HOW HEALTHY IS YOUR COLDFUSION SERVER? PG 7

COLDFUSION Developer's Journal

ColdFusionJournal.com

May 2005 Volume:7 Issue:5

COLDFUSION PREVENTS NUDITY

How a high school student launched a T-shirt business thanks to CF PG 26

PLUS...

macromedia®

One little box.
A whole lot of power.

Put it to work for you.
The shortest distance between you and powerful web applications.

Full speed ahead. The release of Macromedia ColdFusion MX 7 is changing the whole game. This groundbreaking release introduces new features to help address your daily development challenges. These features include:

› Structured business reporting? Check. Printable web content? Check.

ColdFusion MX 7 provides a structured business reporting solution that will have you creating detailed business reports for anyone who needs them. You can also dynamically transform any web content into high-quality, printable documents with ease. The days of needing a third-party application to generate dynamic reports are going, going, gone.

› Make rapid J2EE development a reality.

So, you're heavily invested in J2EE but would love to complete projects in less time? ColdFusion MX 7 is your answer: It delivers RAD productivity on top of the power and scalability of J2EE and deploys onto standard J2EE server environments.

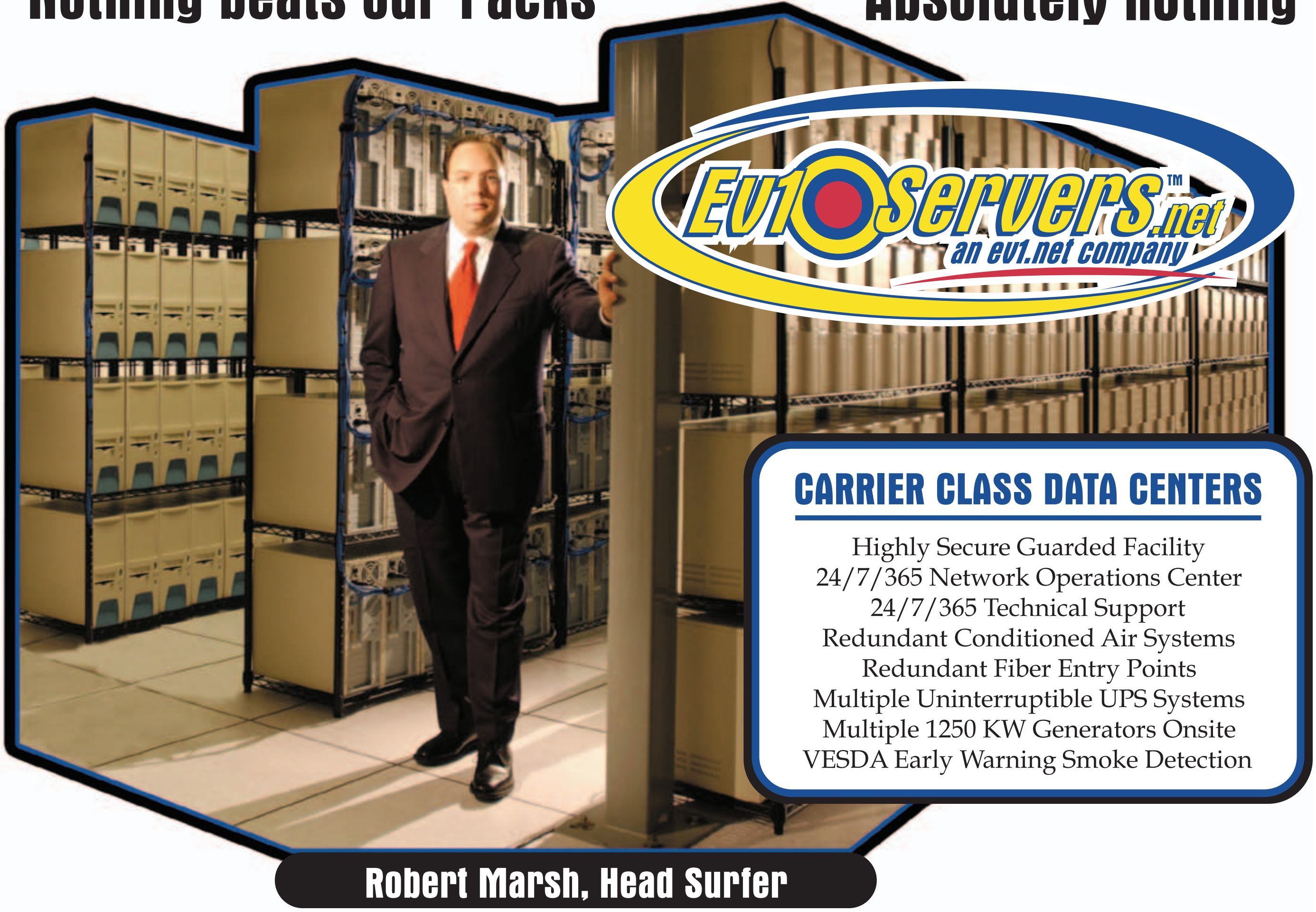
› New mobile applications are a go.

Innovative new features enable ColdFusion MX 7 applications to reach beyond the web. So you can rapidly create new applications, or extend existing ones, to connect with mobile phones and IM clients using a variety of protocols. The new world of mobile communications is exciting, and this your invitation to the party.

To learn more or take ColdFusion MX 7 for a test drive, go to:
macromedia.com/go/cfmx7_demo

Nothing beats our racks

Absolutely nothing



Ev1Servers.net™
an ev1.net company

CARRIER CLASS DATA CENTERS

Highly Secure Guarded Facility
24/7/365 Network Operations Center
24/7/365 Technical Support
Redundant Conditioned Air Systems
Redundant Fiber Entry Points
Multiple Uninterruptible UPS Systems
Multiple 1250 KW Generators Onsite
VESDA Early Warning Smoke Detection

Robert Marsh, Head Surfer

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from $299*

Instant Activation!

Dedicated Server

Dual Xeon 2.4 GHz
2 GB RAM • 2 x 73 GB SCSI HD
Remote Console • Remote Reboot
2000 GB Monthly Transfer Included

23,000+ Servers and Growing!

1-800-504-SURF | ev1servers.net

PLESK7
RELOADED
Preferred Control Panel

IP Compliant. Price subject to change. Quantities Limited.
*Per month. Set-Up fees apply. See web site for complete details.

CFDynamics

Dedicated Server Packages Starting at $189/mo.

All dedicated servers include:

- **FREE STATS SOFTWARE!**
- **No long term commitments!**
- **FREE SQL server access!**
- **FREE MAIL SOFTWARE!**
- **Fair and simple pricing!**
- **Optional server maintenance!**

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

macromedia ALLIANCE PARTNER

Paper Thin Partner

BlueDragon Certified V.I.P. Hosting Partner

When your needs have outgrown shared hosting look to CFDynamics for **total freedom.** With dedicated server packages they're not offering an oxymoron; "virtually private" or "virtually dedicated" is NEITHER private nor dedicated. CFDynamics offers a solution that is **100% completely dedicated**. They don't play games with the fake stuff; CFDynamics only offers the real deal. Real Service. Real Satisfaction. Real Value.

Real Freedom.

Visit us online or call to order!

WWW.CFDYNAMICS.COM

1 - 866 - 233 - 9626
866 - CFDYNAMICS

COLDFUSION Developer's Journal

editorial

editor-in-chief
Simon Horwith simon@sys-con.com

technical editor
Raymond Camden raymond@sys-con.com

editor
Nancy Valentine nancy@sys-con.com

associate editor
Seta Papazian seta@sys-con.com

research editor
Bahadir Karuv, PhD bahadir@sys-con.com

production

production consultant
Jim Morgan jim@sys-con.com

lead designer
Abraham Addo abraham@sys-con.com

art director
Alex Botero alex@sys-con.com

associate art directors
Louis F. Cuffari louis@sys-con.com
Richard Silverberg richards@sys-con.com
Tami Beatty tami@sys-con.com
Andrea Boden andrea@sys-con.com

contributors to this issue
Simon Horwith, Jeff Houser, Mike Nimer, Joe Rinehart,
Neil Ross, Jeff Peters, Hal Helms, Andrew Simon,
Steven Forehand, Phil Hulsey, Sam Farmer

editorial offices

SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645
Telephone: 201 802-3000 Fax: 201 782-9638
COLDFUSION DEVELOPER'S JOURNAL (ISSN #1523-9101)
is published monthly (12 times a year)
by SYS-CON Publications, Inc.

postmaster: send address changes to:
COLDFUSION DEVELOPER'S JOURNAL
SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645

Worldwide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ
FOR LIST RENTAL INFORMATION:
Kevin Collopy: 845 731-2684, kevin.collopy@edithroman.com
Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

For promotional reprints, contact reprint
coordinator Kristin Kuhnle, kristin@sys-con.com.
SYS-CON Publications, Inc., reserves the right to
revise, republish and authorize its readers to use
the articles submitted for publication.

"Ladies and gentlemen, the sky is not falling"

By Simon Horwith

At the time of this writing, Adobe has recently announced its plan to acquire Macromedia in an all-stock trade worth billions ($3.4 billion was the estimate at the time of the announcement). The ColdFusion development community has been abuzz with speculation about how this will affect the future of CF – including many doomsday prognosticators suffering from "Chicken Little Syndrome." Ladies and gentlemen, the sky is *not* falling.

At the time of this writing, Adobe has 4,000+ personnel and in excess of $1.4 billion in the bank (cash and cash equivalents). Macromedia has 1,400+ personnel and $341 million in the bank (cash and cash equivalents). Deferred revenue for Adobe is $57 million and $49 million for Macromedia. What does this mean? It means that Macromedia is now going to be part of a much larger organization with a significantly larger number of resources (financial and otherwise) and industry pull. It's a fair assumption that for the ColdFusion Development Team this will mean that they are more likely to be given the opportunity to implement new ideas. It probably also means that they'll be working on making the ColdFusion Server either integrate or interact with existing Adobe server products (possibly with development tools as well).

Many of us have read the acquisition press release (online at http://www.adobe.com/aboutadobe/invrelations/adobe-andmacromedia.html) – which doesn't say much other than the fact that there's going to be an acquisition. Mike Chambers and Kevin Lynch both blogged about the acquisition at http://www.markme.com/mesh/archives/007504.cfm and http://www.klynch.com/archives/000078.html, respectively. There is one document that I have found to be the most informative and insightful regarding the pending acquisition. The document in question is Adobe's "Adobe to acquire Macromedia investor presentation" – available online at http://www.adobe.com/aboutadobe/invrelations/pdfs/ADBE_MACR_Presentation.pdf – I urge everyone to read it. The investor presentation clearly shows Adobe's vision of how Macromedia and Adobe technologies can combine to meet the changing needs of customers and of technology in general.

– continued on page 25

About the Author

Simon Horwith is the editor-in-chief of ColdFusion Developer's Journal *and is the CIO at AboutWeb, LLC, a Washington, DC based company specializing in staff augmentation, consulting, and training. Simon is a Macromedia Certified Master Instructor and is a member of Team Macromedia. He has been using ColdFusion since version 1.5 and specializes in ColdFusion application architecture, including architecting applications that integrate with Java, Flash, Flex, and a myriad of other technologies. In addition to presenting at CFUGs and conferences around the world, he has also been a contributing author of several books and technical papers. You can read his blog at www.horwith.com.*

simon@horwith.com

Brad Lawryk: "Before MX Kollection, I 'thought' I was doing a **pretty good job**, and now ...

Everyone tells me I´m doing a **brilliant** job!"

If you own Dreamweaver,

MX Kollection is a must!

List Database Information
Create dynamic lists
Order records within a list
Insert, Edit, Delete field in list
Automatic navigation bars
Automatic list filter
Sort column by clicking on the header
Automatic row counter
Delete multiple records directly from list
Create master/detail lists

Manage Recordsets Visually
Create, Edit recordsets visually
Visually add tables to query
Contextual menu to refresh fields
Large query management Zoom In/Out
JOINS between tables
Define and apply complex conditions
Automatically generate SQL conditions
Add GROUP BY for aggregated fields
Extract information from multiple tables
Optimized SQL generation
Smart SQL queries for list filters
Automated database introspection

Rapid HTML Form Creation
Generate Insert Record Form
Generate Update Record Form
Insert, Edit or Delete form field
Redirect to page after form submit
Table and CSS form generation

Form Validation and Error Handling
Validate form fields
Rich formats library
Allow custom validation formats
Error handling
Preserve submitted values on error

Upload Files and Images
File Upload
Image Upload
Resize and sharpen image on upload
Show Dynamic Image
Download Uploaded File
Delete file from specified folder
Show If File Exists on Server

Send E-mails
Send e-mail after form submit

Online HTML Editor
Edit HTML content visually
Use your own CSS styles
Upload and manage server images
Format tables and align images

Enhanced HTML Form Controls
Date Picker
Dependent Drop-downs
Editable Drop-down (Combo-box)
Masked Textfield
Numeric Textfield
Dynamic Search
n...1 Dependent field
Smart Date

Security & User Authentication
Login form generator
Login with User Levels
Remember me feature
Encrypted password
Restrict access to page
Generate password with fixed length
Update user password
Send password by e-mail

DREAMWEAVER PRODUCTIVITY TOOL
FOR DYNAMIC WEBSITE DEVELOPMENT

Download a trial version here - http://interaktonline.com/go/MXKollection/

Find out how easy is to use our tools - **save 30%** by using the coupon in the right.

Caution: InterAKT extensions are known to cause a high increase in productivity. For those of you who can't deal with the free time please continue to hand-code or refer to our competitors for help.

MX Kollection
Suite of Dreamweaver extensions
to speed up dynamic web development
Interakt

$100 coupon - 199920002001

Interakt

macromedia approved

macromedia® ALLIANCE PARTNER

http://www.interaktonline.com/

SYS-CON MEDIA

president & ceo
Fuat Kircaali fuat@sys-con.com

vp, business development
Grisha Davida grisha@sys-con.com

group publisher
Jeremy Geelan jeremy@sys-con.com

advertising

senior vp, sales & marketing
Carmen Gonzalez carmen@sys-con.com

vp, sales & marketing
Miles Silverman miles@sys-con.com

advertising director
Robyn Forma robyn@sys-con.com

advertising manager
Megan Mussa megan@sys-con.com

sales & marketing director
Dennis Leavey dennis@sys-con.com

associate sales managers
Dorothy Gil dorothy@sys-con.com
Kim Hughes kim@sys-con.com

sys-con events

president, events
Grisha Davida grisha@sys-con.com

national sales manager
Jim Hanchrow jimh@sys-con.com

customer relations

circulation service coordinators
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com

manager, jdj store
Brunilda Staropoli bruni@sys-con.com

sys-con.com

vp, information systems
Robert Diamond robert@sys-con.com

web designers
Stephen Kilmurray stephen@sys-con.com
Percy Yip percy@sys-con.com

online editor
Roger Strukhoff roger@sys-con.com

accounting

financial analyst
Joan LaRose joan@sys-con.com

accounts payable
Betty White betty@sys-con.com

subscriptions

Subscribe@sys-con.com
Call 1-888-303-5282

For subscriptions and requests for bulk orders,
please send your letters to Subscription Department
Cover Price: $8.99/issue
Domestic: $89.99/yr (12 issues)
Canada/Mexico: $99.99/yr
All other countries $129.99/yr
(U.S. Banks or Money Orders)
Back issues: $12 U.S. $15 all others

How Healthy Is Your ColdFusion Server?

Results of Challenge #1; plus this month's coding contest

By Simon Horwith

Before describing this month's prize and contest, I would like to thank everyone for their submissions to our first contest (***CFDJ*** vol. 7 issue 3). That contest objective was to modify Ray Camden's blog in any way you choose, and you had to take advantage of at least one new feature in CFMX 7.

Announcing the Winners of Challenge #1

The winners are Nahuel Foronda and Laura Arguello from Blue Instant (http://www.blueinstant.com/). They chose to use Flash Forms and Web Services to create a weather pod. I plan to migrate horwith.com to a new server in the near future, and when I do, their pod will be there as well. I also believe Ray Camden is going to make the pod available for download from his site, either as part of the next version of his blog or as a separate add-on to his blog. Congratulations, Laura and Nahuel.

Challenge #3

As many of you know, I spend a huge amount of my "free time" writing code, books, editing ***CFDJ***, and so on. This month I've decided to take one of the ideas (I've currently got about 20 potential projects in my head) for an application that I've been planning to write and make publicly available, and present it to our readers as a challenge.

The contest is to write an API that exposes information that's useful for monitoring the current status of a ColdFusion MX 7 Server and applications. The information will be the sort that's valuable to Administrators and developers. Here are a few examples of type of information I'd like the API to be able to return:

- Current amount of memory in use by the server
- What applications are running on the server
- Current number of sessions per application
- Average size of sessions
- Page response times and hit-rates
- Average number of CFC method calls, longest, shortest, and average cfc call per page, etc.

There are just suggestions – I leave it up to you to determine what the API can return.

Here are a few resources that you might want to look at (and steal code from) to get started:

- The ColdFusion MX 7 server administrator API
- The Debug information files that show request debug info. (they ship with CF)
- Blogs like Pete Freitag's blog at http://www.petefreitag.com/ and Spike's blog at http://www.spike.org.uk/blog/index.cfm
- The session information tracker debug file I wrote (blogged about at http://www.horwith.com/index.cfm?mode=entry&entry=98F4B071-E081-0478-478E7E447580C494)

Though it's not a requirement, extra points go to any entry that also uses their API in an SMS Gateway implementation. I think it'd be terrific if Server Administrators could use their phone to call their CF Server and choose from a menu of options to find out about current server statistics (you could also add options for rebooting the server, clearing applications and sessions, and so on).

The prize this month is a full year of free ColdFusion Hosting from CFDynamics. CFDynamics are my personal preference for hosting – they give you a lot for your money (though in this case it's free!), have excellent support, and are really good about working with customers to give them whatever they need.

There's no hard deadline for this contest – I expect it will end around the middle of July but because some people get their issues in the mail later than others, I don't want to commit to a hard date. I'll blog about the closing date when it is determined, and will also add a comment to this article on the SYS-CON website (http://cfdj.sys-con.com).

Good luck – I look forward to seeing what people submit!

simon@horwith.com

Reading File as from Java

Sometimes using cffile isn't enough – here's how Java can help

By Jeff Houser

I open my e-mail this morning to find that one of my clients had sent me a file for import to their database. The file is comma delimited. Importing it into SQL Server is fairly simple with Data Transformation Services (DTS), however there's a catch.

The file contains a list of employees and EmployeeIDs. The EmployeeIDs are considered super sensitive, and must be encrypted. DTS can't be used to encrypt data before storing it in the database, so I'll need to do something else. The client didn't send me encrypted data, so what do I do?

Dissecting the Problem

In the past we've used a custom tag to encrypt the data with a BlowFish encryption scheme. (Note: This encryption is built into CFMX7, but the client hasn't upgraded yet). I can perform the encryption in ColdFusion. Maybe I can use cffile to read in the information? Let's see, the file has 60,000 records. The thought of loading so much data into memory and then having to parse it via CF does not sound like a fun project. Is there another way? Well, yes there is.

For 99.9% of the tasks that you perform daily, you'll never need to look beyond the functionality that ColdFusion provides to you, right out of the box. However, once in a while you come across something that you just can't do easily (or at all) from within ColdFusion. Thankfully, ColdFusion is now built on top of Java and you have access to the full array of Java-based objects. Random file access can be done in Java, but not from within CF. Random file is just a fancy way to say that you can access any point in the file, usually by providing the character number of the file. Using some specialized methods, we can also read a file line by line, which is the example we'll explore more in this article.

Here is a simplified version of our client file:

```
Last Name, First Name, EmployeeID
Arehart, Charlie, 78934
Helms, Hal, 78903
Horwith, Simon, 54321
Houser, Jeff, 12345
```

The first record in the file is a header record, naming the columns and the order they appear. Following the header row we have lots of data. For simplicity, I've cut out a lot of the information you would probably receive from the client, such as contact information and company position. The goal is to read in one line from the file, process the record, and then repeat until there are no more entries in the file.

There are two Java classes that we need to make use of to be able to read in the file. The first is BufferedReader class, or java.io.BufferedReader. This class is used to read text from any character input stream. Various items can be used as input streams, but in our case we want to use a text file. Full documentation for the BufferedReader class is located here: http://java.sun.com/j2se/1.4.2/docs/api/java/io/BufferedReader.html.

A constructor, or init method, is a way to initialize a Java object, and many Java objects have one or more constructors. There are two constructors of the BufferedReader class. Both accept a Reader object. One also accepts the size of the input buffer stream. For our purposes we can use the default input buffer stream. But, what is a reader object, and where do we get it?

A reader object is an object created from the reader class. Due to the nature of Java, any subclasses of the reader class can also be accepted as constructor argument. Documentation on the reader class is located here http://java.sun.com/j2se/1.4.2/docs/api/java/io/Reader.html, but the subclass of reader that we are really interested in is the FileReader class, located here: http://java.sun.com/j2se/1.4.2/docs/api/java/io/FileReader.html. The FileReader class is designed for reading in information from a file. It has three different constructors, but the one we are going to use accepts a string that is the location of the file.

The process to follow is to create and init the FileReader class. Then create and init the BufferedReader class using the FileReader class. After that, you can call various methods on the BufferedReader class, such as read to get a specified number of characters or readLine to read a full line of text. All this is interesting, but how do we access the Java Classes from within ColdFusion?

