begin!

ABOUT THE AUTHOR

Richard Schulze is an MCSE/MCT working for Spider Technologies in San Diego, California. He has been working with ColdFusion since the version change from 1.0 to 1.5.

RICK.SCHULZE@SPIDERDIRECT.COM

DigitalNation

www.dedicatedserver.com

Sending e-Mail with ColdFusion

Create an extensible Web-based application that uses only one CF template

BY
JERRY
BRADENBAUGH

You've seen it. You've probably done it, too. I'm talking about those Web sites that have a Contact Us link at the bottom of the page with something similar to mailto:us@company.com in the HREF attribute. They're kind of nice...

...and can spare you the burden of opening your e-mail client. They spawn a new message and populate the To: field and perhaps the Subject: field. The mailto: protocol can also be used in the ACTION attribute of the <form> tag. Pretty cool.

Great as the mailto: protocol is, it has shortcomings. Sometimes users don't have their e-mail client software configured correctly, so the link or form submission fails. Anyone who's installed MSIE on more than one occasion can probably attest to this. Plus, users have to confirm every e-mail sent via form submission. That's a necessary annoyance. Also, what if you want to capture more specific information about the user sending the e-mail? Sometimes a Subject: and Message: field just won't cut it.

That's what this article is about – creating an extensible, ColdFusion, Web-based e-mail application that uses only one CF template. This application offers several useful features; you'll see how to add and change them when you find a flavor that meets your needs. All this information will be in a zipped archive on the *CFDJ* site. Barring minor one-line code changes, the code is ready to use. Here are the features the application will cover:

FIGURE 1 Banner graphic for opening interface

- ***Basic e-mail:*** Forms-based e-mail we all know and love
- ***Handling multiple recipients:*** Grabbing names and addresses from a database and selecting multiple recipients
- ***Keeping user input:*** Storing submitted info in a database for later crunching
- ***Better-looking messages:*** Adding HTML and graphics to your messages
- ***Send attachments:*** Adding the code for sending attachments

ColdFusion makes incorporating these and other features quite easy.

Introducing MailRat

Figure 1 offers the debut of MailRat, the hippest new e-mail app ever to be powered by ColdFusion (let's not vote, though). As awesome and mind-bending as I've made it out to be, MailRat still consists of the following simple form elements:

- Fields for your name and e-mail address
- A SELECT list to choose the recipient's e-mail address(es)
- A field to enter the topic you want to talk about
- A text area to input your message
- A file input field for attachment
- Radio buttons for creating custom layouts for your messages

The process is simple. Just enter the required information, then choose Go MailRat!. ColdFusion calls this same file, index.cfm, and changes the banner image on top to confirm that your message has been sent. Figure 2 shows the change in graphics. From the user's perspective there isn't much to it. Let's drill down a bit and see how it's all happening.

Send e-Mail, Print the Form or Both

index.cfm performs at least one operation and sometimes two. If the page is loaded for the first time, CF simply displays the MailRat interface. If the form is submitted, index.cfm needs to handle the e-mail operation, then print the interface. Let's assume this is the first time the page has loaded. In that case we'll be looking more at HTML than CFML. Listing 1, which is long but simple, contains the HTML output to create the interface.

You won't find many surprises in that listing. It's really a form embedded in a couple of nested tables. The CFML in this portion of the document shows up when generating the SELECT list. Here it is up close:

```
<select name="TheirEmail"
style="width:258;" size="3" multiple>
 <CFOUTPUT QUERY="GetContacts">
   <option
value="#EmailAddress#">#UserName#
 </CFOUTPUT>
 </select>
```

The SELECT list is the product of a database query that occurred earlier in the execution. Here is the <CFQUERY>:

```
<CFQUERY NAME="GetContacts" DATA-
SOURCE="ContactDB">
  SELECT UserName, EmailAddress
  From Contacts
</CFQUERY>
```

The SQL inside this statement requests all usernames and corresponding e-mail addresses from the ContactDB data source, which is an MSAccess database. The SELECT list has OPTIONs with usernames as their text and e-mail addresses as their values. Notice, too, that the SELECT list contains the MULTIPLE attribute, i.e., all values of the selected options will be submitted to the server for processing.

That's all there is to printing the form, except for the graphic at the top. If the MailRat is loading for the first time, ColdFusion displays the default banner. If mail was just sent, CF displays the mail-sent banner. This happens via a simple

Allaire

www.allaire.com

<CFIF> statement based on the value of the name entered. When the page loads for the first time, the default value of #YourName# is an empty string that cues CF to load the default banner.

Creating and Validating the Message

MailRat allows you to choose the type of layout you want the e-mail message to have. I used simple layouts, which basically changes the font family of the text. No reason why you can't go wild and exotic, though. Just make sure whoever receives this has an e-mail client that supports HTML. Most recent clients do, so you should be fine. You also have the option of choosing a file to send as an attachment; just click the button and choose the file from your directory tree. The only drawback is you lose any HTML layout capabilities you may have selected.

What if you want to send the message to several people? After all, the SELECT list has the MULTIPLE attribute. In this case the code is no different. Remember that the value of #TheirEmail# is a comma-separated string, the same e-mail syntax for sending messages to multiple people. Choosing three people from the list would represent the value of #TheirEmail# to person1@somewhere.com, person2@somewhere.com and person3@somewhere.com. That's how the string will arrive when the script is called and how it'll be sent, as the value of the TO attribute.

FIGURE 2: Banner graphic for opening interface

Notice that clicking on the radio buttons dynamically changes the image to match the mock-up of the font each layout will have. This article is about ColdFusion, but we should go over the JavaScript because there's so much interaction between the two. The image swap occurs when using the following code:

```
function swapImg(idx) {
    document ['LayoutImg'].src =
newImgs[idx].src;
    }
```

It only takes three lines of code – not too bad. This function changes the source of the "Layout" image in the document, which you'll find near the radio button. When called, swapImg() expects an integer that'll reference an Image object with the corresponding JPEG image. Each radio button has the following code:

```
onclick="swapImg(some_Integer);"
```

Since there are only three images, the integer is either 0, 1 or 2. These integers reference an array element in the JavaScript array newImgs, which is created with a single call to function preload().

```
 function preload() {
  fExt = '.jpg';
  newImgs = new Array();
  var imgArr = new Array('Boring',
'OK', 'Fancy');
  for (var i = 0; i <
imgArr.length; i++) {
    newImgs[newImgs.length] = new
Image();
    newImgs[newImgs.length - 1].src
= imgArr[i] + fExt;
    }
  }

 preload();
```

This function preloads the images for the script. Image preloading is the process of downloading images and storing them in the browser cache before they're displayed. This may not seem like a big deal, but each time the user clicks a radio box for the first time, the browser normally makes an HTTP request back to the server to grab the image. If the images are loaded in the cache, the browser already has them and won't need to do the extra legwork.

