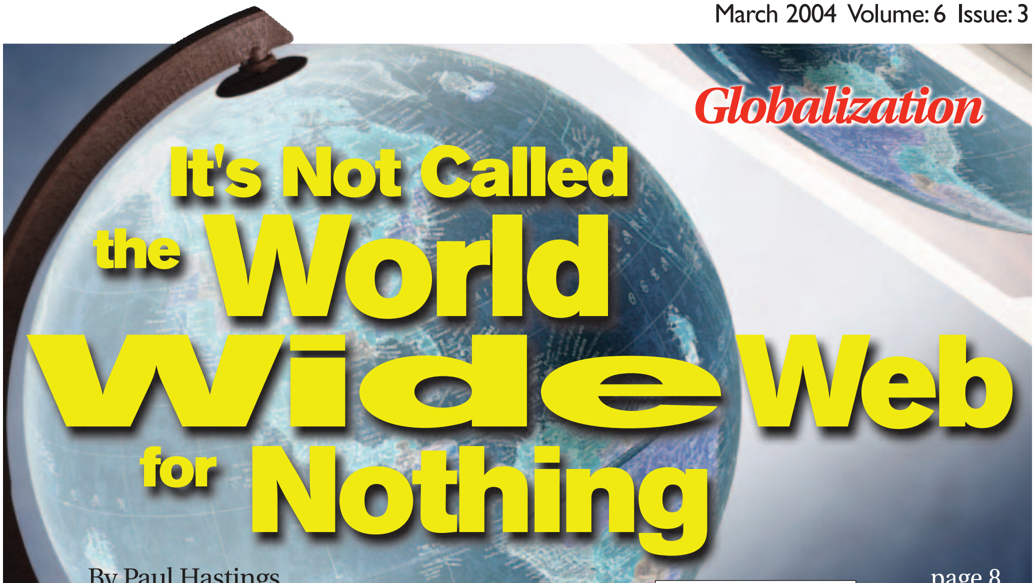
SPOTLIGHT: A PEEK UNDER THE HOOD AT JRUN

U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

ColdFusionJournal.com

March 2004 Volume: 6 Issue: 3

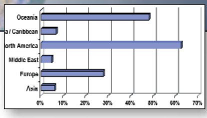


Globalization

It's Not Called the World Wide Web for Nothing

By Paul Hastings

page 8



RETAILERS PLEASE DISPLAY UNTIL MAY 31, 2004
$8.99US $9.99CAN
0 71486 02689 1 04

SYS-CON MEDIA

For the greatest hits
of the 70's, 80's and 90's
call your web host's
tech support.

CFUN04

MDCFUG
301.424.3903
info@teratech.com

Sixth Annual ColdFusion Conference

Two whole days of...
Networking! Learning! Fun!

June 26th & 27th, 2004

Charlie Arehart
Jo Belyea-Doerrman
Raymond Camden
Christian Cantrell
Sandra Clark
Sean Corfield
Robert Diamond
Michael Dinowitz
Steve Drucker
David Epler
April Fleming
Ben Forta *tentative
Shlomy Gantz
Critter Gewlas
Mark Gorkin
Hal Helms

Simon Horwith
Larry Hull
Chafic Kazoun
Matt Liotta
Tom Muck
Rey Muradaz
Nate Nelson
Samuel Neff
Jeff Peters
John Quarto
Neil Ross
Stephen Shapiro
Michael Smith
Geoff Snowman
Jeff Tapper
Dave Watts

5 TRACKS!

Bootcamp - Basic ColdFusion and Flash topics
Advanced - Advanced ColdFusion topics
Empowered - Fusebox & Project management topics
Accessibility - making sites that disabled people can use, section 508
Integration - Flash, Flex and other technologies integrated with CF topics

$199
Per Person
Special Price until 3/31/04

www.cfconf.org/cfun-04/

"I learned numerous techniques last year that have helped myself and our development team to better deliver quality products to our customers. CFUN is also a great venue to meet some of the names you see in CFDJ and DevNet and talk with them one on one."

-Phillip D

COLDFUSION Developer's Journal

editorial

editor-in-chief
Robert Diamond robert@sys-con.com
technical editors
Raymond Camden raymond@sys-con.com
Simon Horwith simon@sys-con.com
executive editor
Jamie Matusow jamie@sys-con.com
editor
Nancy Valentine nancy@sys-con.com
associate editors
Gail Schultz gail@sys-con.com
Jean Cassidy jean@sys-con.com
Jennifer Van Winckel jennifer@sys-con.com
research editor
Bahadir Karov, PhD bahadir@sys-con.com

production

production consultant
Jim Morgan jim@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

contributors to this issue

Selene Bainum, Benjamin Blackwell, Robert Diamond,
Paul Hastings, Hal Helms, Simon Horwith,
Jeffry Houser, Greg Stewart

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.
SYS-CON Publications, Inc., reserves the right to
revise, republish and authorize its readers to use
the articles submitted for publication.
All brand and product names used on these pages
are trade names, service marks, or trademarks
of their respective companies.

ColdFusion – Proven Choice

By Robert Diamond

Macromedia has started an interesting new promotional campaign for ColdFusion that proclaims a whole slew of statistics and gives a partial list of some of the big names using ColdFusion solutions. You can see it on their Web site at http://macromedia.com/software/coldfusion/proven. They tout many of the impressive numbers that we've reported here as they've come out, including more than 300,000 developers at over 10,000 companies worldwide, more than 125,000 ColdFusion servers deployed, relied on by 40% of the Media Metrix Top 50 Web Properties, more than 19.7 million CF pages, and a bunch more. It's worth checking out if you need facts and figures to convince a client and/or boss that ColdFusion is the way to go.

This is very good to see because the success or failure of ColdFusion as a product, and as an industry, is closely tied to the support and promotion that Macromedia (along with other companies in the CFML space, including New Atlanta) and the application vendors put behind it. Equally important, though, to sustaining a healthy industry is the community, and Macromedia's note of "thousands of third-party components and add-ons" hits the nail on the head perfectly.

Those of us who follow the software and development industry on a daily basis will tell you unequivocally that the lifeblood of development software is the level of interest, activity, and passion from its community – and one of the best ways to measure that is through the work that they're churning out for public as well as private consumption. If developers aren't active and developing, then there's not much of an industry. It's a great thing that there are two CFML servers on the market, and countless tools – from content management servers to bulletin boards to shopping carts – so many that we could fill an entire book with them, never mind an editorial column.

Aside from all these higher-level concerns about the industry, an active add-on community offers a great advantage – and that's that it makes all of our lives easier. Why reinvent the wheel if it's rolling along fine? I myself have saved many hours of time by utilizing custom tags, and I know that most of you out there have as well.