Creating the Java Objects with cfobject and CreatObject

There are two ways you can create Java objects with

ColdFusion. One method uses a tag, and the other users a function. They are both similar, so I'll teach you both of them. From within CFML, you can use the cfobject tag; In CFScript the CreateObject tag is your tool for object creation. The attributes for the tag are very similar to the function's parameters. This is the list of parameters:

- ***Type:*** cfobject can be used to create many types of different objects, such as Web services, CFCs, and Java objects. The type attribute is used to specify what kind of object you are creating. Since we are creating a Java object, we'll give it the value of "Java". This is also the first parameter passed into the CreateObject function. It is required in both methods.
- ***Class:*** The class attribute is used to define the name of the Java class that you want to create. There are many Java objects for various different purposes. You can find the complete list here http://java.sun.com/j2se/1.4.2/docs/api/allclasses-noframe.html. The class is the second and final parameter of the CreateObject function. It is required.
- ***Action:*** The action attribute is used to tell the tag to create the object. Its only valid value is "create". It is required for the cfobject tag, but has no parallel in the CreateObject function.
- ***Name:*** The name attribute specifies the name of the variable that will contain the reference to the newly created object. It is required for the cfobject tag. When using the CreateObject function, this value is the one on the left side of the equal sign, similar to a cfset.

To use cfobject create an object of the FileReader class you would use code like this:

```
<cfobject type="Java" class=" java.
io.FileReader" Action="Create"
name="fileReader">
```

To implement the same command using the CreateObject Function, you might use code like this:

```
<cfset FileReader = CreateObject("Java"," java.
io.FileReader")>
```

Or you could execute the function inside of a CFScript block, like this:

```
<cfscript>
 FileReader = CreateObject("Java"," java.
io.FileReader");
</cfscript>
```

All Java objects have a collection of methods associated with them. The names and functionalities of these methods depend upon which object you are creating an instance of. To call a method against an object you must use something called Object Property notation. You are probably already familiar with Object Property notation, because you can use it to reference columns against a query variable, or keys in a structure, or methods in a CFC.

Object Property notation consists of the variable name of the object, followed by a dot, then the property or method you want to access. If calling a method, enclose the argument list in parenthesis. Most Java objects have one or more init methods. One of the init methods on our FileReader object accepts the disk location of the file. You'll probably call it like this:

```
<cfset Result = fileReader.init("C:\test.txt");
```

Since Java does not have named arguments, you cannot use cfinvoke to call methods against a Java object. Many developers prefer to use CFScript when invoking and using objects because the Object Property notation lends itself better to CFScript than to CFML.

Reading the File

Let's take a look at the code:

```
<cfscript>
 fileReader = CreateObject("java", "java.
io.FileReader");
 fileReader.init("C:\\test.txt");
 br = CreateObject("java", "java.
io.BufferedReader");
 br.init(fileReader);

 // Counters for the loop
 TotalNumRecords = 0;
</cfscript>

<cftry>
 <cfloop condition="true">
 <cfscript>
  line = br.readLine();
 </cfscript>
 <Cfset TotalNumRecords = TotalNumRecords + 1>
 <cfoutput>
  Line #TotalNumRecords#: #Line#<br>
  <!---- other processing ---->
 </cfoutput>
 </cfloop>
 <cfcatch type="coldfusion.runtime.
   UndefinedVariableException">
 <!--- this indicates end of file, ok to ignore
   error --->
 </cfcatch>

</cftry>
```

As discussed earlier, the code starts out by creating a FileReader object. The full path name for the FileReader is java.io.FileReader. Then it initializes that object, giving it the location of a file. The next line creates the BufferedReader object. The full path name for the Buffered–Reader object is java.io.BufferedReader. The init method on BufferedReader object uses the fileReader object as input.

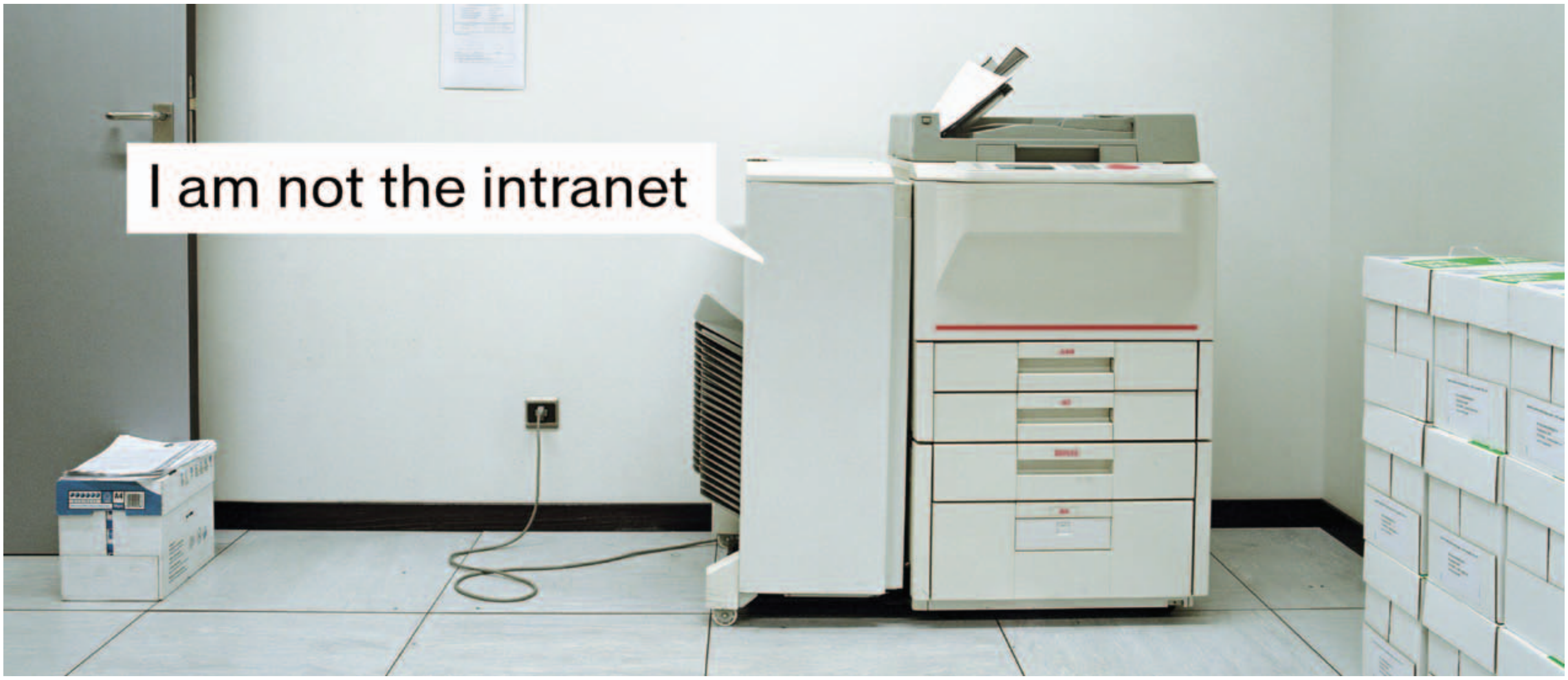
The code does not assign the results of the init operation to any variable, although you could if you wanted. ColdFusion returns the name of the Java object and its location in memory. There isn't much you can do with the information in the context of ColdFusion.

Normally, in Java or other programming languages, you would loop over the input stream until you received an end of file character. Within each loop you would read the next line. CF doesn't have the concept of an end of file character, but we can work around this using the cftry and cfcatch error tags. The code uses a condition loop, with the condition being true. That means the loop will never end. However, the loop is inside a cftry block and the cfcatch is outside of the loop. The catch block will catch an undefined variable exception. When the code in the loop reads the next line and it's not there, this exception will catch when it tries to read the variable.

The code inside the loop is the simple part. It uses the readLine method on the BufferedReader variable to get the next line. It increments counter variable TotalNumRecords and displays the line to the screen. Any additional processing, such as encrypting the data and storing it in the database, is left out of this example, because you know how to do all that.

– continued on page 29

macromedia



There are workarounds for a mediocre intranet, but they still waste time and money.

Meet the Macromedia Web Publishing System. Unlike the average CMS, this works. By allowing business users to publish content, IT bottlenecks are eliminated. Users can simply point and click to update specific pages or sections–while other areas remain protected. So everyone in your organization can share critical information quickly and easily, and you have an intranet that actually lives up to its potential.

Learn how to make your intranet work for you, not the other way around:
macromedia.com/go/webupdate

How to Conquer the Enterprise... with ColdFusion

Exploring the power of CFMX and its object-oriented features

By Jared Rypka-Hauer

Over the course of the last several years, ColdFusion has gone through a time of growth, stress, change, and a gradual increase in complexity, culminating in CFMX 7. That complexity has stretched us as developers, inspiring growth while at the same time causing some of us (especially those without a Computer Science background) to wonder why we should be making all this extra work for ourselves, especially when it comes to object-oriented (OO) development. In some ways, it's like the growth of any other "organism," painful yet critical to the survival of the species.

***Forewarning:** This article uses a lot of OO-specific terminology, because the concept it covers is entirely based in OO design and architecture. While this means that CFMX really is maturing into a powerful Enterprise-class development platform, it also means that we the developers have to run fast and hard to catch up.*

My own journey into object-oriented development began on January 12, 2005, as a result of a particular thread on a mailing list. I had begun to use CFCs for some things and was exhorted to improve my designs or trouble I would see, and it made sense. If I was going to be using a new set of development tools, I reasoned, I wanted to gain all the advantages and cover all the pitfalls. So I started digging and reading, joined some lists, started a blog, and eventually picked up a copy of *Heads Up Design Patterns* for the twin purposes of adapting design patterns to my own applications and learning a bit of Java while I was at it.

It was in Chapter 2 that I encountered the Observer pattern, wherein an observer object is enabled to receive messages by giving a "subject object" permission to programmatically send it messages. I never even finished the chapter. The idea struck a chord in the programmer bean between my ears, and I madly ran to my computer, sat down, and started writing a set of ColdFusion Components (CFCs) that would accomplish this, but for a very particular purpose.

Bringing Publish-and-Subscribe to Messaging

The goal was to enable cross-application messaging, intra-application messaging, and even cross-server messaging by adapting the publish-and-subscribe concept to CFCs and web services. My initial idea was centered around a personal project I've been working on for re-syndicating RSS feeds. The goal was to use an application-scope subscriber CFC to receive syndicated content from a webservice, and then allow the session-scoped CFCs to use the application scope as a content source within the pages. Development work on the idea began about two weeks ago, and it's been through quite a bit over the last few days getting it ready for testing. What I ended up with surprised even me.

Before we get too deep into the details, it is important to make it clear that in the tests, I used a few simple URL variables to control the application, allowing me to observe the changes in the CFC instances as commands were issued. In a real system, these operations would be controlled by application events. The point of the tests was to prove that the concept worked, not to implement a fully-operative production environment. The demo code provided here and as part of the OpenXCF project is not stable, so if you do use it, you use it at your own risk!

The simplest part, it turned out, was writing the actual publisher and subscriber classes. Overall, they're fairly simple objects that have basic methods like update, subscribe, unsubscribe, and so on. The principles are fairly standard object-oriented designs, considering that this is entirely based on the Observer design pattern. In this implementation, however, we can plainly see the power of ColdFusion MX and its object-oriented features.

Publisher.cfc (see Listing 1) maintains a collection of subscribers. In order to register itself with the publisher, a subscriber must provide a reference to itself by calling path.to.publisher.subscribe(this), and an element in "this" must be a UUID as verified by the publisher's call to getUUID() before registering a subscriber. At that point, the publisher adds the subscriber to its structure of registered subscribers. This allows the publisher to either loop over the entire subscribers collection or to access a single subscriber referenced by its UUID.

When the publisher object's controller (in this case, testPubObject.cfc, see Listing 2) wants to send messages to the subscribers, it calls getPublisher().update(messageStruct). The publisher then simply loops over its subscribers structure and on each iteration of the loop it executes subscribers[subscriber].update(messageStruct), effectively using ColdFusion structs as "data packets" and even allowing the system to pass CF "beans" or data-bearing CFCs to the messages array inside each subscriber. Because of the encapsulation of communication between these objects, no object is given the opportunity to interact with objects other than the ones for which communication was intended.

The subscriber objects (see Listing 3) contain an array of structs, each array element representing one message from a publisher. The currently available message handling methods are flush() (which clears the array one element at a time), getMessages(), and setMessage() (which appends an incoming message to the end of the messages array). The intention is to eventually use a method like flush() to provide processing based on the message content before deleting the message from the messages array. Over time, the messages collection may become a separate object, using a queue or a stack to manage the contents of the messages collection rather than diluting the subscriber's functionality.

In that case, message processing would probably be delegated to the subscriber's controller object (testSubObject.cfc, see Listing 4) which will retrieve the messages by calling getSubscriber().getMessages(). This concept of delegation, in OO parlance known as cohesion, is what makes this system powerful and flexible yet secure and stable. If each piece is only able to interact with the objects that the developer allows it to, and all the other parts don't know about the rest of the system, there is little likelihood of any kind of corruption.

The publisher and subscriber CFCs are simple providers, as neither class can function without a composition relationship between itself and its controlling CFC wrapper. In the end, with a publisher in one scope and a subscriber in another scope, we can enable intra-application messaging that we haven't been able to provide in the past without a kludgy interface. Since CFCs have become more and more powerful, and now provide things like protected variable space, we can build secure and stable multithreaded systems that rival any platform out there. In this case, because structures in CFMX are passed-by-reference, and CFCs are internally recognized as a specialized form of structure, it was possible for me to pass a reference to the publisher into the subscriber's subscribe() method, and then use that pointer to pass a reference to the subscriber back into the publisher's subscribers collection.

The system is possible simply because passing structures byRef in CFMX means that setting a variable to equal an existing struct actually means that the new variable points to the same location in memory as the original structure. That way, when data is altered in that region of the server's memory space, any other object or variable pointing to that same location sees the changes immediately. Since this all happens within the memory space of the ColdFusion MX application running on a J2EE server, we can point any variable to any structure, provided that a means to access that memory location is created.

In order to facilitate communication between multiple ColdFusion applications in such a system, there must be a stateful publisher instance in the server scope. In the testing application I wrote, I used the application scope because it was simplest; however, with the new application.cfc provided by CFMX 7, an application can be told to register with a server-scope publisher at application runtime as part of the onApplicationStart event. The implications are staggering, allowing anything from session publishers pushing session event data to the server scope which in turn push data to application-scope subscribers in order to facilitate cross-application logins without using a shared application.cfm file, to allowing an application to push content beans to many session objects asynchronously by issuing one simple method call. The next obvious step is to apply this to web services.

The Web Services Dimension

Web services supply their own unique set of challenges to a system of this nature, specifically the fact that a web service method call is executed on the hosting server, and the need to push content requires a stateful connection to the client's application memory space. In order to work within this set of constraints, I've devised a system wherein the client creates a local instance of a web service and receives access to methods such as getClientFile, subscribe, and unsubscribe. The getClientFile call will return a string that contains the text of a CFC.

The client will then be required to save that file (we'll call it subscriber.cfc for purposes of example) to a web-accessible folder within its own webroot, create an application-scope instance of subscriber.cfc, and then issue a call to webservice.subscribe(urlString), providing the URL of the newly saved CFC as an argument (see Listing 5). Upon receiving the subscribe(urlString) directive from the client, the remote application will create a stateful instance of subscriber.cfc at the URL the client provided and use its one remote method, update(), to push data to the client's application-scope instance of the same class, subscriber.cfc.

Using this method, it is possible to create push-pull CFMX-native web services that can be created programmatically, and that statefully manage data transactions between any level of any number of CFMX-based systems, and that make full use of all the features of CFMX 7. Chief among those new features is application.cfc, which would be used to manage the subscription/unsubscription of any object associated with a publisher. Application.cfc would also be necessary in order to configure multiple applications within a server to check for the existence of a server-scope manager, and to insure that such an instance gets created at application start if need be.

It is also important to note that there is currently no rule against a controller class being able to instantiate both a subscriber and a publisher, thereby enabling the creation of full JMS-style proxied messaging systems that are not limited to any individual application, host, and so on. It is, then, theoretically possible to create a domain model, rather like Windows NT domains, that maintains a trust relationship between any number of systems anywhere in the world.

Enterprise Integration Is No Longer a Dream

I know that the designs I've envisioned for the current platform are just the tip of the iceberg…but Enterprise integration is no longer a dream for us. CFMX suddenly holds the potential to implement everything from remote registries for authentication purposes (something like Enterprise LDAP Messaging, perhaps?); to push-style syndicated content delivery via content packets distributed to any subscribing applications, which in turn provide the content to subscribing session objects.

The system is now part of the OpenXCF project at sourceforge.net. The package has been named Clarion. In addition to the core files, there is a simple proof-of-concept demo that enables the observation of state changes as the system cycles thru its various tasks, which are triggered by clicking on dynamic links. For now, efforts are being concentrated on updating the core files to include complete locking and work out a more complete API for the controller objects. After that will come working out the specifics of the webservice side of things, integrating such tools as authentication, memory structure, and the contents of the subscriber.cfc file that will work as both a local application-scope message container and a webservice API conduit for messages from publisher to subscriber.

I will be keeping a running commentary on things as they progress at my blog, http://www.web-relevant.com/blogs/cfobjective, and the project is available at http://openxcf.sf.net. One of my most significant goals is to roll this up into a re-deployable package and enable the whole CFMX community to participate.

Conclusion

Over the next months, I have every confidence that we'll see a surge of new systems designed to supplement CFMX and empower CFMX developers in ways that, until recently, we've had no facility deal with. These new developments should enable us to put to shame all those platforms that can't compete with the speed and power of CFMX's development cycles. By leveraging the cost of development against the things that we're now capable of using CFMX and its Java underpinnings to accomplish, we find ourselves poised, like never before, to completely conquer the Enterprise.

About the Author

Jared Rypka-Hauer is the founder of Continuum Media Group, LLC. He has recently released his first commercial software -- a developer tool called SQLSurveyor, for database reference and code generation. He's also in the process of releasing a PayPal/ColdFusion integration kit for CFMX. Information on these products and the other services Continuum provides can be obtained on the company's website at www.web-relevant.com. This is his first published article on ColdFusion, and while the use of OO and Enterprise terminology may be a bit rough, he firmly believes that CFMX 7 is the beginning of a brave new world and things can only get better from here.

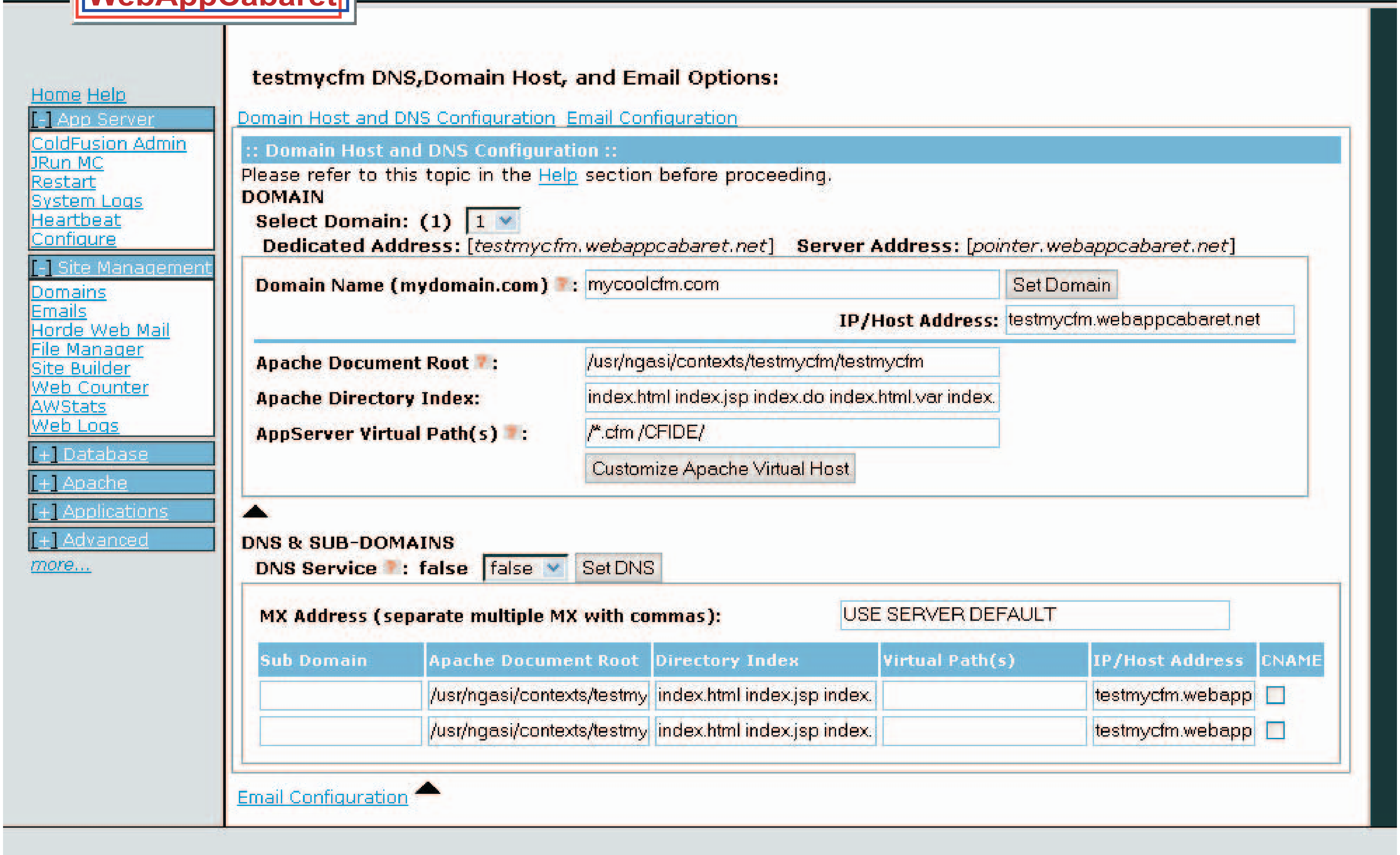
jared@web-relevant.com

Listing 1 Publisher.cfc

```
<cfcomponent extends="clarion.object">
 <!---
  publish-and-subscribe CFC API
  Written Thursday, March 17, 2005
  By Jared C. Rypka-Hauer
     Continuum Media Group LLC
     Burnsville, MN

  This system can act as a composition publish/subscribe system for
    any shared-scope instances.

  The Publisher API has 3 public methods:
   subscribe(subscriber) requres a CFC subscriber instance written to
     the subscriber API
   unSubscribe(UUID) requires a UUID created and maintained by a
     subscriber
   udpate(subscribersStruct) pushes changes to the subscribers

  The Subscriber API must be written to include the following:
   MUST CREATE AND MAINTAIN a UUID for synchronicity with the
     publisher
   getUUID() returns instance UUID
   setUUID() creates a UUID to be maintained in the subscriber's
    instance data
   update() receives the Message structure from the publisher as an
    argument
     - must be coded to handle the messages from the Publisher...
       takes an arg messageStruct
   subscribe() passes itself to the Publisher via publisher.
     subscribe(this)
     - may include code such as: <cfset application.Publisher.
       subscribe(this)>
   unSubscribe() passes it's internal UUID to the Publisher for
```

WebAppCabaret℠

ColdFusion MX 6.1 Enterprise Hosting made easy.



testmycfm DNS,Domain Host, and Email Options:

:: Domain Host and DNS Configuration ::

Please refer to this topic in the Help section before proceeding.