The array newImgs contains the Image objects with all the information about each JPEG. By the way, if you want to add more images for preloading, and they're JPEGs in the same folder as the others on the Web server, just add the first name of the image file as a string to the imgArray. For example, adding EvenFancier.jpg to the group changes the array as follows:

```
var imgArr = new Array('Boring',
'OK', 'Fancy', 'EvenFancier');
```

Okay, the images are preloaded. The user has filled in the goods and selected an HTML layout option. Choosing Go MailRat! sets things in motion. Before sending everything back to the server, it's a good idea to get JavaScript involved again to make sure the user has provided enough information. Function checkMail() gets the call.

```
function checkMail(f) {
    if((f.YourName.value == '') ||
(f.YourEmail.value == '') ||
      (f.TheirEmail.value == '') ||
(f.Message.value == '') ||
      (f.YourTopic.value == '')) {
      alert('You must complete all
fields.');
      return false;
      }
    else if
(isValidEmail(f.YourEmail.value) &&
      isValidEmail(f.TheirEmail.value)) {
      }
    else { return false; }
    }
```

This function first determines if any of the fields (except the optional file attachment field) have been left empty. If so, the user is alerted and the form submission ID canceled. Afterwards, checkMail() relies on isValidEmail() to make sure the e-mail address appears valid. I say *appears* valid because the actual code for validating e-mail addresses, according to the Internet standard, is thousands of characters long. For our purposes, as long as the user enters something, such as someone@somewhere.com, everything will be fine. If the e-mail address fails this test, the user is again alerted and the submission canceled. Otherwise, it's back to the Web server with the form.

```
  function isValidEmail(str) {
    str += '';
    namestr = str.substring(0,
str.indexOf("@"));  // before the '@'
    domainstr =
str.substring(str.indexOf("@")+1,
str.length); // after the '@'
     if ((namestr.length == 0) ||
(domainstr.indexOf(".") <= 0) ||
      (domainstr.length == 0) ||
(domainstr.indexOf("@") != -1)) {
      alert('Hmmm.... "' + str + '"
appears to be an invalid e-mail
address.');
      return false;
      }
     return true;
    }
```

Allaire

www.allaire.com

Back to ColdFusion

The form is submitted and index.cfm is called again, but this time things are different. All the CFML we glossed over at the beginning of the article comes into play. CFML uses a single <CFIF> statement to determine whether to send e-mail (see Listing 2).

The deciding factor for sending e-mail is the value of the variable #YourName#. If it's empty, the odds are there's no reason to send mail. As long as #YourName# isn't an empty string, CF will attempt to send e-mail. Once the script has determined that e-mail needs to be sent, it must check if the user uploaded a valid file for use as an attachment. This is done by not only checking the value of #Attachment#, but ensuring that the file exists. If both of the following conditions are true, the attachment will be sent with the plain text message:

```
<CFIF #Attachment# NEQ "" AND #File-
Exists(Attachment)# EQ "True">
```

If all is well, the following code gets the message where it needs to go.

```
<CFMAIL
  FROM="#YourEmail#"
  TO="#TheirEmail#"
  SUBJECT="#YourTopic#"
  SERVER="pop.serve.com"
  MIMEATTACH="#Attachment#"
  >
#YourName# wrote:

#Message#
```

<CFMAIL>

This is the <CFMAIL> tag, which makes our job of sending e-mail ridiculously easy. Though we used only five of the attributes, let's take a look at all of them.

- ***TO*** *(required):* This attribute expects the recipient's e-mail address(es). You can provide a static value or a ColdFusion variable. Multiple recipient addresses need only be separated by commas.
- ***FROM*** *(required):* This attribute expects the sender's e-mail address.
- ***CC*** *(optional):* This attribute can be set to one or more recipients to whom you'd like to send a copy. Can be set static or as a CF variable.
- ***SUBJECT*** *(required):* This is the subject heading of the message and can be set to a static value or a CF variable.
- ***TYPE*** *(optional):* This value indicates the MIME type of the message. It can only be set to a value of "HTML." If omitted, the message will be generated as plain text.
- ***QUERY*** *(optional):* This attribute indicates the name of a database query performed with <CFQUERY> and can be used to print the results of a query in the message.
- ***MAXROWS*** *(optional):* Indicates the maximum number of e-mail messages to send.
- ***MIMEATTACH*** *(optional):* This attribute specifies the name of the path to a file attachment. The value can be static or a ColdFusion variable.
- ***GROUP*** *(optional):* Used to group a set of query results by column name.
- ***STARTROW*** *(optional):* Specifies which row to begin with in the query result set (the first row is 1 by default).
- ***SERVER*** *(required):* Specifies the name or IP address of the SMTP server to contact in order to send the message. If this attribute isn't set, the script uses the SMTP server specified in the ColdFusion Administrator. For more information see the "Configuring a Default SMTP Server" section at the end of this article.
- ***PORT*** *(optional):* This attribute indicates which TCP/IP port the SMTP listens to for requests. The default is 25.
- ***MAILERID*** *(optional):* This attribute can specify a string in the SMTP header to identify the application. The default value is Allaire ColdFusion Application Server.
- ***TIMEOUT*** *(optional):* This indicates the number of seconds to wait for a response from the SMTP server.

LISTING 1

```
<html>
<head>
  <title>MailRat!</title>
  <style type="text/css">
  <!--

  TD
    {
      font-family: Arial;
      font-size: 11px;
    }

  //-->
  </style>
<script language="JavaScript1.2">
<!--

function checkMail(f) {
  if((f.YourName.value == '') || (f.YourEmail.value == '') ||
    (f.TheirEmail.value == '') || (f.Message.value == '') ||
    (f.YourTopic.value == '')) {
    alert('You must complete all fields.');
    return false;
    }
    else if (isValidEmail(f.YourEmail.value) &&
      isValidEmail(f.TheirEmail.value)) {
      }
    else { return false; }
    }

  function isValidEmail(str) {
    str += '';
    namestr = str.substring(0, str.indexOf("@"));  // before the '@'
    domainstr = str.substring(str.indexOf("@")+1, str.length); //
after the '@'
     if ((namestr.length == 0) || (domainstr.indexOf(".") <= 0) ||
      (domainstr.length == 0) || (domainstr.indexOf("@") != -1)) {
      alert('Hmmm.... "' + str + '" appears to be an invalid e-mail
address.');
      return false;
      }
     return true;
    }

  function swapImg(idx) {
    document ['LayoutImg'].src = newImgs[idx].src;
    }

  function preload() {
    fExt = '.jpg';
```

Our <CFMAIL> tag identifies the sender, recipient, topic, SMTP server and file attachment value. The user provides the script with everything except the SMTP address. What about the message? It's placed between the opening and closing <CFMAIL></CFMAIL> tags. Notice there's no HTML, and there is a handful of carriage returns surrounding the #YourName# and #Message# variables. Since the e-mail will be sent as plain text, you have to do the formatting yourself. What you see is what you get.

If the user hasn't uploaded a valid file attachment, the script uses a slightly different <CFMAIL> tag (see Listing 3).

Since there isn't a valid file attached, CF can utilize the HTML attribute of the <CFMAIL> tag to output a message layout. It all depends on the value of the #Layout# variable, set when the user clicked on the selected radio button. The text in #YourName# and #Message# are still printed as before, but they're wrapped in some HTML.

Once CF communicates with the SMTP server specified in the SERVER attribute and sends the message, why not store some info about the transaction? The following <CFQUERY> statement inserts a record into a waiting database.

```
<CFQUERY DATASOURCE="ContactDB">
  INSERT INTO SentMessages
    (DateStamp, SenderName, SenderE-
mail, Recipient, Topic, Message,
    Layout, Attachment)
  VALUES
(#CreateODBCDateTime(Now())#, '#Your-
Name#', '#YourEmail#',
    '#TheirEmail#', '#YourTopic#',
'#Message#', '#Layout#',
    '#Attachment#')
</CFQUERY>
```

The DATASOURCE attribute is set to ContactDB, an MSAccess database included in the ***CFDJ*** download. The embedded SQL statement inserts a new record into the database, storing basically everything the message sender entered, plus the date and time the script ran by calling CFML functions #CreateODBCDateTime(Now())#. If you're interested in the sender's browser information and IP address, you could easily insert the CGI variables #HTTP_USER_AGENT# and #REMOTE_SERVER#.

Wrapping Up

After the e-mail is sent and the data is stored, the remaining code outputs the mail form to the browser. That's about it. Of course, you can change this application to suit your needs ad infinitum. One of the things you might want to change is the ability to enter any e-mail address without having to choose from a list. No problem. Change the SELECT list to INPUT TYPE=TEXT. Make sure you keep the same name of TheirEmail. Don't worry about validating the recipient's e-mail address before the form is submitted. The JavaScript function checkMail() already does it for you.

Configuring a Default SMTP Server

Notice that <CFMAIL> utilizes the SERVER attribute. You can set this value to an SMTP server. However, the CF Administrator makes it shamelessly easy to set a default SMTP server. That way you don't have to use the SERVER attribute every time you want to send mail. Here's how:

1. Log into your CF Administrator.
2. Choose Mail under the miscellaneous section.
3. Enter the address of your SMTP server.
4. If you want (but probably not), modify the port and time-out settings.
5. Choose Apply.
6. If you want to (probably so), choose Verify.