There are several sites out there offering CFML add-ons, starting with Macromedia's own ColdFusion Exchange at www.macromedia. com/cfusion/exchange/index.cfm, which has 3,988 entries as I'm writing this, and it wouldn't surprise me if it breaks 4,000 by the time you read it. (Another note about Macromedia's exchange is that you can find equally great add-ons for Flash, Dreamweaver, and other MM products.)

Another great site is CFTagStore.com, which proudly proclaims that it's "Built by ColdFusion Developers, for ColdFusion Developers" – and it is! It's a pretty unusual site because it allows anyone who's built a CF custom tag or component to post it on the site and to make some money off of it. These are two great places to start looking; between the two sites you should be able to find nearly anything that's available out there, but if all else fails – Google! And if you still can't find what you're looking for, build it yourself and upload it. We can never have too many tags and components out there.

About the Author

Robert Diamond is vice president of information systems for SYS-CON Media, and editor-in-chief of **ColdFusion Developer's Journal.** *Named one of the "Top thirty magazine industry executives under the age of 30" in* Folio *magazine's November 2000 issue, Robert holds a BS degree in information management and technology from the School of Information Studies at Syracuse University. Visit his blog at* www.robertdiamond.com.
robert@sys-con.com

FREE TRIAL

ON VIRTUAL DEDICATED HOSTING*

FEATURES

- **ColdFusion Administrator Access**
  Receive full access to the ColdFusion Administrator!
- **Remote Management of Server**
  Use Terminal Server to manage your server at the root level.
- **Custom Tag Support**
  Get unrestricted tag use and ColdFusion RDS support.
- **Support for Multiple Websites**
  Host up to 50 domains. The first 10 are free.
- **Unlimited Database Support**
  Includes 100MB SQL space and unlimited DSNs.

WHY SHOULD YOU SWITCH?

- **MORE FLEXIBILITY**
  You can install custom tags and software.
- **MORE RELIABILITY**
  Your site is not affected by others on the server.
- **COST SAVINGS**
  Save by consolidating your domains into one plan. Get 5GB space for up to 50 domains.

macromedia®
ALLIANCE PARTNER

CALL US TODAY TO ACTIVATE YOUR FREE TRIAL!

No obligation to buy!

- 866-CFX-HOST
- www.cfxhosting.com/vpstrial

ColdFusion Developer's Journal
CFDJ
READERS' CHOICE AWARD

Our servers are housed in a fully-redundant data center, monitored 24/7 by experienced technicians.

Supported Software
Latest version of CFMX (v 6.1)
Windows 2003 or 2000
Access and SQL supported

CFXHosting

www.cfxhosting.com 866-CFX-HOST

* Free setup & first month free. Just $189/month thereafter.

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
technical director
Alan Williamson alan@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
director of sales & marketing
Megan Mussa megan@sys-con.com
advertising sales managers
Alisa Catalano alisa@sys-con.com
Carrie Gebert carrieg@sys-con.com
associate sales managers
Kristin Kuhnle kristin@sys-con.com
Beth Jones beth@sys-con.com

sys-con events
president, events
Grisha Davida grisha@sys-con.com
conference manager
Lin Goetz lin@sys-con.com
national sales manager
Sean Raman raman@sys-con.com

customer relations
circulation service coordinators
Shelia Dickerson shelia@sys-con.com
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com
Merline Noel merline@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
Christopher Croce chris@sys-con.com

online editor
Lin Goetz lin@sys-con.com

accounting
financial analyst
Joan LaRose joan@sys-con.com
accounts receivable
Charlotte Lopez charlotte@sys-con.com
accounts payable
Betty White betty@sys-con.com

subscriptions
Subscribe@sys-con.com
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

Tales from the List

Putting an end to the Query of Queries stigma

When Macromedia introduced Query of Queries functionality to the CFML language back in ColdFusion 5, its poor performance quickly became common knowledge. In fact, I personally feel that Query of Queries got a bad rap, and the stigma continues to this day.

By Simon Horwith

In this month's column I will quickly explain what Query of Queries is, why its introduction in CF5 was so important, what its initial shortcomings were, and whether or not Query of Queries deserves a second look in ColdFusion MX. As more developers begin to investigate the use of Query of Queries in ColdFusion MX 6.1, many have turned to the *CFDJ List* and other lists for answers.

Query of Queries ("QofQ" for short) refers to the ability to run an SQL statement (a query) against an existing record set in memory. Prior to its introduction, there were only two ways to filter record set data. One was to loop over the record set in memory and copy the filtered data to another variable while performing conditional logic for filtering. The other method was to pass another SQL statement to the database in order to create a second record set in memory. For obvious reasons, neither method is elegant from a performance or coding point of view.

Query of Queries promised to allow data to be re-sorted, filtered, or joined with other record sets using the same SQL syntax developers are already familiar with – and without having to write any code other than a single call to the <cfquery> tag. This functionality was highly anticipated for two primary reasons: the ability to easily join data from one datasource with data from another, and the ability to run an SQL SELECT statement against data in memory, which is very useful in an application and theoretically should perform much better than making calls to the database. For those of you not familiar with Query of Queries, in order to create a QofQ, simply place a <cfquery> on a page that already has a query on it, give the <cfquery> tag dbtype attribute the value "query" instead of specifying a datasource, and in the SQL statement use the name of an existing record set in lieu of table names.

Query of Queries is a good idea – after all, ColdFusion can obviously read from server memory faster than it can from a database… right? That's what everyone thought before ColdFusion 5 came out, and we were eager for the ability to put a record set in memory once and repurpose that data rather than querying the database again and again. Unfortunately, when ColdFusion 5 came out Query of Queries generally took longer to execute than making another call to the database. The functionality was blacklisted by most developers as something to be avoided in CFML applications.

When ColdFusion MX was released, so much attention was paid to the new Java engine, the new component framework (CFCs), XML parsing, and other new functionality that at first little attention was given to Query of Queries. It's taken a while, but CFML developers have finally caught on to the fact that Query of Queries performs very well in ColdFusion MX and ColdFusion MX 6.1 – and

—continued on page 19

About the Author

Simon Horwith is co-technical editor of **CFDJ**, *and chief technology officer of eTRILOGY Ltd., a software development company based in London, England. Simon has been using ColdFusion since version 1.5 and is a member of Team Macromedia. He is a Macromedia Certified Advanced ColdFusion and Flash developer and is a Macromedia Certified instructor. In addition to administering the* CFDJ List *mail list and presenting at CFUGs and conferences around the world, he has also been a contributing author of several books and technical papers.*
simon@horwith.com

It's Not Called the 'World Wide Web' for Nothing

Build and deploy your CF applications for a global market

Much of the future growth of the Internet will occur in areas where the dominant language is not English. The convergence of this growing area of opportunity with the new features offered by CFMX make CFMX an excellent choice for globalized applications.