DOMAIN

Select Domain: (1) 1

Dedicated Address: [*testmycfm.webappcabaret.net*] Server Address: [*pointer.webappcabaret.net*]

Domain Name (mydomain.com): mycoolcfm.com

IP/Host Address: testmycfm.webappcabaret.net

Apache Document Root: /usr/ngasi/contexts/testmycfm/testmycfm

Apache Directory Index: index.html index.jsp index.do index.html.var index.

AppServer Virtual Path(s): /*.cfm /CFIDE/

DNS & SUB-DOMAINS

DNS Service: false false

MX Address (separate multiple MX with commas): USE SERVER DEFAULT

Sub Domain	Apache Document Root	Directory Index	Virtual Path(s)	IP/Host Address	CNAME
	/usr/ngasi/contexts/testmy	index.html index.jsp index.		testmycfm.webapp	□
	/usr/ngasi/contexts/testmy	index.html index.jsp index.		testmycfm.webapp	□

Imagine a hosting company dedicated to meet the requirements for complex web sites and applications such as those developed with ColdFusion MX.
At WebAppCabaret our standards based process and tools make deploying ColdFusion MX applications as easy as a point-and-click.
We call it **Point-and-Deploy Hosting**.
Our advanced NGASI Web Hosting management Control was designed for the hosting and management of ColdFusion web sites and applications thus cutting down on maintenance time.

Backed by an experienced staff as well as a **Tier 1 Data center** and network. Our network is certified with frequent security audits by reputable security firms.

All ColdFusion hosting plans have separate and individual installation AND instances of ColdFusion MX 6.1 Enterprise with **full access to ColdFusion Administrator** and JRun Management Console; so there is virtually no restriction or customization required for your application.

Complete power and control at the tip of your fingers. We take care of the system and hosting infrastructure so you can concentrate on development and deployment of your application. That is **real ROI**.

Log on now at http://www.webappcabaret.com/cdj.jsp or call today at 1.866.256.7973

WebAppCabaret℠

Point-and-Deploy J2EE Hosting

http://www.webappcabaret.com/cdj.jsp 1.866.256.7973

Prices,plans, and terms subject to change without notice. Please log on to our website for the latest price and terms.

```
    removal
    - may include code such as: <cfset application.Publisher.
unSubscribe(getUUID())>
  --->

 <cfset setSubscribers(structNew())>

 <!------------------------------------------->

 <cffunction name=”setSubscriber” access=”private” returntype=”void”
output=”false”>
  <cfargument name=”subscriber” type=”clarion.providers.subscriber”
required=”true” />
  <cftry>
   <cfif structKeyExists(arguments[“subscriber”],”getUUID”)>
    <cfset variables.subscribers[arguments[“subscriber”].getUUID()] =
arguments[“subscriber”]>
   <cfelse>
    <cfthrow errorcode=”0.1”
     message=”Subscriber Instances MUST supply a UUID to the
      publisher”
      detail=”Subscriber CFC instances must supply a UUID to the
publisher, and a method labeled getUUID() which returns a UUID to the
calling entity.”>
   </cfif>
   <cfcatch type=”Any”>
    <cfrethrow />
   </cfcatch>
  </cftry>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”getSubscriber” access=”public” returntype=”clarion.
providers.subscriber” output=”false”>
  <cfargument name=”subscriberID” type=”UUID” required=”true”>
  <cftry>
   <cfif structKeyExists(variables[“subscribers”],arguments[“UUID”])>
    <cfreturn variables.subscribers[arguments[“subscriberID”]] />
   <cfelse>
    <cfthrow errorcode=”0.2”
     message=”Invalid Subscriber ID”
      detail=”The subscriber UUID passed into method getSubscriber()
was not found, is invalid, or has fallen thru a wormhole and is now
enjoying a Pangalactic Gargle Blaster on Omicron Beta V.”>
   </cfif>
   <cfcatch type=”Any”>
    <cfrethrow />
   </cfcatch>
  </cftry>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”killSubscriber” access=”public” returntype=”void”
output=”false”>
  <cfargument name=”subscriberID” type=”UUID” required=”true”>
  <cftry>
   <cfif
structKeyExists(variables[“subscribers”],arguments[“subscriberID”])>
    <cfset structDelete(variables[“subscribers”],arguments[“subscriberID”]) />
   <cfelse>
    <cfthrow errorcode=”0.3”
     message=”Invalid Subscriber ID”
      detail=”The subscriber UUID passed into method killSubscriber()
was not found, is invalid, or has fallen thru a wormhole and is now
enjoying a Pangalactic Gargle Blaster on Omicron Beta V.”>
   </cfif>
   <cfcatch type=”Any”>
    <cfrethrow />
   </cfcatch>
  </cftry>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”setSubscribers” access=”public” returntype=”void”
output=”false”>
  <cfargument name=”subscribers” type=”struct” required=”true”>
  <cfset variables[“subscribers”] = arguments[“subscribers”]/>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”getSubscribers” access=”public” returntype=”struct”
output=”false”>
  <cfreturn variables[“subscribers”] />
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”subscribe” access=”public” returntype=”void”
output=”false”>
  <cfargument name=”subscriber” type=”clarion.providers.subscriber”
required=”true”>
  <cfset setSubscriber(arguments[“subscriber”])>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”unSubscribe” access=”public” returntype=”void”
output=”false”>
  <cfargument name=”subUUID” type=”UUID” required=”true”>
  <cfset killSubscriber(arguments[“subUUID”])>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”update” access=”public” returntype=”void”
output=”false”>
  <cfargument name=”message” type=”struct” required=”true”>
  <cfset var updateSubs = getSubscribers()>
  <cfloop collection=”#updateSubs#” item=”i”>
   <cfset updateSubs[i].update(message)>
  </cfloop>
 </cffunction>
</cfcomponent>
```

Listing 2 testPubObject.cfc

```
<cfcomponent extends=”clarion.object”>
```

[**Engage and Explore**]
the technologies, solutions
and applications that are
driving today's **Web services**
initiatives and strategies...

www.sys-con.com/edge2005

web services EDGE
conference&expo
FALL SERIES

CALL FOR PAPERS NOW OPEN!*

Web Services Edge
Fall Conference Series

3 Dynamic Conference Programs
Targeting Major Industry Markets

20+ seminars within 5 tracks will address the hottest topics & issues:

- ► Web Services: The Benefits and Challenges
- ► Web Services Security
- ► SOA (Service-Oriented Architecture) and ESB (Enterprise Service Bus) Strategies
- ► Interoperability, Incremental Integration, & Open Source
- ► The Management Process in Developing a Web Services Strategy

Program Features:

- ► Keynotes
- ► Tutorials
- ► Panel Discussions

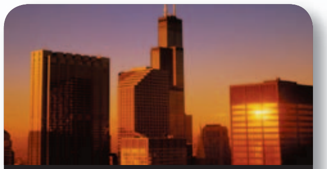
Why Attend:

- ► Improve the return on your technology investment
- ► Develop & sharpen your strategy and identify key action steps
- ► Find new ways to reach and impress customers with Web services
- ► Maximize the power of your enterprise
- ► Protect your business from security threats
- ► Assess Web services as a viable option

Attention Exhibitors:
► An Exhibit-Forum will display leading Web services products, services, and solutions

1
The Westin Washington, D.C.
Washington, D.C.
September 7-8, 2005

2
The Westin Santa Clara Convention Center
Santa Clara, CA
October 24-25, 2005



3
Hyatt O'Hare Airport
Chicago, IL
Oct 31-Nov 1, 2005

Register Today! www.SYS-CON.com/Edge2005

Sponsored by

WebServices Journal · XML Journal · .NET Developer's Journal · eclipse) developer's journal · WebSphere Journal · information Storage+Security journal · wldj · JDJ · LinuxWorld Magazine · MX developer's journal · asp.netPRO · SD Times · CoDe · Software Test & Performance

*Call for Papers email: grisha@sys-con.com

For Exhibit and Sponsorship Information ► Call 201 802-3066

Produced by SYS-CON EVENTS

```
  <cffunction name=”init”>
    <cftry>
      <cfset variables[“publisher”] = createObject(“component”,”clarion.
providers.publisher”).init()>
      <cfcatch type=”any”>
        <cfrethrow />
      </cfcatch>
    </cftry>
    <cfreturn this />
  </cffunction>

  <!------------------------------------------->

  <cffunction name=”getPublisher” access=”public” returntype=”clarion.
providers.publisher” output=”false”>
    <cfreturn variables[“publisher”] />
  </cffunction>

</cfcomponent>
```

Listing 3 Subscriber objects

```
<cfcomponent extends=”clarion.object”>
  <!---
    publish-and-subscribe CFC API
    Written Thursday, March 17, 2005
    By Jared C. Rypka-Hauer
         Continuum Media Group LLC
         Burnsville, MN

    This system can act as a composition publish/subscribe system
      for any shared-scope instances.

    The Publisher API has 3 public methods:
      subscribe(subscriber) requres a CFC subscriber instance written
       to the subscriber API
      unSubscribe(UUID) requires a UUID created and maintained by
        a subscriber
      udpate(subscribersStruct) pushes changes to the subscribers

    The Subscriber API must be written to include the following:
      MUST CREATE AND MAINTAIN a UUID for synchronicity with the
        publisher
      getUUID() returns instance UUID
      setUUID() creates a UUID to be maintained in the subscriber’s
instance data
      update() receives the Message structure from the publisher as an
argument
        - must be coded to handle the messages from the Publisher...
takes an arg messageStruct
      subscribe() passes itself to the Publisher via publisher.
subscribe(this)
        - may include code such as: <cfset application.Publisher.
subscribe(this)>
      unSubscribe() passes it’s internal UUID to the Publisher for
removal
        - may include code such as: <cfset application.Publisher.
unSubscribe(getUUID())>
    --->
```

```
  <cfset setUUID(fetchUUID())>
  <cfset setSubscribed(false)>
  <cfset setMessagesArray(arrayNew(1))>

  <!------------------------------------------->

  <cffunction name=”setMessage” access=”private” returntype=”void”
output=”false”>
    <cfargument name=”message” type=”struct” required=”true” />
    <cfset arrayAppend(variables[“messages”],arguments[“message”])>
  </cffunction>

  <!------------------------------------------->

  <cffunction name=”getTopMessage” access=”public” returntype=”any”
output=”false”>
    <cfreturn variables[“messages”][1] />
  </cffunction>

  <!------------------------------------------->

  <cffunction name=”setSubscribed” access=”private” returntype=”void”
output=”false”>
    <cfargument name=”subscribeStatus” type=”boolean” required=”true”
/>
    <cfset variables[“subscribed”] = arguments[“subscribeStatus”]>
  </cffunction>

  <!------------------------------------------->

  <cffunction name=”getSubscribed” access=”public” returntype=”any”
output=”false”>
    <cfreturn variables[“subscribed”] />
  </cffunction>

  <!------------------------------------------->

  <cffunction name=”setMessagesArray” access=”private”
returntype=”void” output=”false”>
    <cfargument name=”messages” required=”true” type=”array”>
    <cfset variables[“messages”] = arguments[“messages”]>
  </cffunction>

  <!------------------------------------------->

  <cffunction name=”getMessagesArray” access=”public”
returntype=”array” output=”false”>
    <cfreturn variables[“messages”] />
  </cffunction>

  <!------------------------------------------->

  <cffunction name=”subscribe” access=”public” returntype=”void”
output=”false”>
    <cfargument name=”publisher” type=”clarion.providers.publisher”
required=”true”>
    <cfif not getSubscribed()>
      <cfset setSubscribed(true)>
      <cfset arguments[“publisher”].subscribe(this)>
    </cfif>
```

```
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”unSubscribe” access=”public” returntype=”void”
output=”false”>
  <cfargument name=”publisher” type=”clarion.providers.publisher”
required=”true”>
  <cfif getSubscribed()>
   <cfset setSubscribed(false)>
   <cfset arguments[“publisher”].unSubscribe(getUUID())>
  </cfif>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”update” access=”public” returntype=”void”
output=”false”>
  <cfargument name=”message” type=”struct” required=”true”>
  <cfset setMessage(arguments[“message”])>
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”flush” access=”public” returntype=”void”
output=”false”>
  <cfloop condition=”NOT arrayIsEmpty(variables[‘messages’])”>
   <cfset arrayDeleteAt(variables[“messages”],1)>
  </cfloop>
 </cffunction>

</cfcomponent>
```

Listing 4

```
<cfcomponent extends=”clarion.object”>
 <cffunction name=”init”>
  <cftry>
   <cfset variables[“subscriber”] = createObject(“component”,”clarion.
providers.subscriber”).init()>
   <cfcatch type=”Any”>
    <cfrethrow />
   </cfcatch>
  </cftry>
  <cfreturn this />
 </cffunction>

 <!------------------------------------------->

 <cffunction name=”getSubscriber” access=”public” returntype=”clarion.
providers.subscriber” output=”false”>
  <cfreturn variables[“subscriber”] />
 </cffunction>

</cfcomponent>
```

Download the Code...
Go to www.coldfusionjournal.com

Not Your Father’s Charting

A guide to the revamped ColdFusion MX 7 charting feature

– based on an article by Erik Tierney

By Tim Buntel

Since charting was first added to ColdFusion back in version 5, it has consistently provided a lot of customer feedback – both positive and negative. We listened to all of the comments and studied the long enhancement list. Developers are sure to be pleased with the number of charting wishes that have been fulfilled in ColdFusion MX 7. Charting isn’t a new feature, but we’ve definitely given it an “extreme makeover” for this release!

Charts with Style

Did you know that there are more than 45 attributes used to define charts with CFCHART and its subtags? This makes for messy code and a big challenge to get all of your site’s charts to share a consistent look and feel. In ColdFusion MX 7, we’ve added the idea of chart “styles.” These are external definitions of all information to specify a chart’s look and behavior. All you need to do in your CFM pages now is specify the data and which style to use. Of course, you can still use the old, familiar attributes as well. It’s your choice.

ColdFusion MX 7 ships with a number of chart styles predefined and named for the color scheme that they use: beige, blue, red, silver, yellow, and a default style. They are stored in an XML file, and you can see them in your cfusion\charting\styles directory. Any of these styles can be used by name in any ColdFusion template. For example:

```
<cfchart style="silver">
…
</cfchart>
```

These styles look pretty nice, but there are many ways that you can customize them beyond the colors they use. The simplest way is to modify one of these built-in styles. In addition, you can use the WebCharts 3D Desinger tool, included in ColdFusion MX 7, to create your own from scratch.

If you want to modify one of the basic styles, open it in any text editor, make your changes, and save it with a new name. If you keep it in the cfusion\charting\styles directory, ColdFusion pages can call it with the new name (myChart.xml = <cfchart style=”myChart”>) or you can save it elsewhere and call it in the same way you use cfinclude (e.g. a web directory or mapping). So, c:\inetpub\wwwroot\charts\myChart.xml = <cfchart style=”\charts\myChart.xml”>).

The WebCharts3D Designer, created by GreenPoint, Inc., is a tool with advanced style creation and editing that you can use to create far more customized charts. You can use the tool by following these steps:

1. Start WebCharts3D by running the webcharts.bat file in the CFusionMX7\charting directory.
2. Select the type of chart you’d like to use.
3. On the design tab, make the changes you want to the chart’s appearance. (There is a help tab with a chapter called “Designer” that also contains information on using the tool.)
4. Click on the XML Style tab.
5. Click the save button.
6. Name the style with a .xml extension (myChartStyle.xml).

You can then use the style above by either saving it to cfusion\charting\styles or another directory.

With custom chart styles, we should be able to finally say goodbye to the days of users needing to ask Macromedia, “Can you add an attribute to let me configure such and such?” Now you can control every small detail of your chart’s look from colors and fonts to axis scales and number formatting *and* save it in an external file for easy reuse.

From Monochrome to Multicolor

Another item that was near the top of our CFCHART wish list is the ability to create bar charts in which each bar is a different color. With ColdFusion MX 7, it’s not only possible, it’s downright easy, thanks to the new colorList attribute of the chartseries tag.

Let’s say you’re creating a bar chart that shows sales for the previous four quarters. To remind your audience which quarter is which, you decide to color code the data by season with icy blue for winter, pink for spring, green for summer, and orange for fall. Your code would look something like this:

```
<cfchart>
 <cfchartseries type="bar" colorlist="3366FF,FF3399,green,FF6600">
```

```
   <cfchartdata item="winter" value="30">
   <cfchartdata item="spring" value="36">
   <cfchartdata item="summer" value="40">
   <cfchartdata item="autumn" value="33">
 </cfchartseries>

</cfchart>
```

ColdFusion will use each color in the color list for each successive individual bar. If you have more data items than colors in the list, it will start back at the beginning. Of course, you could also create a custom chart style as explained above with the colors you'd like, but this is a fast and easy way to override the colors in an individual series.

En-Title-ment

Support for specifying a chart title came in ColdFusion 5…and went in ColdFusion MX. But now it's back. For example,

```
    <cfchart title="Our Best Year
Ever!"><cfchart>
```

proclaims the success of the past four quarters (which are neatly color coded by season!) And, by using a custom style, you can also have full control over the title placement, color, font, size, and more.

Charts in Reports…

As you know, ColdFusion MX 7 introduces the ability to deliver high-quality, structured reports. These reports can be fully integrated into web applications and provide users with well-formatted data that is easy to understand, print, and e-mail. It's structured business reporting power, otherwise only possible by using expensive, third-party reporting products.

Charts and graphs are very common in reports, so we wanted to bring to you the same superior charts that you can create with CFML in ColdFusion reporting. With the ColdFusion Report Builder, this is a simple task. ColdFusion Report Builder is a free, easy-to-use tool with a familiar, banded report writer interface that enables you to design structured, repeating-region reports for ColdFusion MX 7 applications.

If you want to insert a chart into your report, simply click Insert->Chart. This will bring up the Chart wizard, a full-featured visual chart builder which allows you to edit all of the attributes of the CFCHART tag. You can also use the built-in styles or any of the custom styles you have created for reuse. This way the charts in your report will have a consistent look and feel with the other charts on your site.

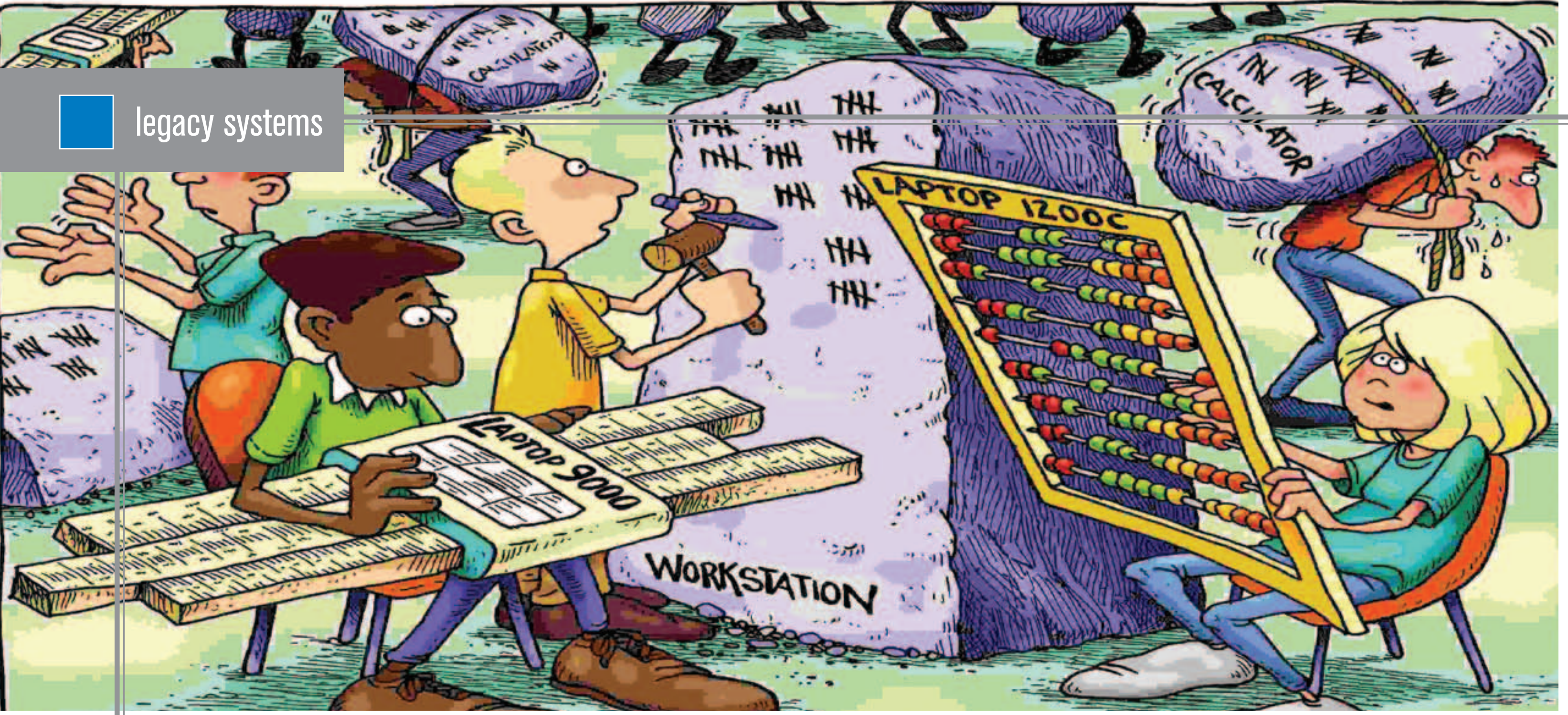
Summary

In figuring out what to improve in its charting makeover, Macromedia listened to its customers. With custom chart styles, there are virtually no limits to how finely you can control the look and feel of charts in your web applications. And it's simple to re-use them in all of your web pages or in new ColdFusion reports. Our CFCHART wish list may not be quite empty, but it's a lot shorter than it was before ColdFusion MX 7

About the Author

Tim Buntel is senior product manager for ColdFusion at Macromedia.

tbuntel@macromedia.com



CFMX and the Mainframe

Making CFMX talk with legacy systems needn't be a daunting task

By Tim Burton

Many of the mainframe computers purchased by companies sometime in the past 3 decades are still in use. This is because they were a major financial investment to buy, they house years' worth of crucial data, resources have been spent to develop applications for these systems, and because quite honestly they still do their job well.