That's all there is to it. If verification fails, check the SMTP server to confirm that it's available and running.

ABOUT THE AUTHOR

Jerry Bradenbaugh works as a senior Web application developer and project lead in Los Angeles. He focuses on robust Web-enabled systems using JavaScript, ColdFusion, Java and database technologies. He's also the Webmaster of HotSyte – The JavaScript Resource (www.serve.com/hotsyte).

hotsyte@mail.serve.com


```
    newImgs = new Array();
    var imgArr = new Array('Boring', 'OK', 'Fancy');
    for (var i = 0; i < imgArr.length; i++) {
      newImgs[newImgs.length] = new Image();
      newImgs[newImgs.length - 1].src = imgArr[i] + fExt;
      }
    }

  preload();

  //-->
  </script>
  </head>
  <body>
  <CFIF #YourName# NEQ "">
    <img src="mailratsent.jpg">
  <CFELSE>
    <img src="mailratwelcome.jpg">
  </CFIF>
  <form action="CustomLayout.cfm" method="post" onsubmit="return
checkMail(this);">
  <table border="1" cellspacing="1">
    <tr>
      <td>
      Your Name
      </td>
      <td>
      <input type=text name="YourName" value="" size="35">
      </td>
      <td rowspan="6">
      <table>
        <tr>
          <td width="75">
          <input type=radio name="Layout" value="Boring"
onclick="swapImg(0);">
          Boring
          <br>
          <input type=radio name="Layout" value="OK"
onclick="swapImg(1);"
          checked> OK
          <br>
          <input type=radio name="Layout" value="Fancy"
onclick="swapImg(2);">
          Fancy
          <br>
          </td>
          <td>
          <img src="OK.jpg" name="LayoutImg">
          </td>
        </tr>
        <tr>
          <td colspan="2">
```

```
        HTML formatting is not available when sending attachments
        </td>
    </table>
    </td>
  </tr>
  <tr>
    <td>
    Your E-mail
    </td>
    <td>
    <input type=text name="YourEmail" value=" " size="35">
    </td>
  </tr>
  <tr>
    <td>
    Their E-mail
    </td>
    <td>
    <select name="TheirEmail" style="width:258;" size="3" multiple>
    <CFOUTPUT QUERY="GetContacts">
      <option value="#EmailAddress#">#UserName#
    </CFOUTPUT>
    </select>
    </td>
  </tr>
  <tr>
    <td>
    Your Topic
    </td>
    <td>
    <input type=text name="YourTopic" value="" size="35">
    </td>
  </tr>
  <tr>
    <td>
    The Message
    </td>
    <td>
    <textarea name="Message" rows="10" cols="30"></textarea>
    </td>
  </tr>
  <tr>
    <td>
    Attachment
    </td>
    <td>
    <input type=file name="Attachment" size="35">
    </td>
  </tr>
  <tr>
    <td>
     
    </td>
    <td>
    <input type=submit value="Go MailRat!">
         
  <input type=reset value="    Clear    ">
  </td>
 </tr>
</table>
</form>
</body>
</html>
```


On-Line Data Solutions, Inc.

www.CoolFusion.com

Worldwide Netfast

www.worldwidenetfast.com

LISTING 2

```
<CFIF #YourName# NEQ "">
  <CFIF #Attachment# NEQ "" AND #FileExists(Attachment)# EQ "True">
    <CFMAIL
      FROM="#YourEmail#"
      TO="#TheirEmail#"
      SUBJECT="#YourTopic#"
      SERVER="pop.serve.com"
      MIMEATTACH="#Attachment#"
      >
#YourName# wrote:

#Message#

  </CFMAIL>
  <CFELSE>
    <CFMAIL
      FROM="#YourEmail#"
      TO="#TheirEmail#"
      SUBJECT="#YourTopic#"
      SERVER="pop.serve.com"
      TYPE="HTML"
      >
    <CFIF #Layout# EQ "Boring">
      <font face="Courier">
    <CFELSEIF #Layout# EQ "OK">
      <font face="Arial">
    <CFELSEIF #Layout# EQ "Fancy">
      <font face="Impact">
    </CFIF>
    <h2>#YourName# wrote:</h2>
    <br><br>
    #Message#
    </font>
    <br><br><br>
    </CFMAIL>
    </CFIF>
```

LISTING 3

```
<CFELSE>
  <CFMAIL
    FROM="#YourEmail#"
    TO="#TheirEmail#"
    SUBJECT="#YourTopic#"
    SERVER="pop.serve.com"
    TYPE="HTML"
    >
   <CFIF #Layout# EQ "Boring">
     <font face="Courier">
    <CFELSEIF #Layout# EQ "OK">
      <font face="Arial">
    <CFELSEIF #Layout# EQ "Fancy">
      <font face="Impact">
    </CFIF>
    <h2>#YourName# wrote:</h2>
    <br><br>
    #Message#
    </font>
    <br><br><br>
  </CFMAIL>
</CFIF>
```

CODE LISTING

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

SiteHosting.net

www.sitehosting.net

Creating Custom Tags for ColdFusion Express

Using custom tag implementation to create powerful Web applications

BY
RAYMOND
CAMDEN

In the summer of 1999 Allaire released the first beta of a new product called "ColdFusion Express." It was designed to be a lightweight version of the ColdFusion Application Server. Since it's free, some of the more powerful features weren't included.

But this doesn't mean you can't create powerful Web applications! One of the biggest features missing from CF Express is support for custom tags. In this article I'll describe a system that will allow you to *fake* the custom tag implementation on the CF Express server. It won't be perfect, but hopefully it'll allow you to get some of the benefits of custom tags without too much extra work.

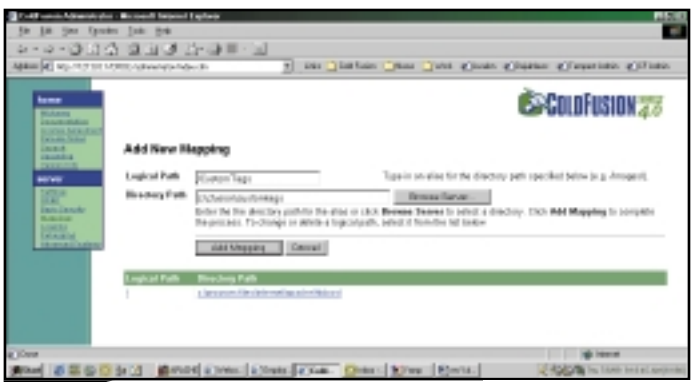


FIGURE 1: CF Express Administrator

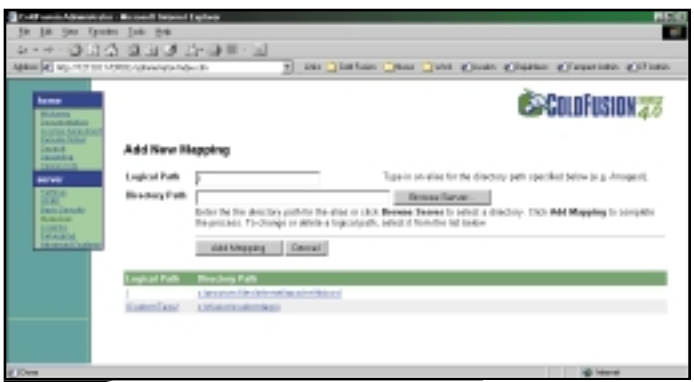


FIGURE 2: Mappings page with new mapping

Let's begin by going over some of the major benefits of custom tags. The main reason we use them is code reuse, which refers to the ability to use portions of code multiple times. For example, you may have written a CFML template that calculates the distance between two longitude and latitude points. This is a complex function that you don't want to rewrite or copy every time you need it. If you did, you'd have to dig through all your files to correct any mistakes or add new features. Custom tags allow you to specify CFML code in one file and use it in multiple areas.