By Paul Hastings

This is the first in a series of articles on globalizing ColdFusion MX (CFMX) applications. Why CFMX? CFMX is really the first version of CF that allows developers to "go to town" with globalized applications. Let me list the three main reasons why I believe this is so:

- It's based on Java, so we can now finally use Unicode for text data (don't worry about what that means right now; just understand that it's really darned important – I'll get to what it means in a bit).
- It's based on Java, so when required we can simply reach down and make use of native Java classes to supplement CF's functionality. This is especially important for locales outside the officially supported CF subset (again, don't worry about what a "locale" is, I'll get to that shortly).
- It's based on Java, so we can more easily make use of the widely available Java globalization libraries like IBM's excellent ICU4J (http://oss.software.ibm.com/icu4j) for things like non-Gregorian calendars, holidays, locale specifics, and so on.

Why now? There are an ever-increasing number of communications on e-mail lists, developer forums, and blogs requesting information about globalization. Quite often the people requesting this information don't know they are dealing with globalization issues; sometimes all they want is for their CFMX application to start speaking French instead of "Martian." These communications have risen enough above the "background noise" level that ***CFDJ*** executive editor Jamie Matusow and technical editors Ray Camden and Simon Horwith thought it was a good idea to do some communicating of their own.

The purpose of this article is to:

- Present you with the "why" of globalizing CFMX applications (basically, I hope to entice you into jumping aboard the globalization bandwagon);
- Introduce you to the terminology used in this field (so you too can jargonize your proposals and pepper your e-mail with acronyms and abbreviations);
- Start you thinking about some of the more important concepts of CFMX globalization, so you can begin planning your own launch onto the world's stage. Globalization does indeed take a fair bit of planning and consideration to save yourself long-term grief.

Since this is more of an introductory and informational article, don't expect much in the way of CF code. I'll reserve the code for later articles that will deal with the specifics of globalizing CFMX applications.

Why Globalize?

The primary answer to that question, like most things in life, is of course money. Let me lay out some facts and figures. It's been predicted that online transactions in 2004 will reach $6.8 trillion with the U.S. having about 47% of that. While 47% is quite a decent chunk, that's actually a significant (and continuing) decline in share from 74% in 2000. That means there are $3.6 trillion in online transactions going on outside the U.S. – and that amount is growing. More important, in my opinion, those transactions aren't being conducted in one monolithic market. They're spread out among dozens, perhaps hundreds, of countries, most of which don't use English and almost none of which use U.S. dollars as their currency.

Since the financial volume of online transactions has a lot to do with per capita income (i.e., Americans simply have more income to spend online), which will eventually become more evenly distributed, it might be useful to examine raw global Internet usage (or eyeballs). Figure 1 shows Internet usage by region. As you can see, Internet users outside North America number more than double those within that region. The Asian region is the current Internet user champion.

If we examine Internet usage growth from 2000 to 2003 (see Figure 2), we see a similar trend – the fastest growth is now found outside North America. The Middle East, of all places, has seen more than 100% growth during this period. Perhaps this is something to keep an eye on; I'm fairly sure the marketing mavens are doing so.

Figure 3 illustrates Internet penetration as a percentage of total population by region. From this data it's clear that while North America is fast approaching total saturation (I can attest to that fact from personal experience; even my long-suffering dear old Mom now makes regular use of the Internet), there's still huge potential in the rest of the world, even in Europe. I'm no marketing expert, but it seems to me this is something significant to consider.

If this information doesn't exactly bring tears to your eyes, perhaps you might consider this simple fact: there are more Spanish speakers living in the U.S. than there are Canadians in Canada. The 38 million Hispanics in the U.S. (estimated by the Census Bureau) have a purchasing power of around $675 billion. You don't have to go to Hong Kong to understand the need for globalization; just take a look around your back yard.

Let's look at one last Internet statistic that further illustrates this point, weblogs (blogs). The NITLE Blog Census shows similar trends in the blog space. Of the 1.7 million blogs surveyed, about 62% were considered to be in English. I find it particularly interesting that the next most popular blogging languages are Portuguese, Farsi (Persian), and Polish. I'm even more surprised to see that there are over 3,000 blogs in Esperanto.

Summing up, it appears that most future Internet growth will be occurring outside the North American region and will involve heterogeneous locales. To me, this just cries out for globalized application development. As CF developers, we all understand how important the Internet is; we just need to focus on *where* it's important.

Terminology

Now that we've covered the need for globalized applications, it's time I defined some of the jargon I've been tossing around and introduce some of those acronyms I promised.

Internationalization

Internationalization (a.k.a. I18N, an abbreviation for the 18 letters between the "I" and "N" in internationalization) refers to the design and development of a CFMX application so that its core functionality is not based on a single locale or language. You can think of I18N as making an application locale or language neutral by removing all obstacles to it being deployed in more than one locale.

Localization

Localization, (a.k.a. L10N, an abbreviation for the 10 letters between the "L" and "N" in localization) is the post-I18N process of adapting an application to a specific locale without any changes to its source code. L10N is the process of applying a locale or language "skin" to an already I18N CFMX application.

Globalization

Globalization (a.k.a. G11N, an abbreviation for the 11 letters between the "G" and "N" in globalization) is sometimes used as a synonym for I18N (mainly by Microsoft), but to me it's L10N implemented across several locales after I18N.

Locale

Locale (nope, no nifty abbreviation) refers to the most elementary part of globalization. Locales are languages and other cultural norms (calendars; date, number, and currency formatting; spelling; measurement systems; page sizes; and so forth) specific to a geographic region. HTML and XML both rather plainly define locales as "(language, country)," that is, primarily as a language identifier. Java and ColdFusion (in a roundabout fashion) include other cultural information that is specific to locale.

Codepage Encodings

Codepage encodings date from a time when software developers lived in caves and did all their coding by torchlight (in my opinion at least), so I am loath to even mention them, though for

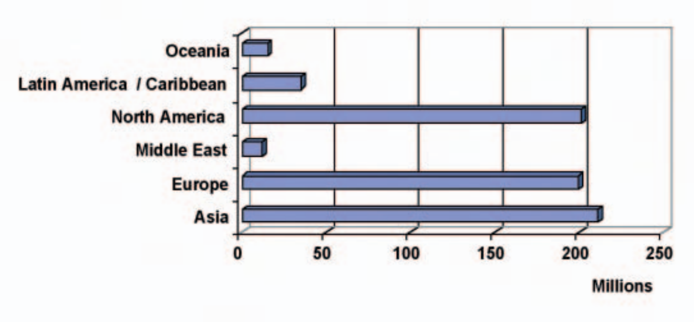


Figure 1: Internet usage by region

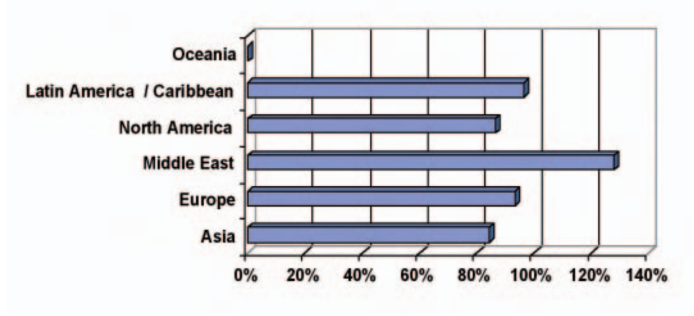


Figure 2: Internet usage growth (2000-2003) by region

the sake of thoroughness I will. A codepage is also known by various other names, including encoding, charset, character set, coded character set (CCS), graphic character set, and character map. Microsoft defines a codepage encoding as a list of selected character codes in a certain order.