"Web-enabling" these legacy systems can be a daunting task…but it needn't be. This article looks at three methods for making CFMX talk with the mainframe; our shop uses all three methods extensively in order to deliver a suite of heavily-used e-commerce applications over the Internet.

The Mandate

The state agency where I work does hundreds of thousands of transactions involving about a quarter billion dollars yearly, nearly all of it supported on the mainframe (Z800 with OS/390) and written in COBOL with a bit of SQL thrown in. Nearly all of the transactions are processed by analysts sitting at "dumb terminals." Our mandate was to web-enable all applicable processes for the agency's customers…as quickly as possible. Our efforts have been enormously successful due in large measure to good development-business collaboration and concerted in-house training. The series of applications that have been deployed started with the lowest hanging fruit on the tree: publicly accessible inquiries, which means using CFMX to grab data from the mainframe's DB2.

The CFMX-ODBC Scenario

We won't explain this scenario in detail, because creating and using database connections is a ColdFusion developer's bread and butter activity. We initially used IBM's DB2 Connect Enterprise Edition as the connection mechanism, which involved setting up and maintaining a special server to act as a conduit/controller for all DB2 connections. It performed poorly and so we migrated to local ODBC drivers on all CFMX boxes.

Publicly-accessible inquiries and reports served to reduce business analyst's telephone answering to a minimum; next to be tackled were more complex inquiries dependent upon logic embedded in COBOL code and simple transactions dependent upon CICS processes – the next scenario.

CFMX-Metaserver-Transidiom-CICS Scenario

Although there are plenty of products now available to web-enable mainframe-resident data, most have limited flexibility and do not lend themselves to a logical sequence of migration from the "monolithic" application architecture of yesteryear to the newer ideal of a loosely coupled, multitier architecture. With these goals in mind, we purchased and deployed Metaserver, a

very flexible middleware product. Based on the Linda Engine developed at Yale, it serves both as the primary location for finance-related business logic and as the connector to the mainframe via yet another product, Transidiom. See Figure 1 for how these pieces fit together.

The Transidiom IDE contains wizards to script the interaction between a terminal user and a series of mainframe screens; it is a kind of "screen mapping" product, somewhat more sophisticated than traditional "screen scraping" products. For an external customer, hitting a form's "Submit" for a simple transaction means these events take place:

(a) The CFML action template instantiates a Java "MetaClient" object (supplied by Metaserver and deployed as .jar files on the CFMX server) and passes over any arguments to Metaserver;

(b) Metaserver creates a mainframe session and passes the data over to

(c) Transidiom, which runs the automated terminal script on the mainframe. Any responses, including error messages, are passed back through the same chain to CFMX and the user. All data passing between CFMX, Metaserver, and Transidiom are sent as XML packets, which are easily parsed in CFML; I wrote a custom CFC that converts any record-set type data from the mainframe into a cfquery to simplify presentation coding.

This solution allowed us to use existing business code as-is and to web-enable transactions quickly; however, we discovered that customers who were submitting hundreds of records caused a performance bottleneck if each record required a separate series of Transidiom scripts. This was not unexpected; besides, Transidiom was envisioned as a temporary solution. The long-term vision is to replace the old Cobol code with Java and possibly use a business-rules engine. Meanwhile, to relieve the bottleneck, we began to employ yet another method to access mainframe data and code: DB2 stored procedures.

SQL Server and Oracle come packaged with procedural languages (Transact-SQL and pl/SQL, respectively), which makes writing and deploying them relatively simple. DB2 stored procedures aren't packaged this way, which makes them a bit trickier to write and deploy; however, they can be written in C++, Java, or Cobol. Like its competitors, DB2 stored procedures can be called from CFML easily with the <cfstoredproc …> and associated <cfprocparam …> tags. The idea being introduced here is that you can use DB2 as an agent for passing data between CFMX and non-relational data and CICS transactions. Let's look at two scenarios our shop uses.

The CFMX-DB2 Stored Procedure Scenario #1

To use these two scenarios, the mainframe's system programmer will need to make a Work Load Manager (WLM) available to you. Once you have access to a WLM, using this scenario takes these additional steps:

(1) Define the Stored Procedure to DB2 using the "create stored procedure" DDL statement. Your database administrator (DBA) can do this by executing the DDL in SPUFI. See the top of Listing 1 for an example; note that the language declared is COBOL; it can also be Java or C++, but in our shop COBOL is the available skill set. Stored Procedures written in COBOL and SQLJ use the static SQL methods and produce a Database Resource Module (DBRM). The DBRM is bound to DB2 by the DBA and DB2 knows all the access paths stored procedures need. Stored procedures written in COBOL or SQLJ provide a lot better performance than stored procedures written using Dynamic SQL. If your application must execute a stored procedure frequently then you can cache the Program Load Module in a WLM by specifying the STAY RESIDENT parameter as YES when the stored procedure is defined in DB2 .

(2) If you need to access VSAM or QSAM files from a stored procedures then you need to tell the mainframe system programmer to allocate/define the files to a WLM. You can read and update VSAM data using this method. However, if the VSAM file allocated in CICS is in 'update mode' you will not be able to update from within the WLM, in which case you need to use the stored procedure scenario #2 (below); the WLM will make an EXCI call to CICS program and the CICS program will do the VSAM update.

(3) Write the stored procedure itself, making sure that you use the same parameter types and other variable names you declared in the previous step. If you are writing stored procedures in COBOL then using structured programming methods will facilitate maintenance and reuse.

Once you write the stored procedure, compile and bind the stored procedure. The bottom of Listing 1 shows the linkage section of a COBOL stored procedure, which is otherwise ordinary COBOL code.

The CFMX-DB2 Stored Procedure scenario #2

The previous scenario will handle most application requirements. If transactional integrity is required (Atomicity, Consistency, Isolation, Durability) or if you want to leverage existing CICS transactions with minimal changes, then CICS programs can be called from within the stored procedure using the External CICS Interface (EXCI); all input and output parameters are passed through the COMMAREA.

Again, setting up the WLM is similar to scenario #1. For making EXCI calls, the mainframe CICS system programmer has to define EXCI transaction for your CICS region. The top of Listing 2 is DB2 DDL creating a new stored procedure; the bottom of Listing 2 shows two fragments of a COBOL stored procedure declaring and making an EXCI call.

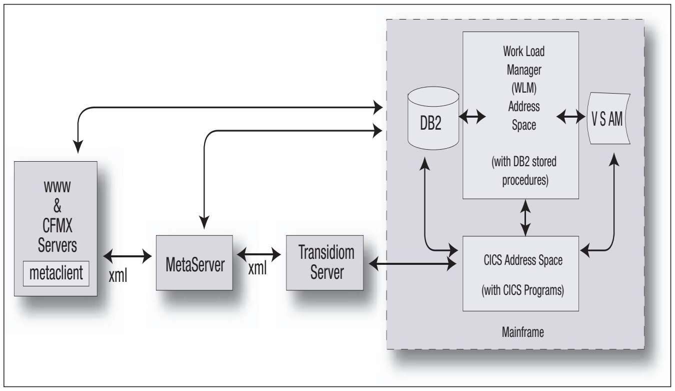


Figure 1: Component Orientation

Runtime performance for both stored procedure scenarios is very fast.

Conclusions

Using the last two scenarios, you will have access to almost any mainframe datastore, which means that within the COBOL modules you can set up data structures, feed them data from those heterogeneous sources, manipulate the data, run cursors, perform complex business logic, and then return the results (including record sets) to CFMX.

The average Cold Fusion developer isn't likely to know much about the mainframe; therefore, in-house systems programmers and COBOL developers must cooperate to deploy these solutions.

Many thanks to Madan Mankala and Jim Leamon for their mainframe expertise.

About the Author

Tim Burton is the EGovernment Applications Architect for a large state agency in Oregon and has been writing CFML since 1998. This is his third career; he previously practiced medicine and made art (metal sculpture). *Tim.BURTON@odot.state.or.us.*

Listing 1

Defining a Stored Procedure in DB2 (the DLL):

```
CREATE PROCEDURE CMCTRAIN.CNASTADR (MSTRID CHAR(9) IN,
                                    RCODE CHAR(4) OUT)
    LANGUAGE COBOL
    DYNAMIC RESULT SET 1
    PARAMETER STYLE GENERAL
    EXTERNAL NAME CNASTADR
    WLM ENVIRONMENT DSNMCTT
    SECURITY DEFINER
    COLLID CMC_TRAIN
    PROGRAM TYPE MAIN
    COMMIT ON RETURN YES
```

COBOL program code's linkage section:
(What precedes and follows is normal COBOL code)

```
LINKAGE SECTION.
 01  INPUT-MSTR-ID                  PIC 9(09).
 01  OUTPUT-RETURN-MESSAGE          PIC 9(04).
*
 PROCEDURE DIVISION USING INPUT-MSTR-ID
                          OUTPUT-RETURN-MESSAGE.
```

Listing 2

Defining a Stored Procedure in DB2 (the DLL):

```
  CREATE PROCEDURE FMTMDEVL.FMSP060 (
                                    IN  USER_ID       CHAR   (08)
,
                                    IN  ENVIRONMENT   CHAR   (04)
,
                                    IN  BILL_CYCL_DATE CHAR  (10)
,
                                    IN  TRANS_CODE    CHAR   (03)
,
                                    IN  OPERATOR_ID   CHAR   (04)
,
                                    OUT RET_BATCH_DATE CHAR  (06)
,
                                    OUT RET_BATCH_NO  CHAR   (03)
,
                                    OUT RETURN_CODE   CHAR   (02)
,
                                    OUT ERROR_DESC    VARCHAR(250)
                                    )
          LANGUAGE COBOL
          DETERMINISTIC
          EXTERNAL NAME FMSP060
          DYNAMIC RESULT SETS 0
          WLM ENVIRONMENT DSNFMT
          COLLID FMT_DEVL_DTEAMS
          PARAMETER STYLE GENERAL
          PROGRAM TYPE MAIN
          STAY RESIDENT NO
          SECURITY USER
          RUN OPTIONS 'MSGFILE(SYSOUT,FBA,121,0,ENQ)'
          COMMIT ON RETURN NO;
```

Lines from the stored procedure dealing with EXCI:

the first line; declares ext option to use EXCI:

```
    00001  CBL XOPTS(EXCI)
```

(What follows is normal COBOL code)…

Lines 269-283; the actual EXCI call:

```
    00269  200-LINK-CICS-PROGRAM.
    00273      MOVE '200-LINK-CICS-PROGRAM'     TO WS-PARAGRAPH-NAME
    00274

    00275      EXEC CICS LINK

    00276          PROGRAM    (WS-TARGET-PROGRAM)
    00277          TRANSID    (WS-TARGET-TRANSID)
    00278          APPLID     (WS-TARGET-SYSTEM)
    00279          COMMAREA   (COMMAREA)
    00280          LENGTH     (LENGTH OF COMMAREA)
    00281          DATALENGTH (LENGTH OF COMMAREA)
    00282          RETCODE    (EXCI-EXEC-RETURN-CODE)

    00283      END-EXEC
```

Download the Code...
Go to www.coldfusionjournal.com

"Ladies and gentlemen, the sky is not falling"

–continued from page 5

The presentation also makes it pretty clear that Adobe shares Macromedia's interest in offering the best platform for delivering rich content to devices and in creating rich applications by combining media, applications, documents, and communication. In other words, Adobe also believes, in the words of the Macromedia slogan, that "Experience Matters." It appears that at the center of this goal are the Flash/FlashPaper and PDF formats as a content delivery format.

In this presentation, Adobe also identifies the benefits it expects to offer customers in the realms of creative professionals, digital imaging and video, and the enterprise (in addition to emerging businesses and technologies). Specific products are highlighted in each of these areas. In addition to Flex, Breeze, and Adobe LiveCycle, ColdFusion appears as part of the product solution for the Enterprise, which is a strong indicator that there is no intention to "do away" with it anytime soon.

So what will the future hold? Nobody can tell for sure, but I will leave you with a few thoughts and observations. The investor presentation gave me the impression that in addition to creating great experiences for users, product integration and workflow are two areas that Adobe feels are very important. I definitely look forward to seeing how this strategy is put into action – especially in the area of ColdFusion development. On a personal note, there are two enhancements to Macromedia products that I for one would love to see as a result of the acquisition. Though I do love Flash, I've also always been a big fan of SVG. I think it would be terrific to see Flash become XML driven (not just on the Flex Server) and have added support for SVG XML. Obviously, it's a fair assumption that PDF support in ColdFusion will only get stronger – I would love to see PDF Forms support, too. I could fill an entire issue of CFDJ with speculation and the details of my personal ColdFusion wish list but the bottom line is that only time will tell.

The last thing I want to leave you with is a statement that I recently made in an online user group presentation, which really sums up my personal opinion about the merger: "The best company at managing digital documents and the best company at managing web content are now one in the same – and this is a very good thing!"

ColdFusion Prevents Nudity

How a high school student became a business owner thanks to CF

By Jon Rowny

Have you ever come up with something clever and thought "Wow that should be on a T-shirt or something"? In the summer before my senior year of high school, my friend Graham Stevenson and I had that thought many times. So many in fact that we decided we should start a T-shirt company to address the growing demand for trendy yet original clothing.

When we first started talking about the idea most people asked us "How are you going to start a business with no money, no investors, and no time?" The answer usually made my peers giggle, "It's easy, I'll use ColdFusion." Of course they didn't know I was talking about the handy server-side scripting tool that would allow me to do everything from marketing my website to order processing and inventory tracking.

Background

I had been using ColdFusion only a bit since I was introduced to it at a summer internship a few summers prior. With Graham's design skills, and my ColdFusion knowledge, we were able to come up with a business plan. That plan consisted of buying a cheap silk screen press, with pizza delivery job savings, and programming a simple order management system based on PayPal's IPN (Instant Payment Notification). Graham designed an array of silly shirts and a neat-looking website.

In the summer of 2003 we launched CureForNudity.com. After a month in business, we had a grand total of one order, about $17 in the bank account. That's when I realized we needed a real shopping cart, more product information, order tracking, shipment tracking, a link exchange program, an affiliate program, a picture submission system, a coupon system, a newsletter, an anti-fraud system, and a cleaner way to check out. I spent the next two years working diligently on my four-step plan to success.

Step One: Getting a Real Shopping Cart

I soon determined that most people would not buy a T-shirt

sion Prevents udity

TOILETS SCARE THE SHIT OUT OF ME
AFGHANISTAN
IRAQ
CANADA
I'm judging you.
got served?
Hairy Potter
HOMELESS
Catnip
"Guaranteed to get you Pussy!"
CCCP
THIS SHIRT BROUGHT TO YOU BY CAPITALISM

from our site because, at the time, it required you to have a PayPal account. In addition pop-up blockers tended to stop PayPal's pop-up shopping cart. We looked at many ColdFusion and non-ColdFusion based shopping carts before realizing that our whopping budget of $0 would not allow us to purchase any pre-fab shopping cart system. Luckily for me, I was able to follow some session management and array tutorials to construct my very own shopping cart. The only thing left to do was ditch PayPal.

Ultimately we decided a merchant account and credit card processor would be the best solution. But we were both under-aged so we decided to use a service called 2CheckOut until my eighteenth birthday. Once we moved to our new shopping cart system, the orders started to pick up a little and we began to feel the burn of paying nearly 5% of each transaction to our processor. I began to look for a payment gateway. Verisign and Authorize.net appeared to be the options. Authorize had a nice, well-defined API which I was easily able to learn; thankfully it was also about half the price of Verisign. Authorize.net also provided me a great anti-fraud suite. However, a month into setting up the Authorize.net gateway I ran into a serious problem: every bank I called laughed at the thought of giving an 18-year old with absolutely no credit history a merchant account. Wells-Fargo laughed the hardest I think. Fortunately for us, the Authorize.net reseller we chose did not want to lose us because of something as silly as our age, so she was able to talk some no-name high-risk bank in New Jersey into giving us an account at a slightly higher percentage. 14 forms and 6 faxes later we were in business.

Step Two: Internet Marketing 101

In April of the next year my friend Kelly Brown, who first introduced me to ColdFusion back at my summer internship, hired me to do some Internet marketing for his company called CareerBank.com. I used the skills I learned during the day to transfer over to CureForNudity.com, and the things I learned working on CureForNudity.com transferred right back into CareerBank.com, which made me look good. Because CareerBank.com is a ColdFusion based website, I was also able to learn a bit about how a real e-commerce business functions.

A first step for Internet marketing was to make my products indexable in the search engines. Search engines seem to have trouble indexing query parameters. Using a little CFScript only a few lines long, I was able to make my URLs more acceptable:

Before: http://www.curefornudity.com/product.cfm?id=6

After: http://www.curefornudity.com/product.cfm/gotserved.htm

Each product was assigned a unique name. This was done automatically by removing spaces and special characters from the title of the shirt. Then a CFScript parses the URL to the right of the slash, removes the *.htm*, and queries the product. The *.htm* tricks the search engines into thinking that this is a static page. The trick worked almost instantly as all of our T-shirts were indexed in Google.

The next step for Internet marketing was an affiliate program. At first we tried to get accepted to some programs like Commission Junction but realized they weren't interested in low-traffic T-shirt startups. Again, there were many pre-fab ColdFusion apps to choose from but again our budget made us realize that they were not an option. So I added another table to the database. In order to gain a little credibility with affiliates we added a tiny output near the copyright info which lets you know the current affiliate ID associated with a potential sale, this way the affiliates can be sure that their sales are being counted. We even created a place for the affiliates to login and see tips on selling our products in addition to code they could cut and paste to their websites, blogs, or even instant message profiles.

My favorite marketing tactic, which will be in use soon on our site, is a picture submission system. We will hold contests every month where users can submit photos of themselves in our T-shirts. Having user-submitted photos helps show that our product is real and tends to ease people into buying. The system is simple: CFFILE with some validation and then an administrator can approve the picture for publishing it in association with the design it represents.

A report generator is necessary to measure the effectiveness of marketing efforts. The backend can build reports for date ranges geographically, by affiliate ID, by coupon used, or by what products were ordered.

Finally, we also needed a simple coupon code manager. We can add a coupon code to give a certain percent, a dollar amount, or a free shirt based on minimums purchased. This was particularly important for tracking our success after purchasing an advertorial in Rolling Stone Magazine.

Step Three: Order Management

There are many e-commerce solutions out there to power your website, but none specifically designed to handle T-shirt inventory, order status, and a printing queue to show what needs to be made in the shop. In each incarnation of the backend to the website, the interface got easier to use. If I thought, "Wow, I really wish the shirts highlighted to show if they'd been printed," I could go upstairs and program that feature. I added feature after feature until the backend allowed me to do anything I wanted from editing people's phone numbers to entering in shipping and delivery dates. I combined the shipping system with a service called Endicia which is a desktop application that can print postage from USPS.

Another hurdle was the inventory system. Because we print our own shirts, the inventory system needed to know only how many T-shirts we have of each color, style, and size; this negates the design of the shirts as an inventory factor. The system alerts me when we fall below 6 or 7 shirts of a particular style, size, or color.

After steadily adding features to the order management system for almost a year, things got a little buggy. (Okay…*very* buggy.) Just as MX 6.1 came about, I decided the site needed a healthy face-lift. At the time I was also thinking about selling the site, so I added form validation to things I would've let slip before. Although a few features in the new system have not yet become fully live, the backend to CureForNudity.com is better than ever and fully customized to sell t-shirts. In addition to updating the code to take advantages of MX 6.1's CFCs I also removed all JavaScript from the entire site.

Step Four: Continue to Grow

With a solid website structure that handles orders smoothly and has the ability to steadily add features as required, CureForNudity.com will continue to grow. Adding features to the site is easier than ever thanks to the Object Oriented Programming features brought about in MX 6.1.

"Search engines seem to have trouble indexing query parameters. Using a little CFScript only a few lines long, I was able to make my URLs more acceptable"

In the future we will expand to include bumper stickers, buttons, accessories and more. The system is already setup to accept new product types without issue. Soon product price groups will be added so I can add a bunch of items to a "10% off group" or a "blue light special group." This will allow us to target sales at products that aren't doing so well. With website concerns out of the way I will be able to focus on marketing our product to wholesale audiences. Before long, we might be in your local boutique. If I run into some free time, maybe I'll program my own wholesale order system complete with contact and sales management.