Another powerful feature of these tags is that they execute in their own environment, which means that when we call a custom tag the CF server creates a separate memory space for its variables, i.e., if your CFML template uses a variable called "Jedi" and your custom tag uses a variable with the same name, the variables won't overwrite each other. The "Jedi" variable inside the custom tag is a completely separate variable.

Both features can be implemented with CF Express. We have to go a bit out of our way to do so, but it's worth it in the long run. Currently, I spend a lot of time writing custom tags, sometimes more than I normally would. But I know it's worthwhile since my code will be more portable. I think you'll find the steps we take in CF Express aren't that painful and will allow you to reap some of the benefits of "real" custom tags.

Creating a Custom Tags Directory

When you install the ColdFusion Application Server, a folder called "CustomTags" is created. When you use custom tags, the server looks in that folder to find your code. Under CF Express we're going to fake this functionality by using the CFINCLUDE tag with a mapping we're going to make with the CF Express Administrator (see Figure 1). We've entered a logical name for our mapping, CustomTags (use whatever name you want), and have mapped it to a folder under the cfusion folder where CF Express was installed. We could've picked any folder, including one underneath the Web root, but I want this mapping to be as close as possible to the implementation under the CF Application Server. Figure 2 shows the Mappings page after we've added the new mapping. (Don't forget to physically create the folder we just added a mapping to!)

Now that we have our mapping, let's try a simple test. The following code shows a simple CFM file that includes a custom tag we'll call TheDate.

```
<HTML>

<HEAD>
<TITLE>Example 1</TITLE>
</HEAD>

<BODY>

Welcome to our Web page.
The current date is:

<CFINCLUDE TEMPLATE="/CustomTags/TheDate.cfm">

</BODY>
</HTML>
```

This code is very simple. The CFINCLUDE statement is all you need to pay attention to. Notice how we use the /CustomTags shortcut? Because of this we don't have to worry about the physical location of TheDate.cfm; we just use the mapping we set up in the Administrator. Our custom tag is shown in the following code. Be sure to save this file in the folder you created earlier.

```
<!---
 TheDate Custom Tag
 Last Updated: July 8, 1999
--->

<CFOUTPUT>#DateFormat(Now())#</CFOUTPUT>
```

Again, the code here is pretty simple. In our custom tag we simply wrap the Now() function with the DateFormat function. This returns the Web page shown in Figure 3.

Designing a Tag Scope

So far, all we've done is set up a CFINCLUDE path. Let's tackle another feature of custom tags – the custom tag scope. As you know, normal custom tags work in their own scope. This means that any variables defined in the tag won't overwrite values in the calling template. To read or set values in this template, you have to access them via the Caller scope. When we use CFINCLUDE, however, the code included will be executed in the same scope as our document. It's impossible for the tag to accidentally overwrite variables we've already defined.

Since we don't have a custom tag scope to work with, we need to create our own – in this case the CFE scope (short for ColdFusion Express). If you develop custom tags for ColdFusion Express, you may want to follow the same format. If enough people do it, we can create a directory of custom tags similar to the large directory of existing tags for ColdFusion Server. Here's some example code to create this scope.

```
<CFIF NOT
IsDefined("Request.ThisTag")>
 <CFSET Request.ThisTag = ArrayNew()>
</CFIF>
<CFSET
ArrayAppend(Request.ThisTag,Struct-
New())>
```

What's going on here? One way we try to avoid messing with the calling document is to work in the Request scope. We start off by creating an array, which I named "ThisTag" (after the imaginary tag we're working with), to store our data. The next line creates a new structure and appends it to the end of the array.

What's the point of the array? If we call our custom tag more than once, using the array allows us to "remember" our old data. We could write a custom tag and call it multiple times; the tags called later in the document can then access the data the earlier tags used. What happened to our simple CFE scope? Let's add one more line to our code.

```
<CFIF NOT
IsDefined("Request.ThisTag")>
<CFSET Request.ThisTag = ArrayNew(1)>
</CFIF>
<CFSET
ArrayAppend(Request.ThisTag,Struct-
New())>
<CFSET Request.CFE =
Request.ThisTag[ArrayLen(Request.This
Tag)]>
```

Our new line, <CFSET Request.CFE = Request.ThisTag[ArrayLen(Request.ThisTag)]>, creates the CFE scope we'll use in the tag. When working with structures, if you use the equality operator (=), you're not creating a copy but a reference. This means that when I set data in the Request.CFE scope, it's setting it in the array of structures we created in the Request scope. Request.CFE is short and won't be too hard to type as we work on our custom tag.

Now that we have a tag scope, let's create a tag and test it out. This tag is a modification of TheDate that works with a few variables in the tag scope:

```
<!---
 TheDate Custom Tag (2)
 Last Updated: July 8, 1999
--->
<CFIF NOT IsDefined("Request.The-
Date")>
 <CFSET Request.TheDate =
ArrayNew(1)>
</CFIF>
<CFSET
ArrayAppend(Request.TheDate,Struct-
New())>
<CFSET Request.CFE = Request.The-
Date[ArrayLen(Request.TheDate)]>

<CFSET Request.CFE.CurrentTime =
Now()>
<CFSET Request.CFE.PrettyDate = Date-
Format(Request.CFE.CurrentTime,"mmmm
d, yyyy")>
<CFSET Request.CFE.PrettyTime = Time-
Format(Request.CFE.CurrentTime,"hh:mm
tt")>

<CFOUTPUT>#Request.CFE.PrettyDate# at
#Request.CFE.PrettyTime#</CFOUTPUT>
```

The first few lines of code we simply cut and pasted from the example above. The only difference is that I renamed "ThisTag" to "TheDate," the name of my custom tag. The real changes begin when I start working with variables I want to keep local to the tag. Notice our first new line: <CFSET Request.CFE.CurrentTime = Now()>. By working with the Request.CFE scope, I won't be overwriting any of the variables in my calling document. It's a bit more work, but I'm willing to take the extra time. This is something I do even when working with "real" custom tags. It's not always easy to take a particular function or process

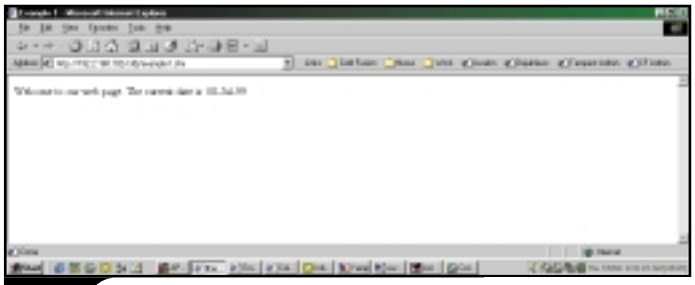


FIGURE 3: Wrapping the Now() function with the DateFormat function that returns this Web page

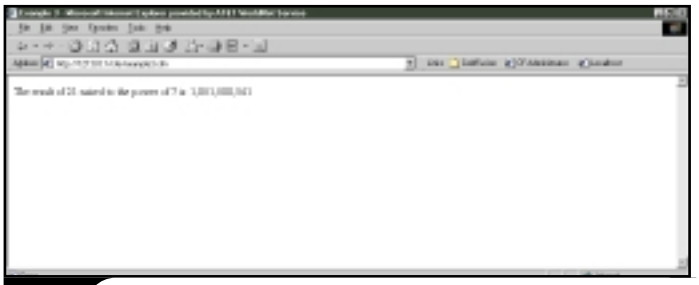


FIGURE 4: The Attributes tag in action

and abstract it into a custom tag. But the extra work you do to make your code portable ends up paying off in the end. I believe the same will be true here. To see an example of this tag in action, run the code included in this example:

```
<HTML>

<HEAD>
<TITLE>Example 2</TITLE>
</HEAD>

<BODY>

 Welcome to our Web page. The current date and time is:

<CFINCLUDE TEMPLATE="/CustomTags/The-Date2.cfm">

</BODY>
</HTML>
```

Passing Attributes to Our Custom Tags

The only thing missing from our custom tag implementation is a way to pass attributes to our tags. Normal custom tags can accept attributes by passing in name-value pairs as in the example below:

```
 <CF_Foo Attribute1="Value1"
Attribute2="Value2">
```

Attribute1 and Attribute2 were passed to the custom tag, Foo. Inside the tag these values can be accessed with the Attributes scope, Attributes.Attribute1, which would evaluate to "Value1."