Codepages are defined for specific languages or groups of languages that share common writing systems. A codepage (in Windows at least) can contain only 256 code points; because of this codepage codes overlap (usually in the upper 128 code points) from one language to the next. You simply cannot mix languages (say Thai and French) in the same text stream because of this overlap.

Things become even more complicated when dealing with Asian languages such as Chinese, Japanese, and Korean (which you will often see abbreviated as CJK) since these languages contain more than 256 characters. DBCS (double-byte character sets) were created to handle these types of languages (which still need to be based within the 256-character limitation). Each DBCS character is represented by a pair (double-byte) of code points, which rendering systems need to interpret as a single character. Even with all this mess, DBCS codepages still overlapped. All of this severely limits the usefulness of codepage encoding for globalized CFMX applications.

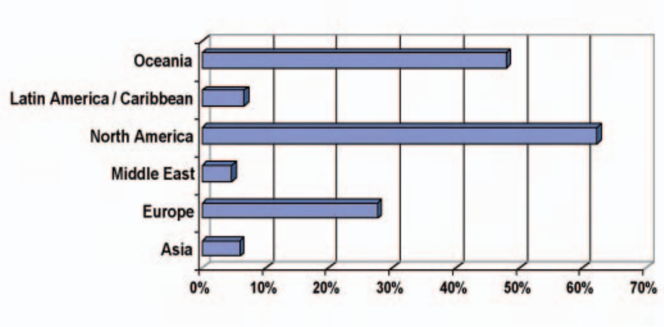


Figure 3: Internet penetration (percentage of population) by region

Unicode

Unicode is an encoding standard that provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language. Unicode is especially well suited to the Internet, since the global nature of the Internet requires applications/solutions that work in any language. The World Wide Web Consortium (W3C) has recognized this fact and now expects all new RFCs (Requests for Comments) to use Unicode for text. Unicode is the de facto character encoding standard for all major computer companies, while ISO 10646 is the corresponding worldwide standard approved by all ISO member countries. Not to worry, these two standards have identical character stocks and binary representations. Unicode: don't leave home without it.

Coordinated Universal Time

Oddly enough, Coordinated Universal Time is abbreviated as "UTC." For all intents and purposes it is the same as Greenwich Mean Time (GMT), which is the global standard for time established in October 1884 at the International Meridian Conference when delegates from 25 nations (including Hawaii and Liberia) met in Washington, DC, and agreed on a system. GMT/UTC refers to time kept on the Greenwich meridian (longitude zero). This time scale is kept by several time laboratories around the world (for instance, the U.S. Naval Observatory) and is determined using highly precise atomic clocks, rather than by observing astronomical phenomena.

CFMX Globalization Concepts

The final section of this article will introduce some of the more important concepts associated with developing G11N CFMX applications. Later articles will more fully develop these ideas with CF code examples.

Unicode

I really can't state this any more plainly; it's critical that every aspect of your CFMX application be Unicode capable – anything less will eventually be fatal. You simply cannot expect to efficiently deploy and manage a globalized CFMX application using codepage encodings. Your database should be fully Unicode capable; if it's not, dump it. This, however, should not be a major issue, as most of the modern big-iron databases such as MS SQL Server, Oracle, and their brethren – as well as some of the more popular desktop databases such as MS Access 2000 – fully support Unicode. Finally, since UTF-8 is CFMX's default encoding, it's just common sense to use Unicode. In practical terms, this means you won't have to worry about inserting/retrieving text data from your database, displaying it, or transferring to other formats (XML, etc.).

Locale

It's vital that your CFMX application be able to transparently determine a user's locale – though it's good practice to always provide a method for a user to manually swap locales – and provide locale-specific functionality such as text translations and object (date/time/numeric/currency) formatting. Locales are, after all, the basic building blocks of globalized CFMX applications. Your application needs to get a user's locale right.

While CFMX supports a hefty subset of the available Java locales, it's just that, a subset. For instance, it doesn't support the locale in which I live (Thailand, th_TH). Because of this, your application should be prepared to make use of Java locale-based classes to supplement the missing locales. For the sake of consistency you should therefore also use Java-style locale notation such as "en_US" (two-letter language code followed by two-letter country code, sometimes followed by a two-letter variant) instead of the CF style of "English (US)". Since switching between CF and Java locale–specific functions requires extra overhead and management, I suggest using the Java-based functions until CFMX supports all Java locales.

String Handling

If you're used to building applications for one locale, you probably have hard-coded application text and its presentation right alongside your CF code. While that might work for one locale, this approach will surely fail once your application needs to handle several locales.

continued on page 46

INTO

WHAT ARE YOU INTO?

At Macromedia, we're continually inspired by the passion of our customers. Visit "Into" to see their stories, or share your own. **www.macromedia.com/into**

Announcing Macromedia MX 2004:
New versions of Macromedia® Flash™, Dreamweaver®,
Fireworks® and Studio. Run with it.

Using JRun Instances for CFMX Deployment

Reap the benefits of the underlying J2EE architecture that JRun provides

By Greg Stewart

One of the unheralded milestones of ColdFusion MX 6.1 is the ability to deploy ColdFusion as an EAR (Enterprise Archive) on JRun 4.

Although a lot of attention has been focused on the improved execution time of ColdFusion applications, the reliability of CFCs, and the enhanced e-mail features, the 6.1 release also introduced many of us to JRun, Macromedia's J2EE server. I won't delve into JRun as a J2EE server. Rather, I'd like to share the reasons for using JRun instances for ColdFusion deployment, and my experiences with deploying CFMX on JRun as an EAR, configuring the instances with Apache and IIS, and configuring individual JVM settings for each instance.

First things first, though: What is an instance? An instance is one J2EE application server service running on a machine. In other words, I can run five JRun servers – each hosting their own applications – with a single installation of JRun. An instance, in the context of CFMX, can be regarded as a copy of CFMX. Multiple instances allow you to have multiple copies of ColdFusion running on a single server without having to buy additional licenses.