Learning That "Confidence Conquers"

The most important lesson I learned from my work CureForNudity.com is confidence. With the support of the ColdFusion community you can do anything on the web, despite your age, income, or experience. You would never believe how little ColdFusion I knew when I started following my first IPN tutorials. I could barely do update, insert, and select. Now I believe that the sky is the limit. A one-person team can go a long way with ColdFusion because even though I am just one person, with the strength of the community behind me I'm like an army of programmers. If I have a question there is an answer on a forum somewhere. If I am totally lost I can always find answers in Google. And when I need some quick reference, Macromedia livedocs is only a click away. Although I've only had brief forays into the world of ASP and PHP, I'm convinced that ColdFusion is the easiest to learn, yet extremely powerful.

About the Author

Jonathan Rowny is currently studying Information Technology at George Mason University. In addition to cofounding CureForNudity.com Jonathan also works part time as a junior developer and internet marketing specialist for AboutWeb LLC.

jrowny@careerbank.com

CFDJ Advertiser Index

ADVERTISER	URL	PHONE	PAGE
ACTIVEPDF	WWW.ACTIVEPDF.COM	866-468-6733	25
BLOG-N-PLAY	WWW.BLOG-N-PLAY.COM	888-303-5282	39
CFDYNAMICS	WWW.CFDYNAMICS.COM	866-233-9626	4
CFUNITED	WWW.CFUNITED.COM	301-424-3903	COVER III
COLDFUSION DEVELOPER'S JOURNAL	WWW.SYS-CON.COM/COLDFUSION	888-303-5282	49
EV1 SERVERS	WWW.EV1SERVERS.NET	800-504-SURF	3
HAL HELMS, INC	WWW.HALHELMS.COM		31
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM/CFDJ	877-215-4678	21
INTERAKT ONLINE	HTTP://KTML.INTERAKTONLINE.COM	4031 401.68.19	6
INTERMEDIA.NET	WWW.INTERMEDIA.NET	800-379-7729	COVER IV
ISSJ	WWW.ISSJOURNAL.COM	888-303-5282	41
IT SOULUTIONS GUIDE	WWW.SYS-CON.COM	201-802-3021	43
JAVA DEVELOPER'S JOURNAL	JDJ.SYS-CON.COM	888-303-5282	35
LINUXWORLD MAGAZINE	WWW.LINUXWORLD.COM	888-303-5282	41
MACROMEDIA	WWW.MACROMEDIA.COM/go/cfmx7_demo	415.252.2000	COVER II
MACROMEDIA WPS	WWW.MACROMEDIA.COM/go/webupdate	415.252.2000	11
MX DEVELOPER'S JOURNAL	WWW.SYS-CON.COM/MX/SUBSCRIPTION.CFM	888-303-5282	45
PAPERTHIN	WWW.PAPERTHIN.COM	800-940-3087	29
SAVVY SOFTWARE	WWW.BESAVVY.COM	866-870-6358	19
SEAPINE SOFTWARE	WWW.SEAPINE.COM/WEBDEV	888-683-6456	9
SYS-CON E-NEWSLETTERS	WWW.SYS-CON.COM	888-303-5282	35
SYS-CON PUBLICATIONS	WWW.SYS-CON.COM/2001/SUB.CFM	888-303-5282	33
SYS-CON REPRINTS	WWW.SYS-CON.COM	201-802-3026	43
WEB SERVICES EDGE 2005	WWW.SYS-CON.COM/Edge2005	201-802-3066	17
WEBAPP CABARET	WWW.WEBAPPCABARET.COM/cdj.jsp	866-256-7973	15

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of. All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in ColdFusion Developer's Journal. Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc.This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions This index is provided as an additional service to our readers. the publisher does not assume any liability for errors or omissions.

Reading File as from Java

– continued from page 10

If not, read up on encryption here http://livedocs.macromedia.com/coldfusion/6.1/htmldocs/functi75.htm#wp1104201 and using the cfquery tag http://livedocs.macromedia.com/coldfusion/6.1/htmldocs/tags-b19.htm#wp1102316.

A Final Thought about CF, Macromedia, and Adobe

I am writing this article fewer than 24 hours after Adobe and Macromedia announced that they would be merging into a single company yet. The mailing lists and blogs are awash with a mix of despair and excitement, or sometimes both. When I write articles like this, I can't help but think about the power that lies in our hands with the ability to write ColdFusion code. No matter what the future plans are, I can't help but look forward with optimism. Rest assured I hope that this event will officially end the "Microsoft wants to buy Macromedia" doomsday threads.

About the Author

Jeff Houser has been working with computers for over 20 years and in Web development for over 8 years. He owns a consulting company and has authored three separate books on ColdFusion, most recently ColdFusion MX: The Complete Reference *(McGraw-Hill Osborne Media).*

jeff@instantcoldfusion.com

Getting Started Integrating CFML with Java & .NET

Integration opens some cool doors

By Charlie Arehart

It's great to have a whole issue of CFDJ focused on integrating CFML with Java. The process is a lot easier than many might think, as the articles in this issue will show.

I'd like to review the very basics to get you started, showing some of the cool doors of opportunity that this integration opens. Perhaps more important for some, though, I'd like to also point out that direct integration of CFML with .NET is also possible.

Surprised to hear about direct integration of CFML with .NET? You may know that you can call .NET objects through web services in CFMX. But you can also call them directly (as opposed to via a web service, as well integrate CFML and ASP.NET) using BlueDragon's .NET edition. Don't worry. If you're only interested in CF, this article is still for you in its introduction to integrating CFML with Java.

Solutions for CF 4.5/5/MX and BlueDragon

Indeed, even if you're still using CF 4.5 or 5, and never transitioned to CFMX, this article will still apply. Some may be surprised to hear that integration of CFML with Java was introduced in CF 4.5. It's certainly true that the Java-based CFMX and BlueDragon engines make this integration easier; still, it is possible to do Java integration with CF 4.5 and 5.

In fact, some readers of ***CFDJ*** may recall the excellent 8-part series by Guy Rish from called "A Cold Cup o' Joe", where he introduced many of us for the first time to the idea, benefits, and challenges of doing such CFML/Java integration. He showed how you can install the Java Runtime on your machine and enable access to it from within CFML with a single change to the CF Admin (you can also see the CF 4.5/5 docs for more.) It's great to see him back in this issue, with "Cold Cup o' Joe – Second Shot" and I wish him luck with this new series.

In the first series, Guy showed how to call Java objects from within CFML, as well as how to call upon and integrate CFML with JSPs and Servlets. Of course, with CFMX and BlueDragon both being built upon Java, you don't need to make any changes to leverage Java. It's just built-in. Still, most of Guy's series will still apply and would be highly recommended for further reading.

The CFMX docs also now have an entire chapter on CFML/Java integration, available online at http://livedocs.macromedia.com/coldfusion/6.1/htmldocs/java.htm. I certainly don't want to repeat all the information provided in these useful and rich resources.

Why Bother With Java?

We all know that CFML is a wonderfully high-level language. Why would we want to bother with Java? Indeed, the reason we like it and are so productive is that we can do nearly everything we want right from within CFML. Still, some hold this simplicity against CFML, lamenting that if they want to do something that CFML can't do, they feel stuck. You're not.

There's an entire world of new possibilities in the form of Java objects and libraries that you can leverage to extend your CFML. These can either be the built-in libraries that come with the Java framework, or that you or others in your organization build, or that you acquire from third parties.

An important point to clarify is that it is not necessary that you know Java to do the kind of integration I'm discussing.

For example, there is no CFML tag or function that will return the IP address for a given domain name. But there is a way to get this information by calling the InetAddress class in the Java API. The Java API documentation for this can be found at http://java.sun.com/j2se/1.4.2/docs/api/java/net/InetAddress.html. With a little reading, you'd learn that the GetByName method accepts a host name and returns an InetAddress object for that host name, which then has a getHostAddress method that returns the actual IP address.

The following CFML example demonstrates one way to call on this object and its methods, using CFOBJECT:

```
<CFOBJECT NAME="iaddrClass"  CLASS="java.net.InetAddress" TYPE="JAVA"
ACTION="CREATE">
<CFSET IADDR=iaddrClass.getByName("www.newatlanta.com")>
<CFOUTPUT>#iaddr.getHostAddress()#</CFOUTPUT>
```

Recall that since CFMX and BlueDragon are built atop the Java API, you have direct access to the Java API and this java.net.InetAddress class. If you run this code (while connected to the internet, since the getByName method does require net access to search for the information about the host name), you will get the following answer:

209.235.23.101

Recall that in order to run this in CF4/5, you need to first modify the CF Admin to enable Java access. See http://livedocs.macromedia.com/coldfusion/5.0/Developing_ColdFusion_Applications/cfobject8.htm.

Beyond the built-in objects in the Java API libraries, you can also access any Java class that is (or has been made) available on the JVM classpath (see the CFMX or BlueDragon Admin console) or which has been placed in either of the following locations under either the CFMX or BlueDragon directories:

- WEB-INF/lib (as a Java archive or .jar file)
- WEB-INF/classes (as a .class file)

The subject of creating Java class files is beyond the scope of this article. I'm sure other articles in the issue will cover the topic, and if they don't recall that the Macromedia docs have an entire chapter on CFML/Java integration. It includes some sample Java code to create your own example class file, and discusses how to compile and place the class files. Beyond that, the documentation discusses several other useful topics regarding integration of Java with CFML, including:

- How cfobject loads the class and makes available any static objects
- How to use the built-in init() method to call the object's constructor
- How to use createObject instead of CFOBJECT, where appropriate
- Passing arguments to methods
- Handling datatype conversion
- Resolving datatype ambiguity with Javacast
- Handling case-sensitivity issues
- Handling java exceptions
- Calling EJBs

One last point of interest in this topic is that CFMX 7 has introduced a new option for Javacast to help cast a variable as a null.

The documentation also discusses integration of CFML with JSPs and servlet, which are the Java form of web application scripting. These are also often referred to as part of the J2EE (Java 2 Enterprise Edition) framework. CFMX was built atop a J2EE server (JRun) and can be run atop others (WebLogic, WebSphere, and so on), as can BlueDragon.

What about .NET?

The previous section focused on integrating CFML with Java. What if your organization is moving to the Microsoft .NET Framework? .NET is Microsoft's response to J2EE, and it offers many of the same features that one may find in J2EE development, and more.

While CFMX is only available to run atop the J2EE framework, it is possible to run your CFML natively on .NET using *BlueDragon for the Microsoft .NET Framework*, a product from New Atlanta (disclaimer: that's the company I work for) which offers an alternative implementation of CFML – abbreviated from hereon as BlueDragon/.NET. You don't need ColdFusion at all if you install BlueDragon, though you can also install both CF and BlueDragon at the same time for testing purposes.

Why should you care? If your organization is moving to .NET, you may feel you have no choice but to rewrite your investment of CFML as ASP.NET. The fact is, with BlueDragon/.NET, you can simply run your CFML natively on the .NET framework. This is a subject for another entire article, and you can find out more at http://www.newatlanta.com/bluedragon/. (In fact, there will be a couple of presentations on the subject at the upcoming CFUnited conference.)

But for the sake of this article, I want to point out that you can equally leverage the underlying .NET framework classes from CFML, just as I discussed in the previous section regarding Java. You can call either the built-in classes of .NET (there are more than 4,000 of them) or ones you build or acquire, and again some of them provide information not available otherwise in CFML.

For instance, if you want to know the name of the computer on which the code is running, the following will leverage the .NET framework's Environment class in the "System" namespace, to return the MachineName. (See the .NET documentation for details on this class, at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/cpref_start.asp.) This CFML will only run on BlueDragon/.NET:

```
<CFOBJECT ACTION="create" type=".net" NAME="env" CLASS="System.Environ-
ment">
<p>Computer Name = <CFOUTPUT>#env.get_MachineName()#</CFOUTPUT>
```

On my machine, this returns CharlieD600, the name of my machine. Not the most compelling example (more on those in a moment), but it shows how very similar this approach is to the Java examples above. Indeed, besides using a .NET class and its method, there is just one difference.

Note the use of TYPE=".net" rather than TYPE="java" as in the previous examples. In BlueDragon/.NET, you can call any underlying .NET class this way. In fact, you could use TYPE="java" as well, and even run the example provided in the previous section, unchanged. Though it runs atop the .NET framework, BlueDragon/.NET also requires installation of the Visual J# Redistributable runtime environment, which makes most of the same Java API libraries available within .NET, and therefore available to CFML pages running on .NET via BlueDragon.

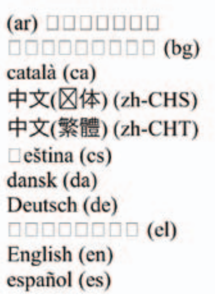
Along the same lines, note that when the CFML code in the .NET example above referred to the MachineName, it used get_MachineName(). The .NET documentation for this class would show that MachineName is a property, rather than a method. But it would also show that when calling the class with Visual J#, one would need to call the property as a method instead. Because of BlueDragon/.NET's underlying reliance on Visual J#, we need to use the same form in our CFML.

Before leaving the subject of .NET integration with CFML, I would add that there is of course a whole lot more that is possible in terms of leveraging underlying .NET libraries, just as there is much more than I showed in the one example of Java integration above. While some of the other articles in this issue (and the Macromedia documentation) will highlight more about Java integration, let me take a moment to show just a little more of what's possible with .NET integration.

Here's one more example, that uses CFML to display the foreign languages available in .NET for supporting globalization:

```
<cfobject name="culturetypes" class="System.Globalization.CultureTypes"
action="CREATE" type=".net" >
<cfobject name="cultures" class="System.Globalization.CultureInfo"
action="CREATE" type=".net" >
<cfset cultarray=cultures.getcultures(culturetypes.NeutralCultures)>
<cfloop from="1" to="#arraylen(cultarray)#" index="i">
    <cfoutput>#cultarray[i].get_NativeName()# (#cultarray[i].get_
Name()#)<br></cfoutput>
</cfloop>
```

A subset of the result shown would include:



```
(ar) □□□□□□□
□□□□□□□□□ (bg)
català (ca)
中文(□体) (zh-CHS)
中文(繁體) (zh-CHT)
□eština (cs)
dansk (da)
Deutsch (de)
□□□□□□□□ (el)
English (en)
español (es)
```

But even beyond this, the .NET framework offers so much more. There is built-in support for calendar, datagrids, adrotators, and so much more. It would be possible to show how to implement even these features within CFML, but again this is beyond the scope of this article. The discussions presented here should be enough to get you started and motivate further interest.

Other Resources for Getting Started

Before concluding, I'd like to point out a few resources to learn more about CFML integration with Java and .NET. Beyond the Macromedia documentation I mentioned above, there are other resources you can turn to in order to learn more about CFML integration and Java. There are two books that cover the topic:

Java for ColdFusion Developers, by Eban Hewitt
Reality J2EE: Integrating ColdFusion and J2EE, by Ben Forta, et al
JRun Web Application Construction Kit, by Drew Falkman

While the latter really isn't about CFML, since CFMX runs atop JRun, it provides the same kind of friendly introduction to leveraging JRun and J2EE in general that was modeled in the infamous *ColdFusion Web Application Construction Kit*.

For more on BlueDragon/.NET, there is an entire manual, *Deploying CFML on ASP.NET Servers*, with many examples. It's available with the download/install of BlueDragon/.NET, or it's available online at http://www.newatlanta.com/products/bluedragon/self_help/docs/6_2/BlueDragon_62_CFML_on_ASP_NET.pdf.

You can also find all the articles in Guy Rish's excellent series from the following link: http://cfdj.sys-con.com/author/113Rish.htm.

About the Author

Charlie Arehart, co-technical editor of ColdFusion Developer's Journal and a Macromedia Certified Advanced ColdFusion developer and trainer, is CTO of New Atlanta Communications, makers of BlueDragon. In this role he continues to support the CFML community, contributing to several CF resources and speaking frequently at user groups throughout the country.

charlie@newatlanta.com

A LIMITED TIME SAVINGS OFFER FROM *SYS-CON MEDIA*

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $340 AND RECEIVE UP TO 3 FREE CDs!*

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

CALL TODAY! 888-303-5282

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

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

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

□ LinuxWorld Magazine

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

□ JDJ

U.S. - Two Years (24) Cover: $144	You Pay: $99.99 /	Save: $45 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $69.99 /	Save: $12
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $89.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ Web Services Journal

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

□ .NET Developer's Journal

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

□ Information Storage + Security Journal

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $39
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

□ Wireless Business & Technology

U.S. - Two Years (12) Cover: $120	You Pay: $49.00 /	Save: $71 + FREE $198 CD
U.S. - One Year (6) Cover: $60	You Pay: $29.99 /	Save: $30
Can/Mex - Two Years (12) $120	You Pay: $69.99 /	Save: $51 + FREE $198 CD
Can/Mex - One Year (6) $60	You Pay: $49.99 /	Save: $10
Int'l - Two Years (12) $120	You Pay: $99.99 /	Save: $20 + FREE $198 CD
Int'l - One Year (6) $72	You Pay: $69.99 /	Save: $2

□ MX Developer's Journal

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

□ ColdFusion Developer's Journal

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

□ WebSphere Journal

U.S. - Two Years (24) Cover: $216	You Pay: $129.00 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189.00 /	Save: $75
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ PowerBuilder Developer's Journal

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

□ WLDJ

U.S. - Four Years (24) Cover: $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
U.S. - Two Year (12) Cover: $120	You Pay: $49.99 /	Save: $70
Can/Mex - Four Years (24) $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
Can/Mex - Two Year (12) $120	You Pay: $69.99 /	Save: $50
Int'l - Four Years (24) $240	You Pay: $120 /	Save: $120 + FREE $198 CD
Int'l - Two Year (12) $120	You Pay: $79.99 /	Save: $40

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

3-Pack

Pick any 3 of our magazines and save up to $210.00

Pay only $99 for a 1 year subscription plus a FREE CD

- 2 Year – $179.00
- Canada/Mexico – $189.00
- International – $199.00

6-Pack

Pick any 6 of our magazines and save up to $340.00

Pay only $199 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $379.00
- Canada/Mexico – $399.00
- International – $449.00

9-Pack

Pick 9 of our magazines and save up to $270.00

Pay only $399 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $699.00
- Canada/Mexico – $749.00
- International – $849.00

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

SYS-CON MEDIA

Conference Review: Powered by Detroit

PbD Was a Great Show

By Michael Smith

I just got back from the "Powered by Detroit" (http://poweredbydetroit.org/) ColdFusion and Flash conference in Detroit Michigan on April 9-10.

I arrived Friday morning in time for the User Group manager sessions and met some old friends and gleaned some tips on running my group (MDCFUG) better. The conference was sponsored by local user groups and there are many strong groups in the area. Over 120 people attended.

The conference hotel was a Hyatt and was right next to the Ford world headquarters building (Detroit is motor town – freeways going everyway – walking is not a big option here!). Next, it was on to the speaker dinner and a chance to chat with other speakers from across the country. I met R Blank from the LA Flash UG who had just flown in to be here from Flash Forward in LA.

Saturday morning kicked off the main conference with local user group leader
Cornel Ivanescu of CI Web Design, who organized PbD (the acronym for Powered by Detroit), speaking after breakfast. Then it was in to one of 3 sessions. I chose to talk with some of the exhibitors instead – including Macromedia User Groups, PaperThin and InterAkt. It turned out that Cornel and InterAkt are both from Romania, but fortunately for me they all spoke excellent English!

Next up were some cool talks on Flash video and streaming content. Then Macromedia's Greg Rewis gave the Day One keynote. He demoed the cool new features in Flash 8 (faster graphics, animated text, alpha channel video) and talked about Rich Internet Applications (RIAs). It is all about the experience, he said (and high sales on RIA sites!)

After lunch I took a break to walk the Henry Ford house and gardens that were opposite the hotel. Then back for some more sessions on Flex, Flash sound and e-commerce. When they finished it was time to check out the House of Fusion community suite – a place to read books, talk with fellow developers and play video games. I tried out a dance video game with Angela Buraglia (Dreamweaver book author and speaker) that involved touch sensitive dance mats and an Xbox. It was my first time with this user interface but I had a good time! I tried to get Michael and Judith Dinowitz (who run House of Fusion) to join in but they couldn't join in until after sundown.

Who's Going to CFUNITED in June?

Suitably fired up by the Community Suite I went for dinner with the InterAkt guys, Charlie Arehart, Barbara O'Neil, Rick Mason and some other attendees to the Big Fish restaurant. Again avoiding driving, we walked there – which definitely confused some

Detroit natives. Once we'd eaten it was time to hit the conference Saturday night event, which was held in the revolving restaurant at the top of the hotel. Access was via a special elevator that goes through the roof. I felt I was in a James Bond movie until I saw Ben Forta at the bar without a Martini. There was lots of ColdFusion to talk about and it seemed that the topic on everyone's lips was "Who's going to CFUNITED in June?" My informal poll of attendees suggested that more than 50% were going. I met a lot of speakers and user group managers and rotated around the Detroit skyline a few times before calling it a night.

Ben Forta's Keynote

Sunday dawned bright and early for me as I was giving the first talk of the day on Project Management. The audience gave lots of interaction for early Sunday morning and we shared many good tips on how to manage a software project. Then I checked out some sessions on CFMX 7 before lunch and the Ben Forta keynote. Cornel had planned a big entrance for Ben, so he was in a Hummer car just offstage. I joined the other user group managers in pushing the car with Ben in it on stage. As we pushed, Cornel said "Macromedia goes a lot faster with User Groups behind it."

Ben talked about some of the lesser known features in CFMX 7, including improved Verity text search and better graphing.

After lunch I saw some talks on Flex and e-Learning. Other speakers at PbD included Kevin Hoyt, Michael Dinowitz, Simon Horwith, Michael Smith, Jeffry Houser, Charlie Arehart and Shlomy Gantz. Then it was on to the first 1Flash/CFM Application Festival and Awards Ceremony. The show ended with raffling off prizes from CFUNITED, Centric Web and many books from the Community suite.