> If your entire development team adheres to this standard, your entire workplace can benefit

So how are we going to implement this in our tags? We're going to agree on a "standard" and stick with it, as we did before. In this case the standard is pretty simple. Before calling our custom tag, we can create our own Attri-butes scope that our tag can access in the same manner as normal custom tags.

Various bits of code in your calling template…

```
<CFSET Attributes = StructNew()>
<CFSET Attributes.Base = 10>
<CFSET Attributes.Power = 3>
<CFINCLUDE
TEMPLATE="/CustomTags/XtoY.cfm">
```

In the preceding example code, we start off by creating an Attributes structure. We set two keys in it (Base and Power), then call our custom tag. It's important to note that each time you call a custom tag you should reinitialize your Attributes structure with the <CFSET Attributes = StructNew()> line. If you don't, a tag could accidentally read the Attributes of a previous tag. The following code example shows the source of XtoY.cfm and the use of the Attributes scope.

Allaire
www.allaire.com

Neo Tech
www.cf-training.com

Allaire

www.allaire.com

SHOP ONLINE AT JDJSTORE.COM FOR BEST PRICES OR CALL YOUR ORDER IN AT 1-888-303-JAVA

JDJStore.com

GUARANTEED BEST PRICES FOR ALL YOUR JAVA RELATED SOFTWARE NEEDS

Take a look at our specials this month!

EASTLAND DATA SYTEMS

Internet Shopping with Java Shopping Cart

…Described as the most progressive and interactive form of shopping on the web today…This Java Applet provides a complete user interface package for Internet Shopping Web Sites. Using Java technology we produce a drag-and-drop shopping user interface that is fun and easy to use, encouraging shoppers instead of frustrating them with confusing controls that are hard to follow. And the easier it is to shop, the more you sell.

$294.99

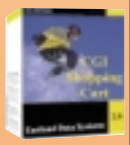
Hybrid Shopping Cart

This Java Applet provides a complete user interface package for Internet Shopping Web Sites. A "Hybrid" is defined as an offspring of two varieties. A blending of the best features from our CGI and Java shopping products, we took the most powerful aspects of Java technology; real-time, on-screen updating and computational capabilities. And combined those with the most desirable features of our CGI shopping Cart, namely it's flexibility and compatibility with web designers with artistic talent.

$294.99

CGI Shopping Cart

The Shopping Cart automates the Shopping Process to make shopping on your site intuitive, straight forward, and enjoyable! It's one of the most affordable Shopping Carts because it was designed for small businesses. Specifically for entrepreneurs who are testing the Internet waters, and can't or don't want to make large investments into bells and whistles for their site. But simply want to make shopping on their site easy for the customer.



$294.99

Guaranteed Best Prices

JDJ Store Guarantees the Best Prices.
If you see any of our products listed anywhere at a lower price, we'll match that price and still bring you the same quality service.

Terms of offer:

- Offer good through October 31, 1999
- Only applicable to pricing on current versions of software
- August issue prices only
- Offer does not apply towards errors in competitors' printed prices
- Subject to same terms and conditions

Prices subject to change.
Not responsible for typographical errors.

Attention Java Vendors:
To include your product in JDJStore.com, please contact jackie@sys-con.com

ALLAIRE

ColdFusion 4.0

With ColdFusion 4.0, create Web applications for self-service HR solutions, online stores, interactive publishing and much more. It's the integrated development environment that has all the visual tools you need to create Web applications quickly and easily. From simple to sophisticated ColdFusion gives you the power to deliver the Web solutions you need – faster, and at a lower cost.

ColdFusion Studio 4.0 . . . . . $354.99
SkillBuilding with ColdFusion Interactive Training CD $284.99

ALLAIRE

HomeSite 4.0

HomeSite is the award-winning HTML editing tool that lets you build great Web sites in less time, while maintaining Pure HTML. Unlike WYSIWYG authoring tools, HomeSite gives you precise layout control, total design flexibility and full access to the latest Web technologies, such as DHTML, SMIL, Cascading Style Sheets and JavaScript. HomeSite 4.0 is the only HTML editor featuring a visual development environment that preserves code integrity.

HomeSite 4.0 . . . . . $87.99

ALLAIRE

JRun - Live Software

JRun is the industry-leading tool for deploying server-side Java. JRun is an easy-to-use Web server "plugin" that allows you to deploy Java Servlets and JavaServer Pages. Servlets form the foundation for sophisticated server-side application development. Java servlets are platform independent, easy to develop, fast to deploy, and cost-effective to maintain.

JRun $558.99

INSTALLSHIELD

InstallShield Java Edition 2.5

InstallShield Java Edition 2.5 is the powerful tool developers require to produce bulletproof InstallShield installations with Java versatility. You can target your application for multiple systems with cross-platform distribution. And InstallShield Java Edition 2.5 offers the key features and functionality designed to let developers go further in distribution and deployment.

InstallShield Java Edition $474.99

PROTOVIEW

Java Enterprise Editions

The Java Enterprise Editions offer you a choice between comprehensive packages of award-winning AWT or JFC components along with enterprise-level support and subscription service. Powerful, extendable, lightweight components built on the foundation of JFC, the ProtoView JFCSuite contains JFCDataCalendar, JFCDataExplorer and JFCDataInput. The JSuite (AWT) includes the DataTableJ grid component with JDBC and Visual Café database support. Also includes TreeViewJ, CalendarJ, TabJ and WinJ.

JSuite Enterprise Edition $818.99
JFC Enterprise Edition $818.95
JSuite $328.99
JFCSuite $408.99

KL GROUP

JProbe Suite

JProbe Profiler is the most powerful tool available for finding and eliminating performance bottlenecks in your Java code. JProbe Coverage makes it easy to locate individual lines of untested code and reports exactly how much of your Java code has been tested. JProbe Threadalyzer lets you pinpoint the cause of stalls and and deadlocks in your Java applications and makes it easy to predict race conditions that can corrupt application data.

JProbe Profiler w/ Standard Support (inc. JProbe Memory Debugger) . . $464.99
JProbe Coverage w/ Standard Support $464.99
JProbe Threadalyzer w/ Standard Support $464.99
JProbe Suite w/ Standard Support $934.99

GALILEO DEVELOPMENT SYSTEMS

Intr@Vision Foundation

Intr@Vision Foundation helps bring ColdFusion development to the next level. It provides an out-of-the-box application security architecture for handling your most complex intranet and extranet needs. Instead of spending 30% of your development time adding security to every application you build, it gives you a proven solution with a single line of code. Intr@Vision Foundation allows your developers to focus on building business solutions, not infrastructure.

Intr@Vision Foundation $3499.99

CATOUZER

Synergy SOHO

Synergy is a ColdFusion-based Web application framework for instantly deployable corporate intranets. It consists of an Application Services Layer, which offers central security and administrative services, and eight core collaborative applications. Synergy's open architecture is designed for implementing existing ColdFusion applications and developing new ColdFusion applications specifically tailored to the customer's needs.