All you need is an instance of CFMX 6.1 Enterprise installed as an EAR and running on a J2EE application server (we'll use JRun, which is bundled with ColdFusion) using either Apache or IIS as the Web server. For more information read Tim Bruntel's excellent article introducing multiple instances: www.macromedia.com/devnet/mx/coldfusion/articles/multiple_61.html.

Running multiple copies of CFMX on your server offers many advantages. For starters, you can optimize applications, and each instance can have its own CF Admin. Now you have the ability to configure ColdFusion server settings and no longer have to share global settings, as you can set these on a case-by-case basis, for example, if you want to change the XML parser for one particular site, but not for others. Wouldn't it be useful if you could control, say, the JVM heap size? Or how about controlling the number of simultaneous requests? In short, each application can be fine-tuned according to its requirements.

High availability is another factor. I think a lot of us have dealt with a shared hosting environment in which one site was threatening to bring down ColdFusion. Whether because of poorly written code or slow-running queries, that dreaded moment of having to restart ColdFusion came, taking all the other sites offline with it. With instances that is no longer the case. Now you can create multiple instances of ColdFusion on one server and assign them to one site (or even multiple sites to one instance), thus sharing the load. Then if something does go horribly wrong, only one instance is affected.

Running applications in separate instances isolates them and limits their access to only the shared scopes (application, server, and session) of their own instances. Furthermore, developers can take advantage of the ability to set up instance-specific sandbox security to enhance the security of their applications.

Another advantage is in the area of clustering. In earlier versions of ColdFusion you could cluster separate servers to load-balance applications. Now you can create several server instances – each running the same ColdFusion application on the same physical server – and cluster the instances, thus reaping the advantages of application failover. Finally, I came across another benefit of instances that I gleaned from the CFCDev listserv (hat tip to Sean Corfield). As you can deploy CFMX EARs across instances, you can package these tailored instances and then redeploy them on another instance or even a different server running JRun. For more on this, take a look at Brandon Purcell's article, "Advantages of using multiple instances for ColdFusion MX for J2EE," which covers the benefits of using instances in more depth: www.macromedia.com/devnet/mx/coldfusion/j2ee/articles/multiple.html.

To underline my earlier points, I'd like to share a few scenarios that prompted my company to use instances. The first relates to reliability: one project I have recently been involved with was split into two applications, the news site and the jobs site. Initially we ran them on two separate servers, but to save some costs – without sacrificing the stability of either site – we opted to give each application its own instance. Furthermore, we are currently considering adding a third instance and clustering it with the more popular of the two sites.

Another good use of instances arose when I needed to use a SAX-based XML parser for a project that was different from the one that shipped with CFMX 6.1. Since I had existing applications that were using the original XML parser, I did not see the point in having to port these just to accommodate the new application. As I was already running CFMX on top of JRun, creating a new instance and configuring it to use Xalan made perfect sense. Using instances to test beta releases or new code is another way to help protect your existing applications from configuration changes or beta software.

I'll only outline the steps involved in order to give you a brief overview. For detailed instructions, I recommend that you look at Purcell's excellent article on installing CFMX as a J2EE application (www.macromedia.com/devnet/mx/coldfusion/articles/multi_instances.html). I'll assume that you already have IIS or Apache installed. Furthermore, where I refer to "cfusion2", this is the instance name I selected; you can, of course, substitute your own instance name. Just remember to modify the commands to reflect your instance name choice.

1. Install JRun using ColdFusion MX 6.1 with the J2EE Configuration option and the built-in Web server.

2. Once the install is complete, i.e., JRun and the CFMX version are up and running, stop the existing CFMX instance service and copy the cfusion-ear folder.

```
{install path\JRun4\servers\cfusion\cfusion-ear}
```

3. Create a new JRun server instance from the JRun management console, and copy the previously mentioned cfusion-ear folder into the newly created instances folder.

4. Though not necessary, I found setting up the instance as a Windows service to be very useful. I'll discuss the benefits of this approach in a bit. The following command will install our newly created cfusion2 instance as a Windows service. Between the quotation marks, the service display name is first, and the service description is second.

```
{install path}\JRun4\bin>jrunsvc -install cfusion2 "Macromedia JRun
CFusion2 Server" "Macromedia JRun CFusion2 Server"
```

5. Create your virtual hosts if you are using Apache, or if you are using IIS, create a few sites and test them. I use Apache, so I configured the virtual hosts and set them up to use "c:\appservers\apache group\apache2\htdocs2" instead of the regular htdocs directory.

6. Run the Web server configuration tool and map an instance to a site (IIS) or your Web server (Apache).

7. Start your ColdFusion instances and check that CFMX is running for the mapping you created. At this point all configured sites or virtual hosts are using the instance that JRun created during the installation, i.e., cfusion. However we want to use our second instance, which leads us to the next step.

8. At this point the IIS and Apache configurations differ:
   - ***IIS:*** Mapping instances to sites is a breeze thanks to the GUI tool, which lists all the sites configured on IIS. Simply select the instance (labeled as JRun server) you want to configure and the IIS site you want to map to.
   - ***Apache:*** Uses virtual hosts, which aren't accessible from the GUI tool. Instead you manually create the mappings to other instances and virtual hosts by editing the httpd.conf file (see Listing 1), saving it, and restarting Apache. (Remember to test the mappings.) You'll also need to create a server store for each instance in {install path}JRun 4/lib/wsconfig/. If you look in that folder you'll see that there is a folder called "1". Simply add another (i.e., "2") and restart your new instance service. This will create a file called jrunserver.store.

9. Now check your configuration. There are two suggested ways:
   - In his tutorial Purcell supplied the following snippet of code, which displays the instance name. Copy it into a .cfm template and execute it:

```
<cfobject action="create" type="java" class="jrunx.kernel.JRun"
name="jr" />
<cfset servername = jr.getServerName() />
<p><cfoutput>JRun Server Name: #servername#</cfoutput></p>
```

   - Another way to check is to log into the respective CFMX administrator instances and change the passwords for each instance and then log out. Then simply try the old and new passwords on each instance, and you'll see that only new ones allow you to log in.

10. If it all worked, you are done, but if not, read on, as we now come to the crux of this article – the stumbling blocks I encountered.

Stumbling Blocks

The JRE

If you've never played around with Java, your system may not be configured properly. By this I mean that the path to your Java Runtime Environment (JRE) may not be set up properly. I discovered this recently after installing JRun on a Windows 2003 Server, and I had a few difficulties launching the GUI version of the management console. I assume that you haven't got any Java SDKs installed but will be using the JRE provided with JRun.