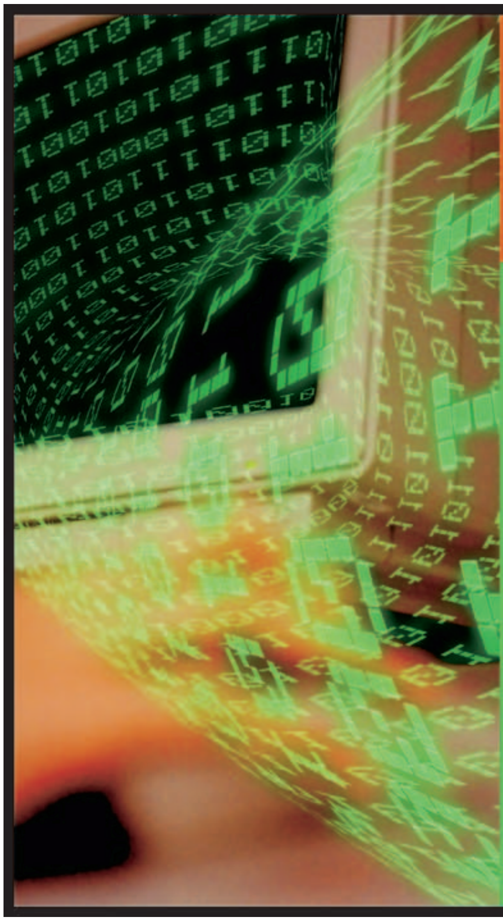
All in all it was a great show and hats off to Cornel for organizing it. See you there next year.

About the Author

Michael is president and founder of TeraTech, a 15-year-old Rockville, MD based consulting company that specializes in ColdFusion development and training. TeraTech is a Macromedia Consulting Partner and winner of the CFDJ award for best consulting company two years in a row. TeraTech runs the CFUN conference and MDCFUG. Michael has been programming for over 25 years and has been coding in ColdFusion since version 1.5.

michael@teratech.com

THE NETWORK EFFECT: J2EE AND .NET
No. 1 i-Technology Magazine in the World
JDJ
Feature: Using Apache Cactus
Testing server-side components
JAVA DESIGN + PATTERNS
FAST PERFORMANCE
PLUS...
THE WORLD'S LEADING i-TECHNOLOGY MAGAZINE WWW.SYS-CON.COM
SYS-CON MEDIA
The World's Leading i-Technology Publisher



ColdFusion U

For more information go to...

U.S.

Alabama
Huntsville
Huntsville, AL CFUG
www.nacfug.com

Alaska
Anchorage
Alaska Macromedia User Group
www.akmmug.org

Arizona
Phoenix
www.azcfug.org

Arizona
Tucson
www.tucsoncfug.org

California
San Francisco
Bay Area CFUG
www.bacfug.net

California
Riverside
Inland Empire CFUG
www.sccfug.org

California
EL Segundo
Los Amgeles CFUG
www.sccfug.org

California
Irvine
Orange County CFUG
www.sccfug.org

California
Davis
Sacramento, CA CFUG
www.saccfug.org

California
San Jose (temporary)
Silicon Valley CFUG
www.siliconvalleycfug.com

California
San Diego
San Diego, CA CFUG
www.sdcfug.org/

California
Long Beach
Southern California CFUG
www.sccfug.org

Colorado
Denver
Denver CFUG
www.denvercfug.org/

Delaware
Kennett Square
Wilmington CFUG
www.bvcfug.org/

Delaware
Laurel
Delmarva CFUG
www.delmarva-cfug.org

Florida
Jacksonville
Jacksonville, FL CFUG
www.jaxfusion.org/

Florida
Winter Springs
Gainesville, FL CFUG
www.gisfusion.com/

Florida
Plantation
South Florida CFUG
www.cfug-sfl.org

Florida
Tallahassee
Tallahassee, FL CFUG
www.tcfug.com/

Florida
Palm Harbor
Tampa, FL CFUG
www.tbmmug.org

Georgia
Atlanta
Atlanta, GA CFUG
www.acfug.org

Illinois
East Central
East Central Illinois CFUG
www.ecicfug.org/

Indiana
Avon
Indianapolis, IN CFUG
www.hoosierfusion.com

Indiana
Mishawaka
Northern Indiana CFUG
www.ninmug.org

Iowa
Johnston
Des Moines, IA CFUG
www.hungrycow.com/cfug/

Kentucky
Louisville
Louisville, KY CFUG
www.kymug.com/

Louisiana
Lafayette
Lafayette, LA MMUG
www.cflib.org/acadiana/

Maryland
Lexington Park
California, MD CFUG
http://www.smdcfug.org

Maryland
Rockville
Maryland CFUG
www.cfug-md.org

Massachusetts
Quincy
Boston, MA CFUG
www.bostoncfug.com

Michigan
East Lansing
Mid Michigan CFUG
www.coldfusion.org/pages/index.cfm

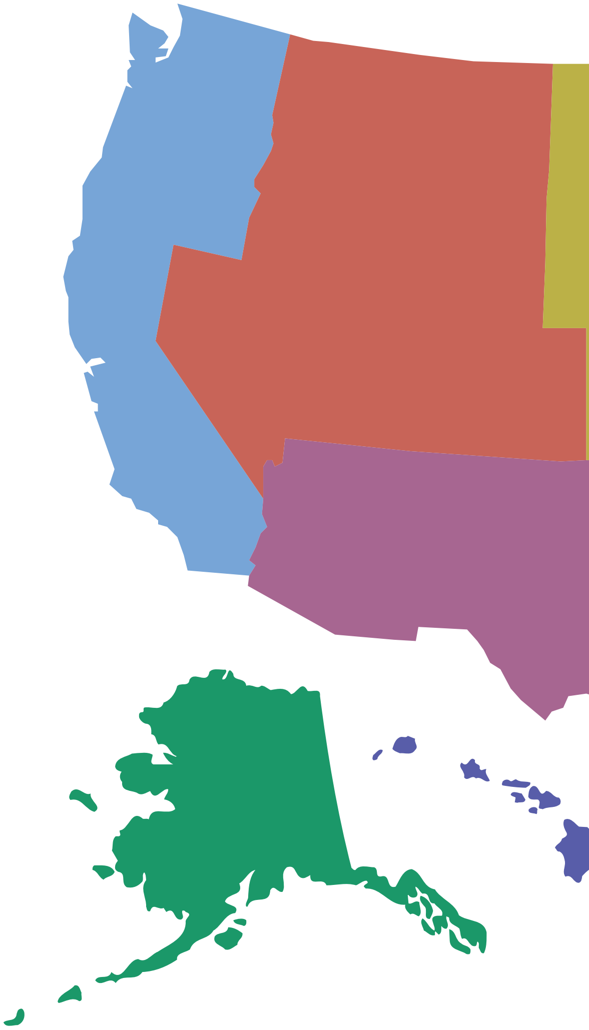
Minnesota
Brooklyn Park
Twin Cities CFUG
www.colderfusion.com

Missouri
Overland Park
Kansas City, MO CFUG
www.kcfusion.org

Missouri
O'Fallon
St. Louis, MO CFUG
www.stlmmug.com/

New Jersey
Princeton
Central New Jersey CFUG
http://www.cjcfug.us/

Nevada
Las Vegas
Las Vegas CFUG
www.sncfug.com/



New York
Albany
Albany, NY CFUG
www.anycfug.org

New York
Brooklyn
New York, NY CFUG
www.nycfug.org

New York
Syracuse
Syracuse, NY CFUG
www.cfugcny.org

North Carolina
Raleigh
Raleigh, NC CFUG
www.ccfug.org

Ohio
Dayton
Greater Dayton CFUG
www.cfdayton.com

Oregon
Portland
Portland, OR CFUG
www.pdxcfug.org

User Groups

http://www.macromedia.com/cfusion/usergroups

Pennsylvania
Carlisle
Central Penn CFUG
www.centralpenncfug.org

Pennsylvania
Exton
Philadelphia, PA CFUG
www.phillycfug.org/

Pennsylvania
State College
State College, PA CFUG
www.mmug-sc.org/

Rhode Island
Providence
Providence, RI CFUG
www.ricfug.com/www/meetings.cfm

Tennessee
LaVergne
Nashville, TN CFUG
www.ncfug.com

Tennessee
Germantown
Memphis, TN CFUG
www.mmug.mind-over-data.com

Texas
Austin
Austin, TX CFUG
www.cftexas.net/

Texas
Corinth
Dallas, TX CFUG
www.dfwcfug.org/

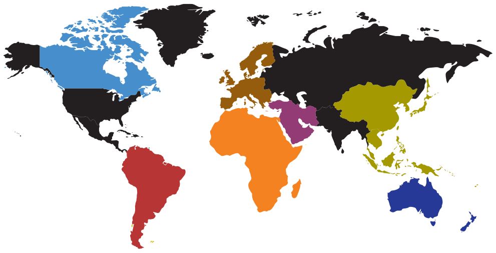
Texas
Houston
Houston Area CFUG
www.houcfug.org

Utah
North Salt Lake
Salt Lake City, UT CFUG
www.slcfug.org

About CFUGs

ColdFusion User Groups provide a forum of support and technology to Web professionals of all levels and professions. Whether you're a designer, seasoned developer, or just starting out – ColdFusion User Groups strengthen community, increase networking, unveil the latest technology innovations, and reveal the techniques that turn novices into experts, and experts into gurus.

INTERNATIONAL



Australia
ACT CFUG
www.actcfug.com

Australia
Queensland CFUG
www.qld.cfug.org.au/

Australia
Southern Australia CFUG
www.cfug.org.au/

Australia
Victoria CFUG
www.cfcentral.com.au

Australia
Western Australia CFUG
www.cfugwa.com/

Brazil
Brasilia CFUG
www.cfugdf.com.br

Brazil
Rio de Janerio CFUG
www.cfugrio.com.br/

Brazil
Sao Paulo CFUG
www.cfugsp.com.br

Canada
Kingston, ON CFUG
www.kcfug.org

Canada
Toronto, ON CFUG
www.cfugtoronto.org

Ireland
Dublin, Ireland CFUG
www.mmug-dublin.com/

Italy
Italy CFUG
www.cfmentor.com

Japan
Japan CFUG
cfusion.itfrontier.co.jp/jcfug/jcfug.cfm

Scotland
Scottish CFUG
www.scottishcfug.com

South Africa
Joe-Burg, South Africa CFUG
www.mmug.co.za

South Africa
Cape Town, South Africa CFUG
www.mmug.co.za

Spain
Spanish CFUG
www.cfugspain.org

Switzerland
Swiss CFUG
www.swisscfug.org

Thailand
Bangkok, Thailand CFUG
thaicfug.tei.or.th/

Turkey
Turkey CFUG
www.cftr.net

United Kingdom
UK CFUG
www.ukcfug.org

Consuming Complex Web Services with ColdFusion

Using Java to Bridge the Gap

By Robert Munn

One of the primary selling points of the ColdFusion platform is the ease of use of the language and the encapsulation of complex programming constructs in easy-to-use tag syntax.

In many cases, you can accomplish programming tasks that would take many lines of code in other languages in one or two lines of code in ColdFusion. Need to query a database?

```
<cfquery …>
Your database code here
</cfquery>
```

It doesn't get any easier than that. The platform's concept of encapsulating complex constructs in a simple tag-based language has held up remarkably well in most areas. In one important and growing field- Web services- the model has not been as successful as in other areas. In part, this limited success has to do with the inherent complexity of web services and the difficulty in translating that complexity into CF's tag or script-based syntax. The language provides two native methods for connecting to web services, using *cfinvoke* or *createObject():*

I. Invoking a Web service using cfinvoke:

```
<cfinvoke webservice=”path_to_wsdl_file” method=”someMethod”  attributecoll
ection=”#myattribStruct#” returnvariable=”myresult”/>
```

II. Invoking a Web service using cfobject:

```
<cfscript>
   ws = createObject(“webservice”, “path_to_wsdl_file”);
   res1 = ws.someMethod(params);
   res2 = ws.someOtherMethod(params);
</cfscript>
```

The former method allows you to access a Web service, pass in an arbitrary number of arguments in a structure, and assign the response to a variable- all in one line of code. For Web services that require relatively simple input, such as the Babelfish translation service documented in the CFMX product documentation, this model works exceptionally well. The latter method requires more lines of code, but lets you instantiate a Web service into an object and then access as many methods of the Web service as you need to fulfill the task at hand. The advantage of the latter method becomes obvious when you call more than one method at a time from a Web service. In the former method, you would write two lines of code to execute two methods, but the CFMX server would instantiate the Web service for each method call. In the latter method, you write three lines of code (ok, five if you count the cfscript tags), but the CFMX server only parses the WSDL file once.

These two methods work well for Web services that use basic datatypes for input or that make minimal use of XML attributes in their WSDL files.

Custom Datatypes

Many complex Web services use custom datatypes as part of their request and/or response model. You may have trouble modeling such datatypes in ColdFusion. The CFMX documentation instructs you to model a complex datatype as a structure. This technique works well in theory, but in practice it may lead you to building very large structures in order to satisfy the datatype that the Web service expects. Furthermore, if a Web service method you want to consume expects custom datatypes that are made up of other custom datatypes, you will need to map all of that data into a single structure to pass to the web service.

At first, that may not seem like such a burden. After all, as a software developer, you handle complex tasks all day long. Creating and handling structures is just another complex task for you to tackle. ColdFusion does not offer a lot of detail in debugging the consumption of Web services. If you fail to properly build your structure- even if you only miss one element in a very large structure- ColdFusion will return an error that it was unable to find the method you called with the parameters you specified. The server provides very little other debugging information.

That isn't even the worst part about trying to build and debug consuming a complex Web service in ColdFusion. The worst part of going through such a painful debugging process is that there is an easier way to consume complex Web services in ColdFusion.

Java to the Rescue

As of CFMX 6.0, the ColdFusion platform runs on the Java J2EE architecture. While the ColdFusion platform allows you to do lots of cool and sophisticated stuff without the complexity of Java, it also allows you to drop down into Java for problems that you just can't lick in CF directly. Normally, this sort of thing might mean writing Java code to handle complex tasks that ColdFusion just doesn't do natively- manipulating images, for instance. Ironically, the growth and acceptance of Java as a platform of choice for Web application development has led to the creation of lots of tools that Java developers can use to automate everyday tasks. Apache Axis (www.apache.org) is just such a tool that developers use to automate the process of building and consuming Web services in Java.

WSDL, or Web Services Description Language, is an XML format that describes a web service. By reading the WSDL document for a particular web service, you can tell everything about that service- the datatypes it uses for requests and responses, the methods it exposes, and how to connect to the service, among other things. There are many good books on web services that explain the WSDL format in detail if you would like more information on the subject.

WSDL2Java – Magic Widget Maker

WSDL2Java and its counterpart, Java2WSDL, are components of the Axis package. In essence, they enable you to automatically translate between Java and WSDL. Since you are not building Web services in Java, you need not concern yourself with Java2WSDL. WSDL2Java accepts a valid WSDL file as input. WSDL2Java is HTTP aware, so it will accept a URL to the WSDL file as input. From the WSDL file and any associated references the WSDL file contains, WSDL2Java creates Java classes that encapsulate the Web service described in the WSDL file. Those classes contain all the datatypes you will need, all method calls you can make, and all the classes to find and bind to the web service itself.

Using WSDL2Java with ColdFusion

CFMX comes bundled with a version of Axis running as part of the underlying Java engine, and you can leverage the bundled Axis engine in order to connect to a web service in Java. In order to use CFMX's Axis engine, you need to set your Java *classpath* to reference the Axis files that CFMX uses. The *classpath* is where the Java interpreter looks for files that are called as part of a Java program. In this case, there are several Java .jar files (packages) that you will need to reference in order for your code to work both locally and on your CFMX server. Listings 1 and 2 show a batch file that sets the *classpath* for servers running CFMX J2EE mode and Integrated mode respectively. Many thanks to Tom Jordahl of Macromedia for pointing me in the direction of the correct .jar files:

The common thread in these two listings is that these files, with the exception of jrun.jar, are located in {cfusion_root}/lib, so if these listings don't work for you, modify the paths to point to {cfusion_root} for your CFMX environment. The location of these files in your ultimate production environment is irrelevant at this point; what matters is where you are executing WSDL2Java and building your Java wrapper class. You will need CFMX running on this environment as well. I developed all of this code on my laptop, where I am running the Developer version of CFMX 7 in J2EE mode.

You will also need a Java JVM in order to call WSDL2Java. Since you are running CFMX, you already have a JVM installed, and you can use it if you like. I work with Java directly, and I have the latest Java 5 JRE from Sun, which includes a JVM, installed on my system. You can download the latest JRE from http://java.sun.com.

Now that you have a JVM installed and your classpath mapped, you can execute babelwsdl2j.bat. I recommend you copy babelwsdl2j.bat into its own folder and execute it from the command line. You could just drop the program on your desktop and click on it, but if it errors you will not be able to see the error message returned. babelwsdl2j.bat creates a set of folders and Java classes inside those folders to encapsulate the Babelfish Web service.

By convention, the outer folder will normally be something generic, like com, net, or gov, with an inner folder for the name of the organization or site that owns the Web service. Other nested folders are dictated by the structure of the Web service, as in Figure 2. That structure is a convention, not a rule. The folders could be named anything.

After you have executed the batch file, copy the resulting folder structure into a document directory. I have been

using c:\inetpub\wwwroot\java as the base folder for my development. It allows me to have my Java source code and my CFMX code in one place during development.

Eclipse – The IDE of Choice for Java Developers

If you have never programmed in Java, this next part may look intimidating, but trust your skills and you can tackle it. If you do not have Eclipse, you will need to download and install it. If you have it already, you can skip this section. Note that if you are running CFEclipse but you have only installed the minimal install package provided by the CFEclipse team, you will need to download and install a full copy of Eclipse for this exercise.

The Eclipse Foundation (http://www.eclipse.org) describes the Eclipse IDE as a platform for everything and nothing in particular. Despite the rather enigmatic description, Eclipse is far and away the most popular IDE for Java developers in use today. (Last I checked, more than 50% of Java developers in a recent survey were using Eclipse as their IDE of choice). You can download the latest Eclipse platform from their Web site. As of the writing of this article, the latest Eclipse version is 3.1 M6.

As a CF developer, you may also want to download the CFEclipse plugin (http://cfeclipse.tigris.org). You can get more information about CFEclipse from its Web site, but in short, CFEclipse is a plugin to the Eclipse IDE that adds support for CF development. It is a great plugin and well worth having whether or not you plan to use Eclipse for Java development on a regular basis.

Installing Eclipse

Download the ZIP file. Open the ZIP file and extract the contents to a folder of your choice- e.g. c:\Program Files. Double click on eclipse.exe to get started. Eclipse is will error if it can't find a JVM on your machine but if you installed the Sun JRE back in the previous section, Eclipse should be able to find it.

Setting up Eclipse

Eclipse will ask you to create a default workspace. You can accept the default or change it to suit your environment if you like. From there, you will need to create a project. Eclipse organizes everything into Workspaces and Projects. To create a project, click on New Project and select a new Java project. Uncheck the Use default workspace check box and select the location where you copied the folder structure created by WSDL2Java. Click Finish. You should see a tree view on the left of your project- including the Java classes created by WSDL2Java. You are almost done with the Eclipse setup.

Remember that *classpath* variable? It becomes very important again at this point. The base Java JRE does not include Apache Axis, and unless you specifically tell Eclipse where to find the Axis packages necessary to compile the generated code from WSDL2Java, you will not be able to compile and run your program.

Click on Project | Properties in Eclipse. Select Java Build Path on the left and click Add External JARs. Select the .jar files that you included in the batch file used by WSDL2Java. The .jar files will appear on the left in the project tree view.

You have now setup Eclipse sufficiently to write, compile, and run a Java program to access your Web service.

Building Your Wrapper Class

A wrapper class is exactly what it sounds like – a Java class that "wraps" functionality from many classes in a single class.