Synergy SOHO $499.99

ORDER ONLINE WWW.JDJSTORE.COM

```
<!---
 XtoY Custom Tag
 Last Updated: June 8, 1999
--->

<CFIF NOT IsDefined("Request.XtoY")>
 <CFSET Request.XtoY = ArrayNew(1)>
</CFIF>
<CFSET
ArrayAppend(Request.XtoY,Struct-
New())>
<CFSET Request.CFE =
Request.XtoY[ArrayLen(Request.XtoY)]>

<CFSET Request.CFE.X =
Attributes.Base>
<CFSET Request.CFE.Y =
Attributes.Power>
<CFSET Request.CFE.Result =
Request.CFE.X ^ Request.CFE.Y>

<CFOUTPUT>#NumberFormat(Request.CFE.R
esult)#</CFOUTPUT>
```

Like our previous custom tag, our first four lines (after the comments) set up a scope for our tag variables. The next few lines take the two attributes and set them in the CFE scope. Then we do some simple math (the ^ operator takes the value on the left side and raises it to the power of the value on the right side) and output it using the NumberFormat function. The template that calls this custom tag and passes in the Base and Power Attributes is shown in the following code.

```
<HTML>

<HEAD>
<TITLE>Example 3</TITLE>
</HEAD>

<BODY>

 The result of 21 raised to the
  power of 7 is:

<CFSET Attributes = StructNew()>
<CFSET Attributes.Base = 21>
<CFSET Attributes.Power = 7>
<CFINCLUDE
TEMPLATE="/CustomTags/XtoY.cfm">

</BODY>
</HTML>
```

The only difference between this example and the earlier ones is the use of the Attributes tag. Admittedly, it's not as nice as being able to pass in attributes directly to the tag, but it works. The tag expects to find Base and Power and uses these values to generate a result. Of course, if these two attributes aren't passed, the tag will fail. A good custom tag writer would add in checks to make sure the tag is called in the proper manner. Figure 4 shows an example of this run in the browser.

Where Do We Go from Here?

A lot of what I've described here can be perceived as "hacks" or workarounds. But let's think about it for a minute. Allaire has created a great tool that's absolutely free. ColdFusion Express can easily be used to generate powerful Web applications. In this article I've described a series of workarounds that, if performed consistently, can be used to incorporate custom tags and most of their powerful features. Best of all, if your entire development team adheres to this standard, your entire workplace can benefit from the wonders of custom tags…at

ABOUT THE AUTHOR

Raymond Camden is a senior developer at Creative Internet Solutions (www.creativeis.com), the Web application development firm of Control Data. Raymond is a member of Team Allaire and an Allaire Certified Instructor. Raymond has contributed numerous custom tags to the Developer's Exchange and is one of the authors of "Mastering ColdFusion 4" from Sybex.

JEDIMASTER@CREATIVEIS.COM

Statement of Ownership

CFDev.com

www.cfdev.com

Infoboard

www.infoboard.com

ADVERTISING INDEX

ADVERTISER	URL	PH	PG
ABLE SOLUTIONS	WWW.ABLECOMMERCE.COM	360.253.4142	2
ALLAIRE	WWW.ALLAIRE.COM	888.939.2545	27,37,39,47
BIZNIZ WEB	WWW.WEBPUBLISHINGTOOLST.COM	281.367.4016	50
CATOUZER	WWW.CATOUZER.COM	604.662.7551	55
CFDEV.COM	WWW.CFDEV.COM	800.854.7838	49
COMPUTERJOBS.COM	WWW.COMPUTERJOBS.COM		3
CONCEPTWARE AG	WWW.CONCEPTWARE.COM	061.964.7320	4
CYSCAPE	WWW.CYSCAPE.COM	800.932.6869	52
DATARETURN	WWW.DATARETURN.COM	800.767.1514	17
DIGITAL NATION	WWW.DEDICATEDSERVER.COM	703.642.2800	35
EKTRON	WWW.EKTRON.COM	603.594.0249	15
ENTERPRISE USA	WWW.ENTERPRISEUSA.COM	248.888.1473	46
EPRISE	WWW.EPRISE.COM	800.274.2814	19
FINDADEVELOPER.COM	WWW.FINDADEVELOPER.COM	440.257.6690	51
FINDAHOST.COM	WWW.FINDAHOST.COM	440.257.6690	33
FUNDERE SOFTWARE	WWW.FUNDERE.COM	310.392.1005	13
FUSION FX	WWW.FUSIONFX.COM	800.780.2422	24
GALILEO	WWW.GALILEODEV.COM	770.643.9176	9,
HOSTPRO	WWW.HOSTPRO.NET	888.638.5831	23
INFOBOARD	WWW.INFOBOARD.COM	800.514.2297	49, 51
INFORUM SOLUTIONS	WWW.INFORUMSOLUTIONS.COM	978.887.0080	11
INTERMEDIA	WWW.INTERMEDIA.NET	650.424.9935	56
NEO TECH	WWW.CF-TRAINING.COM	973.540.9672	46
ON-LINE DATA SOLUTIONS	WWW.COOLFUSION.COM	516.737.4668	42
ORIGENT	WWW.ORIGENT.COM	617.623.8044	31
PIRANHANET	WWW.PIRANHANET.COM	206.374.0880	21
RSW SOFTWARE	WWW.RSWSOFTWARE.COM	508.435.8000	25
SITEHOSTING.NET	WWW.SITEHOSTING.NET	888.463.6168	43
VIRTUALSCAPE	WWW.VIRTUALSCAPE.COM	212.460.8406	26
WORLDWIDE NETFAST	WWW.NETFAST.NET	800.362.9004	42

Able Solutions

Enter the realm of browsable store building and administration – from your browser. Build "your_site.com" with secure Merchant Credit Card Processing. Maintain inventory, add discounts and specials to keep your customers coming back. Increase sales with cross selling and membership pricing.

11700 NE 95th Street, Suite 100, Vancouver, WA

www.ablecommerce.com • 360 253-4142

Catouzer, Inc.

With the Synergy 1.0 Web application framework, creating custom intranet applications is a breeze. The Synergy Application Development Kit (ADK) gives you the tools to rapidly develop your custom applications, which can be fully integrated and managed under the Application Services Layer (ASL).

1228 Hamilton Street, Suite 501, Vancouver, B.C. V6B 2S8, Canada

www.catouzer.com • 604 662-7551

CFDEV.com

CFDEV.COM is fine-tuning Site-a-Matic,. its soon-to-be-released Web Developer's Toolkit. Site-a-Matic is an application management program that allows developers to quickly deploy applications, reducing development time from hours to minutes. In one week CFDEV.COM received over 300 downloads and plans to add new custom tags each week. Free tags are available at the site as well.

www.cfdev.com 1-800-791-1916

Computerjobs.com

ComputerJobs.com is a leading Internet-based job search company dedicated to helping computer and information technology (IT) professionals find great jobs. Our user–friendly, proprietary Web site provides IT job seekers and hiring companies a convenient, effective way to connect. In addition to thousands of job listings for candidates, we provide companies who list jobs with us access to résumés of the hottest IT professionals on the Web.

3200 Windy Hill Road, Suite 700 West, Atlanta, GA 30339

www.computerjobs.com

Conceptware

Conceptware provides professional solutions and first-class products for the I*Net. With the launch of Spectra, the new Content Management and e-commerce framework by Allaire Corporation, Conceptware has announced its Internet suite, GateBuilder, which is based on Spectra.

Industriestraße 30-34, Eschborn, Germany 65760

www.conceptware.com +49-(0)6196-47 32-0

Data Return Corporation

Data Return offers extensive support for customers utilizing ColdFusion from Allaire. With customers delivering over 50,000 user sessions per day, we know how to provide high availability solutions for this advanced application server. Our technical support staff has extensive experience in coding custom ColdFusion Tags as well as managing Microsoft SQL server. We also support CyberCash for customers interested in real-time credit card processing. Our combination of support and experience offer an ideal environment for deploying your applications developed for ColdFusion.

801 Stadium Dr., Ste 117, Arlington, TX 76011

www.datareturn.com • 800 767-1514

digitalNATION - a VERIO company

digitalNATION, VERIO's Advanced Hosting Division, is the world's leading provider of dedicated server hosting, with over 1,650 servers online today. dN's superior connected network and service abilities have earned dN a solid reputation as a first-choice provider of dedicated server solutions (Sun, Windows NT, Linux and Cobalt). digitalNATION has been providing online and network services for over six years. One of the first ISPs to provide dedicated servers running Microsoft Windows NT, the dN staff has unparalleled experience in this industry.