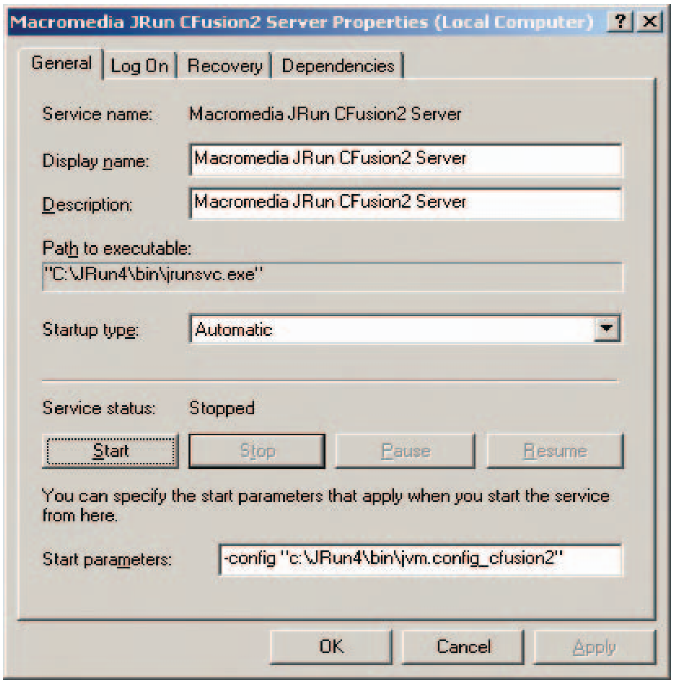
To troubleshoot this setup you'll need to check your JRE path. There are two ways to do this.

1. From the command line type "java –h". If the help information pops up, you are okay! If not, go to Step 2.

2. Check your path. At the command line, type "path" and hit enter. This will return all the path information configured on your system. Look for {install path} JRun4\runtime\jre\bin\. If you can find it, and Step 1 failed, check that it does not end with a semicolon (;). If you can't find it, go to your system properties (right-click on the My Computer icon and select Properties) and select the Advanced tab. Right at the bottom you'll see a button for Environment Variables. The dialog you see now has two windows. In the second (System Variables), you should be able to find a Path variable. Highlight and select the edit option, then go to the end of the entries in Variable value and add:

```
;{install path} JRun4\runtime\jre\bin\.
```

Click OK and open another command-line prompt (this is important) and try to run "java –h" again. You should now be set to use the command-line tools and Java GUIs for administering your JRun server.



Service name: Macromedia JRun CFusion2 Server

Display name: Macromedia JRun CFusion2 Server

Description: Macromedia JRun CFusion2 Server

Path to executable:
"C:\JRun4\bin\jrunsvc.exe"

Startup type: Automatic

Service status: Stopped

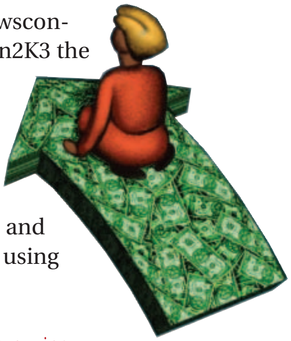
You can specify the start parameters that apply when you start the service from here.

Start parameters: -config "c:\JRun4\bin\jvm.config_cfusion2"

Figure 1: Using the Iterator to display the contents of ArrayList

The Web Server Configuration Tool

I encountered another obstacle when attempting to launch the Web server configuration tool. There are two ways to access this tool, the first located under JRun4\lib (use java -jar wsconfig.jar), and the second located under JRun4\bin (a compiled version, use wsconfig.exe). Under our installation of Win2K3 the precompiled version would not launch, so we had to resort to using the JAR file to create our instance mappings to a virtual host. Type the following command (all on one line) to create a mapping for Apache and the default cfusion instance without using the GUI tool:



```
C:\>{install path}JRun4\runtime\jre\bin\java -jar
{install path} JRun4\lib\wsconfig.jar -host 127.0.0.1 -server cfusion -
ws Apache -dir {install path}Apache2\conf -norestart -map
.cfm,.cfc,.cfml -coldfusion -v
```

The next problem involved the JRun proxy service. When you initially install the instance and configure it, the proxy service is set to initialized (go to the JRun JMC and select your instance/services/proxy to check it). When IIS issues the first request, the service automatically starts running. However, in my experience with using Apache, you need to give the proxy service a gentle nudge, else when you try to access the site, you'll get a "500 internal errors" message from Apache. So refer back to the services section and start it by clicking the Play icon under Actions from the JMC, and the previously encountered 500 errors should disappear. The Apache log file entries that were generated (and if you follow Purcell's instructions you'll have separate logs for each virtual host) provided the clues (see Listing 2 for some sample entries).

The proxy service is responsible for redirecting the requests to JRun and in turn the CFMX application instance; if the service is not up and running properly, the requests cannot be directed properly. Macromedia has posted a TechNote outlining the meaning of the error message and possible workarounds at www.macromedia.com/support/jrun/ts/documents/tn17333.htm.

I mentioned earlier the reliability of instances, and that if an instance fails it restarts pretty quickly – which is great most of the time. However, there may be times when you want to stop the instance for good. Stopping it from the JMC will not have the desired effect, as it will restart in no time. To stop the instance for good, you'll need to stop the Windows service of the instance you created previously.

An alternative to using the Windows service to stop/start an instance is to use the command line and use JRun to do so:

```
{install path} JRun4\bin\jrun -stop cfusion2
```

which will stop our second ColdFusion instance.

I also encountered a delay when stopping and starting instances. Even though the service appeared to be running, and the JMC console also seemed to indicate this, I found that it lags a little behind and the site does not respond immediately. That about covers the stumbling blocks I encountered. In the next section I will take a look at making configuration changes.

Configuration Changes

These changes can be done either by using the JMC or by editing the {install path}JRun/bin/jvm.config file. I found using the JMC to be unreliable, i.e., often the settings didn't work. On those occasions when using the JMC did work, any change made to the configuration was applied to all instances. We don't really want that, so we'll create a separate instance-specific jvm.config file. Before you do this, create a backup of your original jvm.config file.

Simply open the existing jvm.config file and save it as jvm.config_cfusion2 (I chose a name related to that of the instance I previously created, i.e., cfusion2). To demonstrate the effect on your instance, modify all the entries so that this new file will be applicable to only your Cfusion2 instance (see Listing 3, in which the changes to the original file are marked in red).

> **"Running multiple copies of CFMX on your server offers many advantages. For starters, you can optimize applications, and each instance can have its own CF Admin"**

Now you have your new jvm.config_cfusion2 file and all you need to do is tell the instance to start up using this configuration file. Since I installed my instances as services I want them to load this file at startup. There are two ways you can do this.