The Java class in Listing 3 is a simple wrapper class that allows you to call

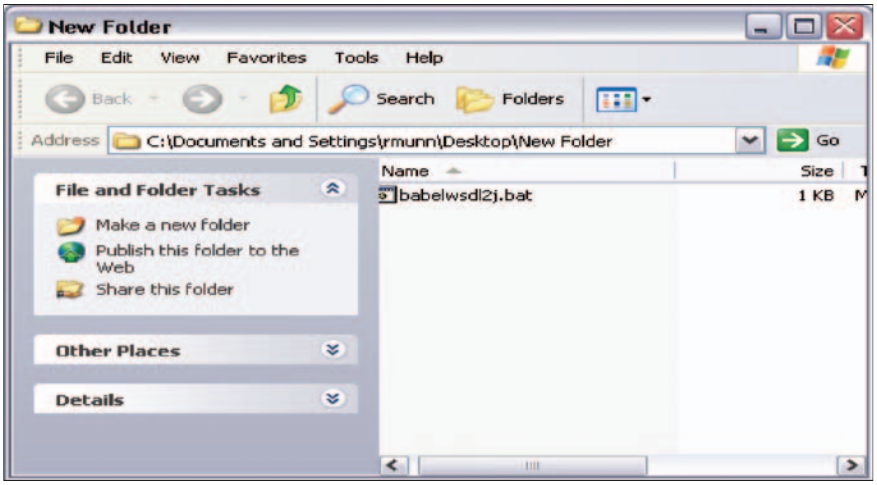


Figure 1: Run your batch file from its own folder

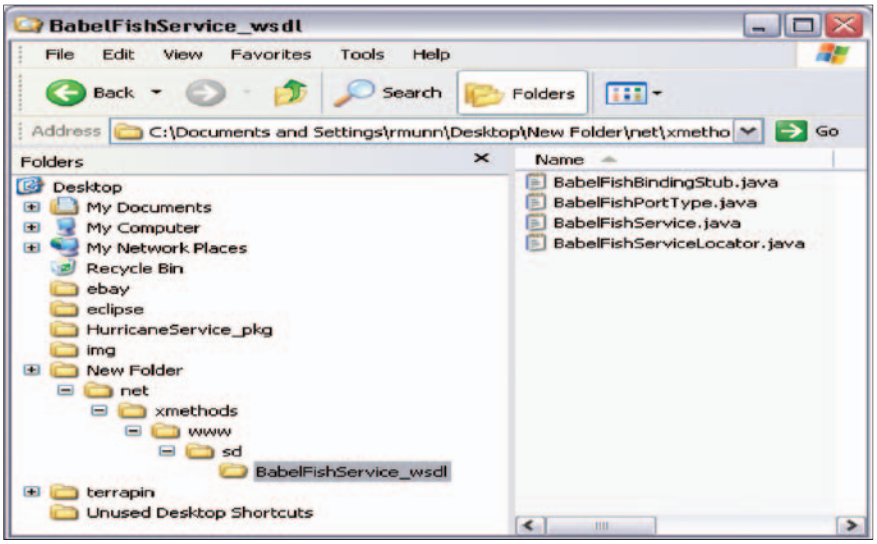
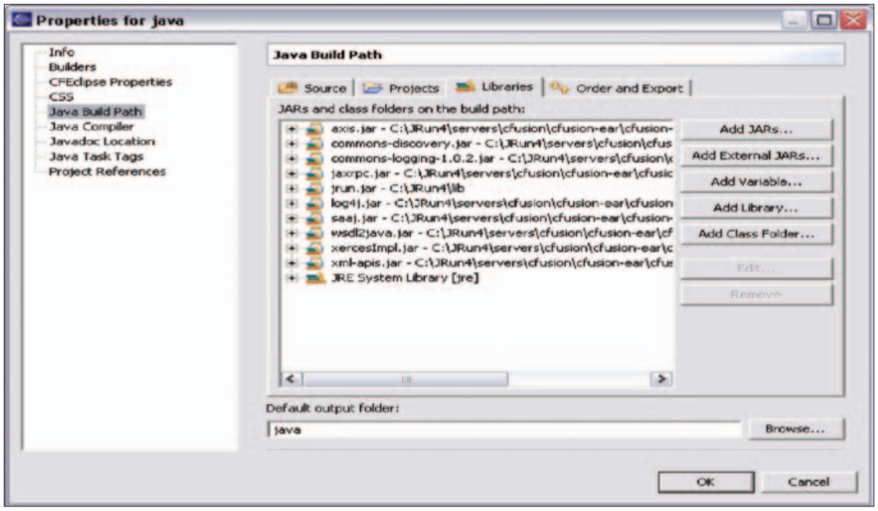


Figure 2: Folders and files generated by babelwsdl2j.bat

Connect

LINUXWORLD
MAGAZINE
Why Migrate - Linux?
A comprehensive solution for enterprise IT
Facing Microsoft's Fury...
Taking the Geek Out of Linux
with Eric Steven Raymond
The Rise of Linux in the Enterprise
Big Gains in Small Business
FREE
NOW

FREE RESOURCE CD INCLU
information
STORAGE+
SECURITY
journal
www.ISSJournal.com
In this issue:
IP Storage
SAN NAS
Disaster Recovery
Enterprise Security
Infosecurity
Microsoft
RSA
Security
SecureID for Windows
SAN
NAS
Emerging technology trends and market maneuvers



JARs and class folders on the build path:

```
axis.jar - C:\JRun4\servers\cfusion\cfusion-ear\cfusion-
commons-discovery.jar - C:\JRun4\servers\cfusion\cfus
commons-logging-1.0.2.jar - C:\JRun4\servers\cfusion\c
jaxrpc.jar - C:\JRun4\servers\cfusion\cfusion-ear\cfusic
jrun.jar - C:\JRun4\lib
log4j.jar - C:\JRun4\servers\cfusion\cfusion-ear\cfusion
saaj.jar - C:\JRun4\servers\cfusion\cfusion-ear\cfusion-
wsdl2java.jar - C:\JRun4\servers\cfusion\cfusion-ear\cf
xercesImpl.jar - C:\JRun4\servers\cfusion\cfusion-ear\c
xml-apis.jar - C:\JRun4\servers\cfusion\cfusion-ear\cfus
JRE System Library [jre]
```

Default output folder: java

Figure 4: CCOJASDirectoryWatcher event gateway instance settings

the Babelfish translation Web service. Breaking down the program, you will see two primary sections in the code. The first section is a set of *import* statements. These statements include the parts of the Java class hierarchy you will need to run your program. The first three are standard import declarations in a wrapper class like this one; you can simply copy them wholesale from Listing 3. If you don't include them, Eclipse will let you know they are missing and will even suggest adding the import statements. The next two statements import classes created by WSDL2Java to find and bind to the Babelfish service.

The second section contains the actual Java program. Java classes always start with a class declaration. In this case, *public class babel{}* is your class declaration. Everything inside the opening and closing curly braces is the core of your Java class.

Next in line is the *main()* method, *public static void main(String[] args){}*. Technically, you don't need the *main()* method in your Java class. However, including it will allow you to test your class from within Eclipse. Before you compile and copy your class files into the ColdFusion environment, you will comment out whatever you put in the *main()* method. *main()* is executed whenever a Java class is instantiated, and you don't want anything to just execute on its own when you call your class from ColdFusion.

After *main()*, you find *public babel()*, the default constructor for the *babel* class. A constructor is a method with the same name as the class itself. When the Java interpreter instantiates a new class, the interpreter calls a constructor method as part of the instantiation. For simple examples like this one, you may find that you only need an empty constructor.

Next, you see the heart of the class, *public String getBabel(String mode, String inputText)*. All of the bits of this statement have special meaning. *Public* means you can call the method from anywhere. That will be important later when you call the method from ColdFusion. *String* is the datatype returned by the method. In this case, String is a simple datatype, but in more complex Web services, the return datatype could be a custom datatype returned by the Web service. *getBabel* is the name of the method. *String mode, String inputText* are the names and datatypes of the arguments the getBabel method expects to receive. As with the String datatype that getBabel returns, the arguments are simple datatypes, but they could be anything, including custom datatypes that the Web service expects as input.

The code in *getBabel()* is the heart of your wrapper class. Look at it line by line:

```
    BabelFishServiceLocator service = new BabelFishServiceLocator();
```

This line instantiates a new object of type *BabelFishServiceLocator* called *service* using the class *BabelFishServiceLocator*. This class tells the Java program where to find the Web service you are connecting to.

```
    BabelFishBindingStub wsstub = (BabelFishBindingStub) service.getBabelFishPort();
```

For non-Java programmers, this line is a little tricky. It creates an object called *wsstub* of type *BabelFishBingdingStub*, but it uses a method of the just-created *service* object to instantiate this new object. The syntax *(class) obj.method*() tells Java to instantiate the new object using the obj.method, but that the returned object will be of type *(class)*.

```
    String res = wsstub.babelFish(mode,inputText);
```

Finally! This line calls the *babelFish()* method of the web service, passing the arguments *mode* and *inputText*. It then assigns the result to the String variable *res*.

```
    return res;
```

The last line of the *getBabel()* method returns the variable *res* to the calling application.

Now that you have a wrapper class written in Java to access the Babelfish web service, you need to write your ColdFusion code to make use of it. This code listing demonstrates how easy it is to use the Java wrapper class you have created.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>babelfish Web services example</title>
</head>

<body>

<cfset res = "">
<cfscript>
  wsObj = createObject("java","babel");
  ws = wsObj.init();
  res = ws.getBabel("en_es","Hello World");

</cfscript>
<cfoutput>
<cfdump var="#res#">
</cfoutput>
```

```
</body>
</html>
```

You use createObject() to instantiate the class. The syntax is *createObject(type,name)*. For Java classes, the type is "java", and the name refers to the name of the compiled Java .class file, without the .class extension. You can use the init() method as in this example to explicitly call a constructor in the Java class. It is worthwhile reading up on this function in the CFMX documentation if you are not familiar with it. Init() is a special method that calls a constructor for the class in question. If you do not init() a class, CFMX will implicitly call the default constructor when you access the first method of the class, but until you call init(), only static methods and properties are available from ColdFusion. (For that reason, I have made the explicit init() call in this example).

After the init() method is called, you can access the getBabel() method of the class. **res = ws.getBabel("en_es","Hello World");** uses the popular "Hello World" example to translate Hello World from English to Spanish (en_es). The results of the method call are assigned to res, and the example uses <cfdump> to output the value of res. Going to back to the Java wrapper class, you will recall that getBabel() returns a String, so you could just as easily output res using a straight output, #res#, without <cfdump>. Things get much more interesting when the Java class returns a complex datatype.

If you have never tackled a Java project before now, you may find yourself spending some time just becoming familiar with Java notation and learning about Eclipse (or whatever Java IDE you use). Java is a compiled, heavily typed language. You may need at least a basic understanding of Object-Oriented programming to tackle a Java wrapper like this one on your own. Fortunately, if you need to gain that background knowledge, you can find many resources on the Web to help you learn about OOP and Java.

The sample Babelfish application gives you a good idea of how to create the Java class files for your Web services project, how to write a wrapper in Java to access those classes, and how to access your wrapper class from within ColdFuson. If you have a project that involves connecting to a more complex Web service, especially one that uses custom datatypes, you may find several things you do not see in this example.

First, when you run WSDL2Java, you may find that it has created dozens of classes rather than the four classes created for the BabelFish Web service. You may also find a far greater number of methods available in those classes than in the sample application. Before you despair, just keep in mind that you can always go back and look at the WSDL file itself to help sort out what should go in your Java wrapper code. Most of all- persevere. Once you have managed to tackle a single project using this methodology, you will feel a lot more comfortable the next time around.

About the Author

Robert Munn has spent the last ten years designing, building, and managing Web-based applications. He currently manages the IT Web Applications group for Peregrine Systems, an enterprise software company based in San Diego. He has a degree in Classics and Spanish Literature from Tufts University. He wrote his first program in Basic at age 11 on a Tandy Model III and has been hooked on software ever since.

Robert.Munn@peregrine.coma

Listing 1 – J2EE mode

```
# babelwsdl2java.bat
# using J2EE mode
set CLASSPATH=c:\JRun4\servers\cfusion\cfusion-ear\cfusion-war\web-inf\
cfusion\lib\axis.jar;c:\JRun4\servers\cfusion\cfusion-ear\cfusion-war\
web-inf\cfusion\lib\saaj.jar;c:\JRun4\servers\cfusion\cfusion-ear\cfu-
sion-war\web-inf\cfusion\lib\commons-logging-1.0.2.jar;c:\JRun4\serv-
ers\cfusion\cfusion-ear\cfusion-war\web-inf\cfusion\lib\jaxrpc.
jar;c:\JRun4\servers\cfusion\cfusion-ear\cfusion-war\web-inf\cfusion\
lib\log4j.jar;c:\JRun4\servers\cfusion\cfusion-ear\cfusion-war\web-inf\
cfusion\lib\xercesImpl.jar;c:\JRun4\servers\cfusion\cfusion-ear\cfu-
sion-war\web-inf\cfusion\lib\commons-discovery.jar;c:\JRun4\servers\
cfusion\cfusion-ear\cfusion-war\web-inf\cfusion\lib\wsdl4j.jar;c:\
JRun4\lib\jrun.jar;c:\JRun4\servers\cfusion\cfusion-ear\cfusion-war\
web-inf\cfusion\lib\xml-apis.jar
java -cp %CLASSPATH% org.apache.axis.wsdl.WSDL2Java http://www.
xmethods.net/sd/2001/BabelFishService.wsdl
```

Listing 2 – Integrated Mode

```
# babelwsdl2java.bat
# using J2EE mode
set CLASSPATH=c:\CFusionMX7\lib\axis.jar;c:\CFusionMX7\lib\saaj.
jar;c:\CFusionMX
7\lib\commons-logging-1.0.2.jar;c:\CFusionMX7\lib\jaxrpc.jar;c:\CFu-
sionMX7\l
ib\log4j.jar;c:\CFusionMX7\lib\xercesImpl.jar;c:\CFusionMX7\lib\com-
mons-disc
overy.jar;c:\CFusionMX7\lib\wsdl4j.jar;c:\CFusionMX7\runtime\lib\jrun.
jar;c:
\CFusionMX7\lib\xml-apis.jar
java -cp %CLASSPATH% org.apache.axis.wsdl.WSDL2Java http://www.
xmethods.net/sd/2001/BabelFishService.wsdl
```

Listing 3 – babel.java

```
/*
- babel.java
- a simple wrapper to call the Babelfish translation Web service from
xmethods
*/

import java.net.MalformedURLException;
import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;

import net.xmethods.www.sd.BabelFishService_wsdl.BabelFishBindingStub;
import net.xmethods.www.sd.BabelFishService_wsdl.BabelFishServiceLoca-
tor;
```

```
public class babel

{
    /*
      public static void main(String[] args) throws ServiceException,
MalformedURLException, RemoteException

      {

           BabelFishServiceLocator service = new BabelFishServiceLoca-
tor();

           BabelFishBindingStub wsstub = (BabelFishBindingStub) ser-
vice.getBabelFishPort();

           //BabelFishPortType port = service.getBabelFishPort();
           //BabelFishBindingStub stub = new BabelFishBindingStub();

           //        babelFish req = new babelFish();
           String res = wsstub.babelFish(“en_es”,”Hello World”);
        System.out.println(“response:” + res);

      }
    */
      public babel(){
      }

      public String getBabel(
                    String mode,
                    String inputText) throws ServiceException, Remo-
teException{


           BabelFishServiceLocator service = new BabelFishServiceLoca-
tor();

           BabelFishBindingStub wsstub = (BabelFishBindingStub) ser-
vice.getBabelFishPort();

           //BabelFishPortType port = service.getBabelFishPort();
      //   BabelFishBindingStub stub = new BabelFishBindingStub();

           //        babelFish req = new babelFish();
           String res = wsstub.babelFish(mode,inputText);
           return res;
      }