5515 Cherokee Ave, Alexandria, VA 22312-2309

www.dedicatedserver.com • 703 642-2800

Ektron

Ektron supports the next-generation needs of e-businesses by providing dynamic Web infrastructure solution tools designed for use by nontechnical staff. Ektron's flagship offering, eContentManager, gives staff members across an organization the hands-on ability to make real-time additions and updates to Web content without requiring knowledge of a programming language -- while still allowing for centralized administrative control and security. With competitive advantages such as ease-of-integration and drag & drop everything, Ektron is looking to provide these empowering products to customers, resellers and integrators.

5 Northern Blvd., Suite 6, Amherst, NH 03031

www.ektron.com • 603-594-0249

FINDaDEVELOPER.com

- Post & Search Job Listings
- List Your Web Development Services For Free

www.findadeveloper.com

Enterprise USA, Inc.

Enterprise USA is a software product and computer consulting firm based in the Metropolitan Detroit area of Michigan, with offices in Farmington and Lansing. Enterprise USA specializes in providing computer consultants for Internet development, Linux and the Microsoft product line. In particular they provide supplemental staffing of computer programmers, help desk professionals, network integration specialists, and project managers.

33425 Grand River Ave, Suite 201, Farmington, MI 48335
www.enterpriseUSA.com • 248 888-1473

Eprise Corporation

If your customers are looking for a content management solution, Eprise Participant Server FastStart Kit for Allaire ColdFusion developers can save you time and resources. Participant Server is a flexible-content management framework that enhances high-value business relationships through the delivery of timely, targeted, Web-based communications.

1671 Worcester Road, Framingham, MA 01701
www.eprise.com 800 274-2814

Fundere Software

Fundere Corporation, a leading ColdFusion developer, offers software solutions that enable customer service organizations to respond, track and analyze high volumes of customer inquiries. Fundere's SupportAgent software has a built-in natural language engine that allows customers to enter their queries using plain English. Built on top of ColdFusion, the SupportAgent can leverage any existing Web application or legacy database.

18 Horizon Avenue,. Venice, CA 90291
www.fundere.com 310 392-7805

Fusion FX, Inc.

Fusion FX, Inc., is a full-service Web site design and hosting company. Our strength in ColdFusion Development, as well as with custom AbleCommerce Store building, and the fact that we have a Team Allaire member on staff will give you peace of mind because you know your ColdFusion development is in good hands.

819 Peacock Plaza, Suite 535, Key West, FL 33040
www.fusionfx.com • 800 780-2422

Galileo Development Systems

The Intr@Vision family of products provides the basis for automating your business processes over the web using a standard, extensible framework. With a focus on reducing the cost of doing business and improving cash flow, Intr@Vision Time and Intr@Vision Expense offer a web-based alternative to your current paper processes. Contact us today for more information on our products.

2695 Long Lake Dr., Roswell GA 30075
www.galileodev.com • 770 643-9176

HostPro

HostPro offers packages to suit just about anyone's growing bandwidth, storage and feature requirements. Our web site hosting solutions are packaged to meet specific needs based on popular criteria. This includes UNIX and NT packages, e-commerce solutions that range from simple to turn-key, databases like mSQL to mission-critical MS SQL, multimedia like RealAudio, RealVideo and all the programs and languages to extend your site's capabilities

3250 Wilshire Blvd., Suite 707, Los Angeles, CA 90010
www.hostpro.net • 213 252-9779

Highlight your website with
infoboard
NT and UNIX
Cold Fusion Hosting
Development Consulting
Oracle, Informix, MS SQL, E-Commerce Plug-ins
1-800-514-2297
sales@infoboard.com
www.infoboard.com
<allaire> Alliance Partner

Get Trained. Get A Job.

WWW.CF-TRAINING.COM

JOBSEEKERS:
Search through hundreds of listing from dozens of companies & recruiters at:
www.coldfusionjobs.NET

COLD FUSION TRAINING:
Search for a facility near you, view course schedules, or purchase books & materials.
www.cf-training.com

Looking for Cold Fusion Developers?

ADD US TO YOUR BOOKMARKED SITES!
We offer unlimited job listings and many other services.Mention promotion code 1102X

WWW.COLDFUSIONJOBS.NET

INFORUM Solutions

INFORUM Virtual Office Park integrates communication, data collection, knowledge management, office management, and planning tools in a single Web-based system. With it any nontechnical individual or organization can create custom, interactive, fully functional Web sites in a matter of hours, not weeks! Link virtual offices to form virtual office parks. You and your most far-flung colleagues can collaborate more effectively, manage projects more successfully, and achieve goals more quickly.

458 Boston Street, Topsfield, MA 01983
www.inforumsolutions.com 800 442-0599

Intermedia, Inc.

Our advanced virtual hosting packages (powered by Microsoft Windows NT and Internet Information Server 4.0) offer an environment supporting everything today's advanced Web developer or sophisticated client could ask for. Complete ODBC support is available on plans B and C. We support Microsoft Index Server on all hosting plans.

953 Industrial Avenue, Suite 121, Palo Alto, CA 94303
www.intermedia.net • 650 424-9935

NetFast

NetFast strives to become the foremost leader in Web application software. Our applications empower and enable function-building efforts of Web developers and designers. These ready-to-use software components can be bundled into existing Intranet and Internet sites to increase usability, drive traffic and differentiate online offerings. The fusion of application development and hosting provides substantial revenue opportunities for VARs and integrators of NetFast products.

6699 Port West Drive, Suite. 130, Houston, TX 77024
www.netfast.net 800 362-9004

On-Line Data Solutions

CoolFusion.com is dedicated to providing unique and powerful add-on solutions for ColdFusion development and implementation. The site is hosted and maintained by On-Line Data Solutions, Inc. – a leader in ColdFusion integration. Our ColdFusion integration products line is called inFusion – a combination of "infuse" and ColdFusion. For information about our flagship product, inFusion Mail Server, we invite you to read the online information (where you can also download the latest beta) and join the inFusion Mail Server support list.

24 Elm Street, Centereach, NY 11720-1706
www.coolfusion.com 516 737-4668

PiranhaNet

PiranhaNet offers complete Web site and information design services. From full–featured Web presences and Web-based document management systems to streaming video and e-commerce, PiranhaNet can deliver a total solution. Our work speaks for itself. Tour a virtual gallery of our featured projects, and learn how PiranhaNet helps clients succeed at www.piranhanet.com/services.htm.

2112 6th Avenue, .Seattle, WA 98121
www.piranhanet.com 206 374-0880

RSW Software

RSW Software is a wholly owned subsidiary of Teradyne, Inc., and specializes in Web application testing software. Established with the goal of providing best-in-class testing products, RSW offers a suite of products called the e-TEST Suite, which automates the process of testing business-critical Internet and intranet applications.

44 Spring Street, Second Floor, Watertown, MA 02172
www.rswsoftware.com • 508 435-8000

Sitehosting.NET

Successful electronic commerce starts at SiteHosting.net; a division of Dynatek Infoworld, Inc., which provides total Web development services. We offer personal and efficient customer service with reliability at value prices. All our plans include access to SSL (Secure Socket Layer). We support ColdFusion, Active Server Pages, Real Audio/Video, Netshow Server, and more. Our hosting price starts at $14.95/month.

13200 Crossroads Parkway North, Suite 360, City of Industry, CA 91746
www.sitehosting.net • 877 684-6784

Virtualscape

Why host with Virtualscape? Nobody else on the Internet understands what it takes to host ColdFusion like we do. Virtualscape is the leader in advanced Web site hosting. From Fortune 500 extranets to e-commerce sites and more, developers recognize our speed, stability, reliability and technical support.

215 Park Avenue South, Suite 1905, New York, NY 10003
www.virtualscape.com • 212 460-8406

CFDJnews

Visual Mining Selects Allaire's ColdFusion Server

(Rockville, MD, and Cambridge, MA) – Visual Mining, Inc., a leading provider of enterprise corporate decision portals, and Allaire Corporation announced that Visual Mining has selected Allaire's ColdFusion server to power Visual Mining's corporate portal product, Decision.Control. www.visualmining.com and www.allaire.com.