Option 1

If it's already installed as a service you'll need to uninstall it first. To remove the service, type:

```
{install path}JRun4\bin>jrunsvc -remove "Macromedia Jrun Cfusion2
Server"
```

(which was my Windows service display name). Please note that if you forget to stop the service first, its removal can take a while, as the service first needs to be stopped. Then reinstall it using the following additional switch:

```
-config jvm.config_cfusion2
```

So the command would now look as follows:

```
{install path]jrun4/bin/jrunsvc -install cfusion2 "Macromedia JRun
CFusion2 Server" "Macromedia JRun CFusion2 Server" -config
jvm.config_cfusion2
```

And now start the service and check your changes by logging on to the ColdFusion Administrator and accessing the system information. Under Java Class Path check that there is no reference to cfusion, but instead cfusion2, as this is the new instance name. To be sure, you can check the original instance and see that it is using only the Java Class Path of cfusion, which was our original instance. When making configuration changes I had the most success when stopping and restarting both the Apache the Server and the service instance.

Option 2

This is more elegant and involves adding a start parameter to your service from the service window (see Figure 1):

```
-config "{install path}JRun4\bin\jvm.config_cfusion2"
```

Restart the instance to allow the changes to take effect and then check your changes as you would have if using Option 1.

What if you don't want to install instances as a service but still want to make configuration changes? You can. I briefly mentioned the ability to stop an instance using the command:

```
{install path} JRun4\bin\jrun
-stop <instance name>
```

Similarly, you can start an instance in such a way and specify a config file as well. Try this command where "jvm.config_cfusion2" is the modified configuration file and cfusion2 the instance in question

```
{install path} JRun4\bin\jrun -start -config jvm.config_cfusion2
cfusion2
```

Now you should have a CFMX installation that reaps the benefits of the underlying J2EE architecture that JRun provides by adding portability, robustness, security, and optimization to your applications and service offerings.

Acknowledgments

- Brandon Purcell: Thanks for the articles and always answering my questions promptly.
- Simon Horwith: For the insights shared on the *CFDJ List* and reviewing this article.
- Sean Corfield: For his comments, posts, and blog.

About the Author

Greg Stewart is a senior developer with an insatiable appetite for all things Web- and software development–related. He has been developing applications using ColdFusion since 1996 and has been involved with projects in the UK, U.S., and Australia. You can visit him in his corner of the Web at www.tcias.co.uk.

gregs@tcias.co.uk

NORM MEYROWITZ
PRESIDENT OF PRODUCTS

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

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

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

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

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

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

Introducing
Macromedia MX 2004.

www.macromedia.com/go/2004

Listing 1

```
Listing 1
# JRun Settings (as configured by JRun)
LoadModule jrun_module "C:/AppServers/JRun4/lib/wsconfig/1/mod_jrun20.so"
<IfModule mod_jrun20.c>
    JRunConfig Verbose false
    JRunConfig Apialloc false
    JRunConfig Ssl false
    JRunConfig Ignoresuffixmap false
    JRunConfig Serverstore "C:/AppServers/JRun4/lib/wsconfig/1/jrunserv-
er.store"
    JRunConfig Bootstrap 127.0.0.1:51020
    #JRunConfig Errorurl <optionally redirect to this URL on errors>
    AddHandler jrun-handler .jsp .jws .cfm .cfml .cfc
</IfModule>

NameVirtualHost 127.0.0.1
########################################################
# JRun Instances Virtual hosts
########################################################
# Virtual host for Instance 1
<VirtualHost 127.0.0.1>
    ServerAdmin webmaster@site1.com
    DocumentRoot "C:/AppServers/Apache Group/Apache2/htdocs2/"
    ServerName www.site1.shed
    ErrorLog logs/site1-error_log
    JRunConfig Verbose false
    JRunConfig Apialloc false
    JRunConfig Ignoresuffixmap false
    JRunConfig Serverstore "C:/JRun4/lib/wsconfig/1/jrunserver.store"
    JRunConfig Bootstrap 127.0.0.1:51020
    #JRunConfig Errorurl <optionally redirect to this URL on errors>
    AddHandler jrun-handler .jsp .jws
</VirtualHost>
# Virtual host for Instance 2
<VirtualHost 127.0.0.1>
    ServerName www.site2.shed
    ServerAdmin webmaster@site2.com
    DocumentRoot "C:/AppServers/Apache Group/Apache2/htdocs3/"
    ErrorLog logs/site2-error_log
    JRunConfig Verbose false
    JRunConfig Apialloc false
    JRunConfig Ignoresuffixmap false
    JRunConfig Serverstore "C:/JRun4/lib/wsconfig/2/jrunserver.store"
    JRunConfig Bootstrap 127.0.0.1:51002
    #JRunConfig Errorurl <optionally redirect to this URL on errors>
    AddHandler jrun-handler .jsp .jws
</VirtualHost>
```

Listing 2

```
[Mon Jan 19 00:28:53 2004] [notice] jrApache[65285] Can't persist
servers, null server list.
[Mon Jan 19 00:28:53 2004] [notice] jrApache[65285] Couldn't initialize
from remote server, JRun server(s) probably down.
[Mon Jan 19 00:28:53 2004] [notice] jrApache[65285] JRun will not accept
request.  Check JRun web server configuration and JRun mappings on JRun
server.
```

Listing 3

```
#
# VM configuration
#
# Where to find JVM, if {java.home}/jre exists then that JVM is used
# if not then it must be the path to the JRE itself
java.home=C:/AppServers/JRun4/runtime/jre
#
# If no java.home is specified a VM is located by looking in these
places in this
# order:
#
#  1) bin directory for java.dll (windows) or lib/<ARCH>/libjava.so
(unix)
#  2) ../jre
#  3) registry (windows only)
#  4) JAVA_HOME env var plus jre (ie $JAVA_HOME/jre)
#

# Arguments to VM
java.args=-server -Xmx512m -Dsun.io.useCanonCaches=false -
Xbootclasspath/a:"{application.home}/servers/cfusion2/cfusion-ear/cfusion-
war/WEB-INF/cfusion/lib/webchartsJava2D.jar" -XX:MaxPermSize=128m -XX:+

UseParallelGC -DJINTEGRA_NATIVE_MODE -DJINTEGRA_PREFETCH_ENUMS

#
# commas will be converted to platform specific separator and the result
will be passed
# as -Djava.ext.dirs= to the VM
java.ext.dirs={jre.home}/lib/ext

#
# where to find shared libraries
java.library.path={application.home}/servers/cfusion2/cfusion-ear/cfusion-
war/WEB-INF/cfusion/lib,{application.home}/servers/cfusion2/cfusion-
ear/cfusion-war/WEB-
INF/cfusion/jintegra/bin,{application.home}/servers/cfusion2/cfusion-
ear/cfusion-war/WEB-
INF/cfusion/jintegra/bin/international,{application.home}/servers/cfu-
sion2/cfusion-ear/cfusion-war/WEB-INF/cfusion/lib/_nti40/bin
system.path.first=false

# JVM classpath
java.class.path={application.home}/servers/lib,{application.home}/servers/
cfusion2/cfusion-ear/cfusion-war/WEB-INF/cfusion/lib/cfusion.jar,{applica-
tion.home}/servers/cfusion2/cfusion-ear/cfusion-war/WEB-
INF/cfusion/lib,{application.home}/lib/jrun.jar,{application.home}/lib
```

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

—*continued from page 7*

should be anything but avoided in CFML applications.