}
```

Download the Code...
Go to www.coldfusionjournal.com

www.mxdevelopersjournal.com 0.9
MX developer's journal
Toppling the...
Tower of Babel with MX
Weaving Works into Tools
Understanding Levels in Fireworks
The Logo Browser Project
ColdFusion Components Under a Red Sky
Introducing Director MX
How to produce the intense application:
BabelChat

Creating Configuration Files

Make your work cleaner and easier to maintain

By Guy Rish

The new release of ColdFusion has some interesting new Java features that just beg for a return and so here we are again! As with the previous series, I'll take the time to cover the basics of these new features before I dig a bit deeper.

Event Gateways

Undoubtedly one of the coolest new Java features in ColdFusion MX 7 is event gateways – opening up a whole new set of possibilities that were, at best, difficult, and at worst, impossible, with previous versions of ColdFusion. Providing a ColdFusion-friendlier means for working with new network protocols to more mundane things like interacting with operating system services, gateways will allow you to expand your Web applications in directions that were previously inaccessible.

The product team did an amazing job of providing a number of samples that really demonstrate the capabilities of this new feature. Add to that the really excellent Macromedia Developer Center article, some exciting postings in various blogs and some advanced articles waiting in the wings for ***CFDJ*** – all in all, there's definitely more than a little material available.

So what's the goal with this series? To provide easy-to-consume walkthroughs and tutorials for doing simple things like working with the gateway pages in the ColdFusion Administrator; and to provide a solid ground about the nuts and bolts of writing gateways.

The Basics

An event gateway is just a Java class that acts like a service to extend the ColdFusion Server. It differs from things like CFX tags in that they are threaded and can provide asynchronous communication with a ColdFusion application.

A template can call a CFX tag but that template will hang around until it is done processing before continuing to the next tag in the template. The expectation is that the CFX will perform some immediate task and return. It's possible to write a multithreaded CFX tag but there is no framework for having those subordinate threads report back once the main body of the CFX has returned control to the calling template. Event gateways solve this problem.

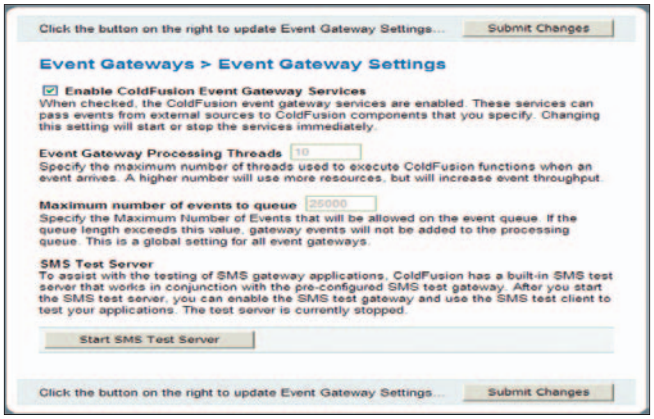
Event Gateway Administrator Pages

More than merely allowing you to configure things, the ColdFusion Administrator's **Event Gateway** pages can tell you a great deal about this new facility. As I cover the configuration pages I will introduce you to bits of gateway jargon that will be useful later when we start building applications that use gateways (for a complete list of the jargon check the product documentation; *Developing ColdFusion MX Applications, Using Event Gateways, Event gateway terms and concepts*).

Settings

The **Event Gateways Settings** page (see Figure 1) offers up a few minor, but important options, though in the Developer Edition of the product most of these settings are locked. The first is a check box for enabling gateways, which is checked by default. The second and third settings, processing threads and the size of the event queue respectively, will allow you to fine-tune gateway performance. The Developer Edition has these locked at 10 threads and 25000 events, which is quite generous for basic development.

[*Note:* Also on this page is a button to start the built-in SMS test server. You can use this sample SMS gateway and client simulator to explore more. I'm not going to cover it in this article as it is already discussed in some detail in the product documentation .



Click the button on the right to update Event Gateway Settings...

Event Gateways > Event Gateway Settings

☑ Enable ColdFusion Event Gateway Services
When checked, the ColdFusion event gateway services are enabled. These services can pass events from external sources to ColdFusion components that you specify. Changing this setting will start or stop the services immediately.

Event Gateway Processing Threads 10
Specify the maximum number of threads used to execute ColdFusion functions when an event arrives. A higher number will use more resources, but will increase event throughput.

Maximum number of events to queue 25000
Specify the Maximum Number of Events that will be allowed on the event queue. If the queue length exceeds this value, gateway events will not be added to the processing queue. This is a global setting for all event gateways.

SMS Test Server
To assist with the testing of SMS gateway applications, ColdFusion has a built-in SMS test server that works in conjunction with the pre-configured SMS test gateway. After you start the SMS test server, you can enable the SMS test gateway and use the SMS test client to test your applications. The test server is currently stopped.

Click the button on the right to update Event Gateway Settings...

Figure 1: ColdFusion Administrator's Event Gateways Settings page

Gateway Types

The **Event Gateways Gateway Types** page (see Figure 2) provides a way for you to identify gateway classes to the ColdFusion Server, thus creating a type. There are four editable fields and a check box on this page for creating a gateway type and a list of the presently configured gateway types.

The first field, **Type Name**, is the name you will give the type (this is the *gateway type* in the jargon). There are no specific requirements, such as matching the name of the Java class or such, only something that is unique and at best meaningful to the purpose of the gateway.

The **Description** field provides an opportunity to input some textual content for providing more explanatory information about a gateway. The field has a limitation of 550 characters – so try to keep it simple.

The **Java Class** field is where you enter the name of the gateway's Java class, complete with package. This field is not so frugal as the Type Name or the Description, though if you've a class name in excess of 550 characters you might consider some package reordering and less verbose naming standards. The Java class must indicate a class that is already present in the ColdFusion Server's classpath or else it will fail when you try to add the new gateway type. Since adding jars to the Server's classpath requires a restart you will want to verify this prior to adding new gateway types. New jars can be added in the **ColdFusion Class Path** field on the **Java and JVM** page in the Server Settings section of the Administrator.

The **Startup Timeout** field indicates the number of seconds that the Server will allow during the startup of an event gateway before it kills it; by default this is 30 seconds. This seems a bit excessive since most system operations occur in milliseconds – and even long operations in the thousands of milliseconds fall well below this threshold.

Out of the box the Developer Edition has seven gateway types configured (shown in Figure 2 above). These default gateways provide a number of very useful facilities that satisfy many of the both basic and things-in-vogue services.

Gateway Instances

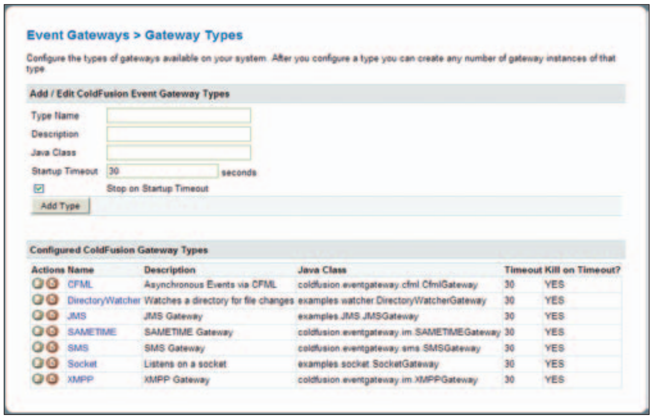
The **Event Gateways Gateway Instances** page (see Figure 3) allows you to create instances of a specific gateway type (a *gateway instance* in the jargon). This page provides three edit fields and two dropdown lists for configuring an instance and a list of the configured gateway instances.

The **Gateway ID** field allows you to specify a label name for a particular instance. This is the name (or *gateway ID* in the jargon) that will be used to programmatically reference the event gateway for sending messages. It does not have any specific requirements, though it should be unique and meaningful. The form field displayed in the browser has a limit of 550 characters which ought to be plenty even for host providers that might like to prefix a specific gateway instance name with a domain name or a corporate server prefixing with a department name.

The **Gateway Type** dropdown list allows you to select the *gateway type* the new instance will be. These values are drawn, obviously, from the types configured earlier on the **Event Gateways Gateway Types** page.

The **CFC Path** field allows you to specify a ColdFusion component that will handle the specific functionality of your gateway instance. There are no hard limitations on the length of the information entered into this form field. The CFC specified here is called a *listener CFC* in the jargon and will be discussed in greater detail in the next article.

The **Configuration File** field allows you to indicate the specific configuration file that will be applied to this gateway instance. The default installation of the ColdFusion Server keeps a number of configuration fields in the [ColdFusion install



Event Gateways > Gateway Types

Configure the types of gateways available on your system. After you configure a type you can create any number of gateway instances of that type

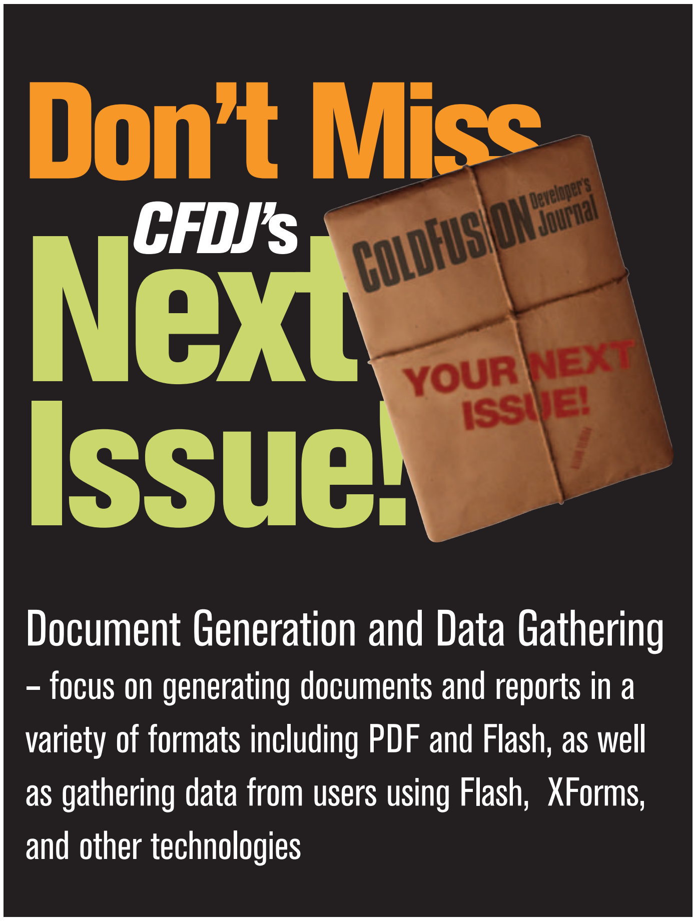
Add / Edit ColdFusion Event Gateway Types

Type Name
Description
Java Class
Startup Timeout 30 seconds
☑ Stop on Startup Timeout

Configured ColdFusion Gateway Types

Actions	Name	Description	Java Class	Timeout	Kill on Timeout?
	CFML	Asynchronous Events via CFML	coldfusion.eventgateway.cfml.CfmlGateway	30	YES
	DirectoryWatcher	Watches a directory for file changes	examples.watcher.DirectoryWatcherGateway	30	YES
	JMS	JMS Gateway	examples.JMS.JMSGateway	30	YES
	SAMETIME	SAMETIME Gateway	coldfusion.eventgateway.im.SAMETIMEGateway	30	YES
	SMS	SMS Gateway	coldfusion.eventgateway.sms.SMSGateway	30	YES
	Socket	Listens on a socket	examples.socket.SocketGateway	30	YES
	XMPP	XMPP Gateway	coldfusion.eventgateway.im.XMPPGateway	30	YES

Figure 2: ColdFusion Administrator's Event Gateways Gateway Types page



path]\gateway\config directory. The de facto file extension for an event gateway is .cfg but it is really nothing more than a Java configuration file. There is no hard requirement that gateway configuration files reside in this default directory so placing them in a subdirectory specific a ColdFusion application would be a good idea.

The **Startup Mode** dropdown list has three options; Automatic (the default option), Manual and Disabled. An Automatic gateway instance is started when the ColdFusion Server starts up and Disabled ones cannot be started. Instances with a Manual startup mode must be started from the Administrator or through the new Admin API's ***cfide.adminapi.eventgateway*** component. For more details on using the Admin API refer to the product documentation: *Configuring and Administering ColdFusion MX, Using the ColdFusion Administrator, Administrator API*.

The table of **Configured ColdFusion Event Gateway Instances** at the bottom of this page is more than merely a listing of the configured instances; it is also a control panel for starting and stopping each of the instances.

Putting It Together

So, armed with a basic understanding of how to configure a gateway, let's put together a simple example (a *gateway application* in the jargon).

I've created a directory in my Server's wwwroot for this sample; [ColdFusion install path]\wwwroot\ccojas\part1. Herein I've placed two files, *CCOJASDirectoryWatcher.cfg* (shown in Listing 1) and *CCOJASDirectoryWatcher.cfc* (shown in Listing 2).

This file is a slightly modified version of the *directory-watcher.cfg* file copied from [ColdFusion install path]\gateway\config. The first property,**directory**, defines the directory to monitor, which has been set to the sample's base directory; [ColdFusion install path\wwwroot\ccojas. The third property, **interval**, sets the number of milliseconds to wait in-between checks (basically how long the monitoring thread will sleep). The **interval** has been set to 5 seconds so you don't have to wait around a full minute before seeing activity in the logs.

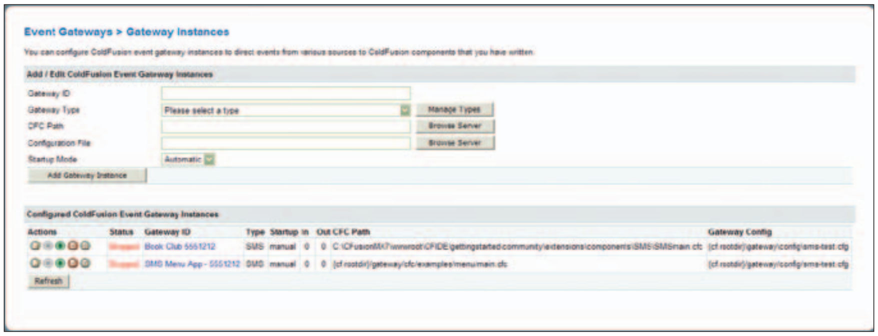
This component implements three methods which handle additions, deletions, and changes to files in the watched directory. These methods are defined by the DirectoryWatcher gateway type already configured in the Administrator as part of the default installation. Finding the documentation on a specific gateway is not yet as easy as looking up a CFML tag or function, but I expect that will change in subsequent product releases (this information can currently be found in the product documentation: *Developing ColdFusion MX Applications, Using Event Gateways, Using the example event gateways and gateway applications, Example event gateways*).

I will be discussing the specific contents of this file in the next article so I won't muddy up the example with a length discussion at this point. A cursory examination of the code will show that each of these methods merely writes a message to a log.

Once these files are in place you can create a new event gateway instance using the settings shown in Figure 4.

Since the gateway is set for a Manual startup, you'll need to start it by pressing the green play button in the gateway's control panel, **Configured ColdFusion Event Gateway Instances**, at the bottom of the page. Once the gateway is started, create a new file (I just created a text file using the context menu in the Windows File Explorer) in the watched directory; [ColdFusion install path]\wwwroot\ccojas. Looking in the ColdFusion Server's logs directory you should see a new log file called *CCOJASDirectoryWatcher.log* which should show a new last entry declaring the creation of the new text file you created with an ADD entry. Back in the watched directory, open the file and type some gibberish and save it. This too will create a new entry in the log file indicating the changed file with a CHANGE entry. And just to tidy things up, after checking the log for the CHANGE entry, go back and delete the file, which will result in a DELETE entry in the log.

[*Note:* Once you see things are working it is probably a good idea to stop this gateway instance – there is no useful reason to have a piece of demonstration code waking up to read a directory listing every 5 seconds. If you did not fill out the gateway instance information exactly and it is set for an Automatic startup (which, again, is the default option) you will want to change it to Manual. It does no good to stop the gateway if it is just going to startup automatically when you restart the ColdFusion Server.]



Event Gateways > Gateway Instances

You can configure ColdFusion event gateway instances to direct events from various sources to ColdFusion components that you have written.

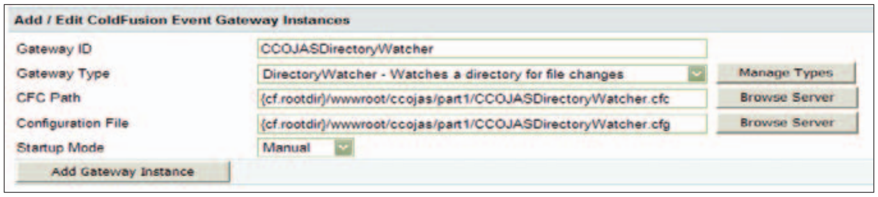
Add / Edit ColdFusion Event Gateway Instances

Gateway ID	
Gateway Type	Please select a type
CFC Path	
Configuration File	
Startup Mode	Automatic

Configured ColdFusion Event Gateway Instances

Actions	Status	Gateway ID	Type	Startup	In	Out	CFC Path	Gateway Config
	Stopped	Book Club 5551212	SMS	manual	0	0	C:\CFusionMX7\wwwroot\CFIDE\gettingstarted\community\extensions\components\SMS\SMSmain.cfc	{cf.rootdir}\gateway\config\sms-test.cfg
	Stopped	SMS Menu App - 5551212	SMS	manual	0	0	{cf.rootdir}/gateway/cfc/examples/menu/main.cfc	{cf.rootdir}\gateway\config\sms-test.cfg

Figure 3: ColdFusion Administrator's Event Gateways Gateway Instances page



Add / Edit ColdFusion Event Gateway Instances

Gateway ID	CCOJASDirectoryWatcher
Gateway Type	DirectoryWatcher - Watches a directory for file changes
CFC Path	{cf.rootdir}/wwwroot/ccojas/part1/CCOJASDirectoryWatcher.cfc
Configuration File	{cf.rootdir}/wwwroot/ccojas/part1/CCOJASDirectoryWatcher.cfg
Startup Mode	Manual

Figure 4: CCOJASDirectoryWatcher event gateway instance settings

Subscribe
Today!
SAVE 16%
12 Issues for $89.99
OFFER SUBJECT TO CHANGE WITHOUT NOTICE
AN ADVANCE LOOK AT BLAC
COLDFUSION
SPECIAL ISSUE: COLDFUSION SECURITY
COLDFUSION Developer's Journal
ColdFusionJournal.com
September 2004 Volume: 6 Issue: 9
SANDBOX SECURITY
By Alex Hubner
Security: ColdFusion Security Best Practices Knowing the security risks are there is half the battle
Bryan Murphy 16
Security: Public-Key Encryption Making strong encryption nearly painless
Wayne Graham 20
cfcUnit: ColdFusion Unit Testing Framework Paul Kenney's version
Harry Klein 30
<bf> on <cf>: Defending ColdFusion Against...
Ben Forta 36
Real-World Uses: Using ColdFusion for Network Monitoring Find the exact piece of data you need
Ryan Emerle Matthew I. Fusfield 42
Security: Top Ten Web Security Tips You can't be too careful
Michael Smith 48
Editorial Reaching Deeper Simon Horwith page 5
Community Focus Problems Securing Your Applications? Simon Horwith page 7
CF101 ColdFusion's Application Framework Jeffry Houser page 28
CFUGs ColdFusion User Groups page 40
FOUNDATIONS 'Selling' ColdFusion Hal Helms page 46
web services EDGE conference 2005 East February 15-17, 2005 Hynes Convention Center Boston, MA
SYS-CON MEDIA

The eventgateway Log

The *eventgateway.log* in the Server’s logs directory is a good way to keep tabs on the general goings-on of all your gateways. Informational details like which gateways the Server has started up as well as error specific messages about any problems that occur in a gateway.

While most errors in configuration settings will show up as errors in the ColdFusion Administrator right away, such as setting the **CFC Path** field in the **Event Gateways Gateway Instances** page to a CFC that does exist, this is still a good place to look if you are manually moving files and directories around or installing gateways from CAR files. Likewise, as you’ll see in a later article in this new series, when the Server’s gateway thread pool or message queue is exceeded it will be flagged in the log.

Coupled with the return of the series, I’m going to start a SYS-CON blog, at ccoj.coldfusionjournal.com, specifically dedicated to the content presented in the new series as well as providing links and material relating to the old series. Interestingly enough, I still get an e-mail or two every few months from people who stumble upon prints of the old CFDJ series and want to pick up the code for one of the tools presented therein or discuss one of the finer points. So as much as allowing readers a chance to provide feedback on the current series, it will give me a chance to bring relevant content update to date with the current version of ColdFusion. I hope you’ll drop by after the launch.

Wrapping It Up

Event gateways are undoubtedly going to be a major growth area for ColdFusion development just as custom tags, user-defined functions and components. Getting a solid grounding in how to administer them and how to write them will be a key set of skills for building advanced applications; opening integration opportunities that were near impossible.

The next installment in the series will look closely at the new CFML functions for passing messages to gateways and the components that handled those requests. [*Author’s Note:* All of the sample code for this article is packaged in a Zip file available at the Web site, ccojas-part1-code.zip.]

About the Author

Guy Rish is a ColdFusion and .NET developer at Vente, Inc. (http://www.venteinc.com) as well as president at Gestaltech (/www.gestaltech.com) He is an active developer and writer for various languages and technologies. He has contributed work in books on ColdFusion MX, Flash MX, and Dreamweaver MX.

grish@gestaltech.com

Listing 1: CCOJASDirectoryWatcher.cfg

```
#
# DirectoryWatcherGateway configuration file
#

# The directory you want to watch.  If you are entering a Windows path
# either use forward slashes (C:/mydir) or escape the back slashes
(C:\\mydir).
directory=C:/CFusionMX7/wwwroot/ccojas

# Should we watch the directory and all subdirectories too
# Default is no.  Set to ‘yes’ to do the recursion.
recurse=no

# The interval between checks, in miliseconds
# Default is 60 seconds
interval=5000

# The comma separated list of extensions to match.
# Default is * - all files
extensions=*

# CFC Function for file Change events
# Default is onChange, set to nothing if you don’t want to see these
events
changeFunction=onChange

# CFC Function for file Add events
# Default is onAdd, set to nothing if you don’t want to see these
events
addFunction=onAdd

# CFC Function for file Delete events
# Default is onDelete, set to nothing if you don’t want to see these
events
deleteFunction=onDelete
```

Listing 2: CCOJASDirectoryWatcher.cfc

```
<cfcomponent displayname=”CCOJASDirectoryWatcher” output=”no”>
   <cffunction name=”onAdd” access=”public” output=”no”>
      <cfargument name=”CFEvent” type=”struct” returntype=”void”
required=”yes”/>

      <cfset data=CFEvent.data>
      <cflog file=”CCOJASDirectoryWatcher” application=”No”
         text=”ACTION: #data.type#;  FILE: #data.filename#; TIME:
         #timeFormat(data.lastmodified)#”>
   </cffunction>

   <cffunction name=”onDelete” access=”public” returntype=”void”
output=”no”>
      <cfargument name=”CFEvent” type=”struct” required=”yes”/>

      <cfset data=CFEvent.data>
      <cflog file=”CCOJASDirectoryWatcher” application=”No” □
 text=”ACTION: #data.type#;  FILE: #data.filename#”>
   </cffunction>

   <cffunction name=”onChange” access=”public” returntype=”void”
output=”no”>
      <cfargument name=”CFEvent” type=”struct” required=”yes”/>

      <cfset data=CFEvent.data>
      <cflog file=”CCOJASDirectoryWatcher” application=”No”
         text=”ACTION: #data.type#;  FILE: #data.filename#; TIME:
         #timeFormat(data.lastmodified)#”>
   </cffunction>
</cfcomponent>
```

Download the Code...
Go to www.coldfusionjournal.com

New Pre-Conference Classes
Beyond Basic SQL for CF - Nate Nelson
Intro to .NET for CFers - Geoff Snowman
Designing & Developing OO Applications w/ CFCs - Hal Helms
Maximizing Code ReUse w/ CFCs & Custom Tags - Simon Horwith
"Forms and Beans" applying simple OO principles to standard CF forms methodology - Joe Rinehart
Leveraging .Net Strengths from CFML - Charlie Arehart
Advanced Cascading Stylesheets - Sandra Clark
FLiP and Fusebox walkthrough - Jeff Peters

CFML Platform Integration

Deployment

Geoff Snowman

Simon Horwith

Tim Buntel

Charles Arehart

Christian Cantrell

Jeff Peters

Manager Empowered Programming

Raymond Camden

coldfusion

Steve Drucker

Michael Smith

Ben Forta

Sean Corfield

Advanced CF

MX Integration

Michael Dinowitz

Hal Helms

Dave Watts

Shlomy Gantz

Accessibility / Usability

Boot Camp

the premier coldfusion
CFUnited
technical conference

June 29 - July 1, 2005
Washington DC Area
7th Annual ColdFusion Conference

3 full days!
New Venue
75% Bigger

www.cfunited.com

Produced by
TeraTech
Programming
www.teratech.com
301.424.3903
info@cfunited.com

macromedia
TeraTech
Microsoft
New Atlanta
CFDynamics
Fog Creek Software
Savvy Software, Inc
Interakt
Universal Mind
Paper Thin
MX developer's journal
ColdFusion Developer's Journal
HostMySite.com

CFUN has become the premier CF specific event, and Michael Smith and his team deserve all sorts of praise for their hard work in pulling it all off yet again.

Ben Forta

"...this event really is the best gathering in the world for people developing or managing CF systems. It's here that we can understand what happened, hear what's happening, and learn what's going to happen. You can't beat it."

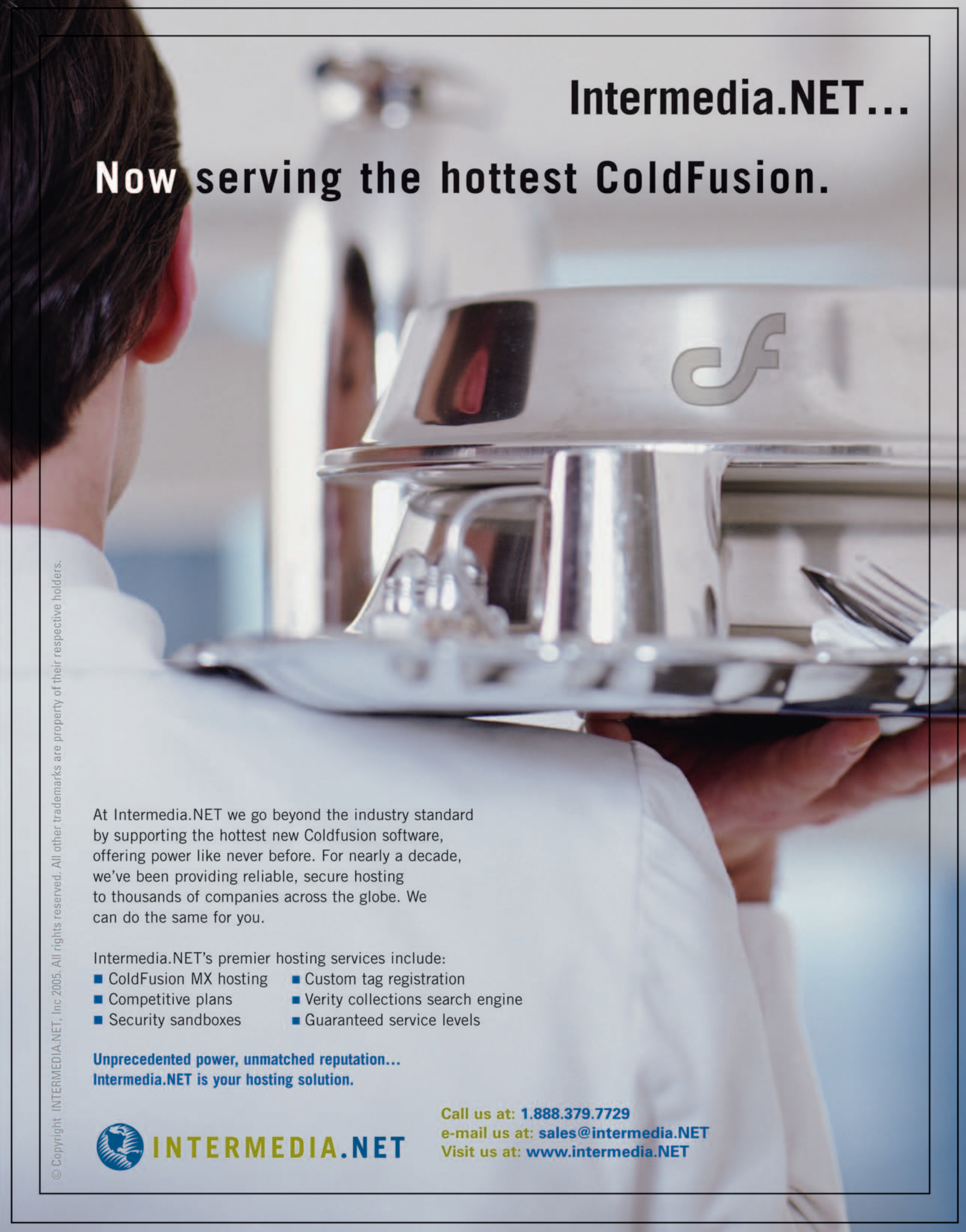
Chuck Hoffman

"Great place to network yourself and pick up new techinques and ideas. Also to meet ones peers, and see what the future holds for all those involved with ColdFusion"

Daniel Gregorio

"Introductions to the latest developing techniques. Get inspiration in new ways to develop projects. Ge'erally, to "re-kindle" your cf "fire" by being part of a group excited by and interested in cf development."

Kathleen Ballard



Intermedia.NET...

Now serving the hottest ColdFusion.

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

Intermedia.NET's premier hosting services include:

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

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

INTERMEDIA.NET

Call us at: 1.888.379.7729
e-mail us at: sales@intermedia.NET
Visit us at: www.intermedia.NET