Ektron Realigns Pricing

(Amherst, NH) – Ektron, Inc., has reduced pricing for its eWebEditPro, eContentManager and eWebBuilder products, making them available to organizations seeking next-generation Web development and content-management solutions. For more information go to www.ektron.com.

Allaire Announces ColdFusion 4.5 for Linux

(Boston, MA) – Allaire Corporation released a new version of its cross-platform Web application server, ColdFusion. As one of the first commercial application servers for Linux, ColdFusion 4.5 gives the rapidly growing Linux community a powerful new platform for building and delivering enterprise and e-commerce systems. www.allaire.com.

Macromedia Releases Authorware 5.1

(San Francisco, CA) – Macromedia, Inc., introduces Macromedia Authorware 5.1, an upgrade to its authoring software for deploying learning applications for the Web, networks and CD-ROM. The upgrade is free to registered users of Macromedia Authorware 5. For further information go to www.macromedia.com.

Allaire JRun to Power In-Store Information Kiosks for Home Depot

(Cambridge, MA) – Home Depot, the world's largest home improvement retailer, has chosen Allaire JRun to power its in-store information kiosks. "We were looking for a solution that would enable us to deliver product and vendor information quickly and efficiently to our customers via our in-store information kiosks." said Ron Griffin, CIO of Home Depot. For more information on JRun go to www.allaire.com.

WebTrends Launches Enterprise Reporting Server Version 2.0

(Portland, OR) – WebTrends Corporation has released Enterprise Reporting Server 2.0. This version provides customers with advanced reporting features to help turn e-business intelligence into results. With the release of an NT edition, Enterprise Reporting Server is a true cross-platform Web traffic analysis solution operating on Linux, Solaris and Microsoft Windows NT platforms. For more information or to download a free trial version see www.webtrends.com/products/ers.

Cyscape

www.cyscape.com

Career Oppt

www.sys-con.com

A Platform for the **21st Century**

Allaire's e-business platform promises profound changes

BY **ADAM BERREY**

In 1523 Spanish conquistador Francisco Pizzaro conquered the Inca army and captured the emperor, Atahuallapa, effectively securing the conquest of the entire Inca Empire in a single battle. Remarkably, Pizzaro won the battle with a ragtag team of 168 Spanish soldiers on foreign soil against the 80,000-man Inca army. Pizzaro's astounding victory wasn't the result of a particularly strong military strategy (or moral character).

He was able to overcome enormous numerical odds with a superior technology platform. In his case the technologies were domesticated horses, steel weapons and guns. Pizzaro's victory underscores the awesome advantage created by new and better technology platforms.

While Allaire isn't planning to equip a new wave of conquistadors, we are putting together a new technology platform that will give ColdFusion developers a distinct advantage over the competition. During the past year, Allaire launched one major new product (Allaire Spectra) and acquired two industry-defining companies – Bright Tiger and Live Software. Now we're putting all of these technologies together as an integrated e-business platform.

The easiest way to understand the benefits of this platform is to see it as a set of interconnected services for handling all of the challenges associated with developing Web applications. Each component builds on the other, making the whole stronger while giving you the flexibility to pick and choose the pieces you need.

The Allaire E-Business Platform

It's valuable to take a moment and review the different products and how they work together. The platform has four major components.

- ***Application Server:*** Not surprisingly, at the heart of the Allaire e-business platform are services provided by the ColdFusion Web application server. You use these every day already, from state management and security to database connectivity. The core services in the ColdFusion server provide a foundation for any e-business application, and the application server services in the Allaire platform are quickly expanding with the new JSP and Servlet technology provided by JRun.
- ***Packaged Systems:*** On top of ColdFusion, Allaire Spectra 1.0 delivers six new higher level services for content management, personalization, roles-based security, workflow and process automation, business intelligence and syndication. Accessed through some 300 new tags, these services make it significantly easier and faster to build large-scale content and e-commerce applications.
- ***Visual Tools:*** To productively develop using the services in ColdFusion and Allaire Spectra, the Allaire platform includes HomeSite, ColdFusion Studio and the Spectra Webtop (a very cool browser-based interface for working with Spectra services).
- ***Web Systems Management:*** The final set of services (yet to be shipped) are management services for application deployment and replication, server monitoring, real-time server maintenance, notification and operational reporting. Based on Bright Tiger, these services will make it easier to manage highly reliable, large-scale production sites.

Key technologies in all four parts of the platform are available today, and a lot more is coming in 2000.

Moving Forward in 2000

The scope of the services across the Allaire platform is expanding rapidly. In ColdFusion 4.5 there is new support for Java connectivity, XML parsing and service-level failover. The software includes an entirely new project management system for working more effectively with complex applications. It also has a wide range of new editor features that speed up development, such as split-screen editing and collapsible code.

Moreover, ColdFusion 4.5 runs natively on Linux, expanding the range of operating systems available to you. In 2000 the scope of the application server services will expand significantly with the release of new JRun technologies and the increasing integration between JRun and ColdFusion.

On top of the core products a range of free software development kits (SDKs) and open source technologies are going to make it easier to take advantage of cutting-edge application strategies. For building distributed Wwb applications that dynamically exchange data between servers, there is a new release of the Web Distributed Data Exchange (WDDX) SDK. Now available on www.wddx.org, the WDDX 1.0 SDK expands support for binary objects in WDDX and adds new functionality that makes it even easier to build distributed Web applications with XML.

The fastest-growing new category of Web browsers, the new microbrowsers, will be embedded in most new commercial cell phones in 2000. There are currently about 300 million active cell phones, many more than PCs, and this market continues to grow. The new wireless Web application SDK that Allaire has developed with Phone.com will give you the sample code and tools you need to build ColdFusion applications that can be deployed to microbrowsers.

Because we believe that working with dynamic content is so central to any commerce or enterprise Web application, we're releasing a free SDK for the Spectra content object API (COAPI). The COAPI sits at the heart of the Spectra architecture and gives you a free, incredibly powerful set of tools for working with dynamic content.

While perhaps not as dramatic as the invention of firearms or the domestication of horses, the development of the Allaire e-business platform promises profound changes. For me, there's no better reminder of the impact than the 1,200 developers who attended the first Allaire developer conference. From applications that launch public dot.com companies to systems that revolutionize the operations of established firms, ColdFusion developers are literally inventing the new Internet economy. And we've only just begun.

ABOUT THE AUTHOR

Adam Berrey was part of the original founding team at Allaire. For the last four years he has led the ColdFusion product management and marketing effort. As director of product marketing, he is responsible for setting product strategy and coordinating product management and marketing for ColdFusion, JRun and other infrastructure product initiatives.

ABERREY@ALLAIRE.COM

Catouzer

www.catouzer.com

NT WEB HOSTING

Intermedia.NET

TAKE control of your Web Site

TEST DRIVE FREE DEMO ACCOUNT

Unlimited Email accounts

Unlimited traffic/bandwidth

New! Clockwork™ Mail
Manage email from your browser

FOR FREE SET UP
USE PROMO CODE CFDJ

NT Hosting with IIS 4.0
MS SQL Server 7.0, Access, FoxPro
Active Server Pages
Front Page 98 and 2000
Cold Fusion 4.01
SSL Secure Server
Shopping Carts: StoreFront, Drumbeat ECommerce
CyberCash, PaymentNet
WebTrends Statistics

Reseller Program Available

Instant Control!

Add domains
Add disk space
Add email accounts
Add mailing lists
Register Cold Fusion Custom Tags
Register ODBC
Register Active X DLL and OCX

sign up online
and activate your account in 10 minutes

www.intermedia.net

Intermedia.NET, Inc.
953 Industrial Ave
Palo Alto CA 94303
sales@intermedia.net

800.379.7729

Copyright Intermedia.NET Inc. 1999. All Rights Reserved. Microsoft, FrontPage and Windows NT are registered trademarks of Microsoft Corporation. Cold Fusion, WebTrends, CyberCash and all other trademarks are property of their respective holders.