Now that the word is out that Query of Queries performs very well, developers are embracing and experimenting with its functionality. Once the focus moved from performance concerns toward implementation, many posts began to appear on *CFDJ List* and other lists inquiring about Query of Queries behavior and syntax.

Longtime list member Tim Raster recently wrote the list regarding a problem that developers have apparently been running into quite a bit lately – the use of dates in a Query of Queries. It seems that more and more, developers are running into problems trying to use record set date columns in a Query of Queries. Tim's e-mail showed relatively straightforward syntax for record filtering based on a date, and he claims to have tried formatting the date with and without single quotes, with simple dateformat(), createODBCDate(), etc. – all to no avail. Every attempt results in a "Query of Queries runtime error" of one sort or another. Then Brad Comer inquired as to whether or not Tim had tried wrapping a <cfqueryparam> tag around any values that the date is compared with. Not only has Query of Queries been ignored by developers, but the <cfqueryparam> tag is another wildly underused tag as well, so it should come as no shock that such a simple solution would be so elusive. The need to use <cfqueryparam> is most likely due to ColdFusion not being able to differentiate between a string and a date data type column.

Another post, which Ray Camden quickly answered, was a question as to why a Query of Queries SELECT statement might always return no records even when the data and value in question are clearly there? Ray's answer – case sensitivity. Query of Queries is case sensitive, so a comparison – even a simple equality check, must cast both sides of its equation to all uppercase or lowercase using CFML functions and (to manipulate the column) with the UPPER() or LOWER() SQL function.

Still other posts frequent the list inquiring about INNER and OUTER JOINS in QofQ (only INNER JOINS are supported – and even then, only in the WHERE clause), aggregate functions (most of which are supported), subqueries (sorry, not supported), etc. Many of these questions can be answered simply by taking a quick peek at the documentation (see the QofQ livedocs at www.macromedia.com/support/documentation/en/coldfusion or download the documentation in PDF format from www.macromedia.com/support/documentation/en/coldfusion).

Query of Queries is just one of the functionalities that ColdFusion MX and MX 6.1 have breathed new life into. <CFMAIL>, <CFHTTP>, <CFSETTING>, and many other tags have interesting new features and functionality that are just waiting to be explored by developers. If you haven't already, I encourage you to take a few minutes here and there to skim the MX 6.1 documentation and to experiment with new and old functionality with MX 6.1; you won't be disappointed.

Expressing Yourself in ColdFusion

Using expressions to process data

By Jeffry Houser

Hello and welcome to the second installment of CF101. Last month I talked about variables in ColdFusion and how to create them using the <cfset> tag. I also showed you how to display the value of a variable using the <cfoutput> tag and a ColdFusion expression. I decided to leave an in-depth discussion of ColdFusion expressions to another time. That time is now.

Expressions Explained

You can use an expression to tell your CFML engine to process some data. A simple expression can be made up of a single element, called an operand. A complex expression is made up of multiple operands strung together with an operator. I'll discuss complex expressions a bit later.

The simplest expression is a single operand. There are many different types of operands that can be used in a ColdFusion expression. Here are some of the different elements:

- ***Literals:*** A literal value is a hard-coded value. Any of the simple data types, such as integers, real numbers, strings, dates, or Booleans, can be accepted as a literal value. Strings or dates are always enclosed in quotes (single or double). Quotes are not required for numeric literals. Boolean values are "Yes" and "No", "True" and "False", and any number and 0. Surrounding the value with quotes is optional with all but "Yes" and "No".
- ***Variables:*** Variables can be used as an expression. When a variable is used, the value of the variable is returned. A variable can contain a simple value such as a number or a string, or a complex value such as an array or structure.
- ***Functions:*** Any of ColdFusion's built-in functions can be used as a ColdFusion expression. With over 250 built-in functions, you can do a lot. I'll save the discussion of functions in CFML for another day. You can read more about ColdFusion's functions at http://livedocs.macromedia.com/coldfusion/6.1/htmldocs/function.htm.

Let's look at an example to see how expressions are used.

Putting an Expression to Use

You may remember from last month that I said the <cfset> is used in this format:

```
<cfset variableName = value>
```

However, this is actually a simplified version. The value portion of the tag is actually an expression. The following would be a more accurate description of the <cfset> tag:

```
<cfset variableName = expression>
```

Look at the following examples:

```
<cfset FirstName = "Jeffry">
<cfset LastName = "Houser">
<cfset Age = 28>
<cfset FName = FirstName>
```

Four variables are created with this segment of code. The first two lines create variables, FirstName and LastName, that contain string values. The value of each variable is a literal expression, with the value being a string. The third line creates a variable called Age using a numeric literal expression. The fourth line creates a variable called FName. Here our expression is a variable, not a literal value.

What can be done with the variables once they are created? The simplest example is to display them, using the <cfoutput> tag.

```
<cfoutput>
 #FirstName#<br>
 #LastName#<br>
 #Age#<br>
 #FName#<br>
</cfoutput>
```

The code starts with the <cfoutput> tag. Then comes an expression that displays the value in the FirstName variable. The expression is followed by an HTML line break. Then comes an expression that displays the LastName variable, and another line break. Third, an expression displays the age variable. Are you noticing a pattern yet?

The <cfoutput> tag block contains ColdFusion expressions and HTML. The pound signs allow the CFML server to tell the difference between expressions and plain text or HTML. The expressions are evaluated and the results are returned to the browser. The resulting browser display will look something like this:

```
Jeffry
Houser
28
Jeffry
```

Both code segments will have to be put in the same template to correctly display the output. If you've had any experience with ColdFusion before reading this article, you have probably already been exposed to expressions.

Pounding the CFML Pavement

As you examine the two code segments presented earlier, you'll probably notice that inside the <cfoutput> block we delimit the ColdFusion expressions using the pound sign, yet do not use any pound signs inside the <cfset> tags. Why is this? The pound sign is not needed inside as part of a ColdFusion tag such as <cfset>. You could write the <cfset> code segment with all the expressions in pound signs, like this:

```
<cfset #FirstName# = #"Jeffry"#>
<cfset #LastName# = #"Houser"#>
<cfset #Age# = 28>
<cfset #FName# = #FirstName#>
```

However, it is generally not considered good practice to do so. Specifying the pound signs means that your CF Server has to do more work, thus resulting in code that is less efficient than it could be. All those pound signs can also make the code more confusing to read. You don't usually need to use the pound sign if your expression is inside a ColdFusion tag or function. So when do you? A good rule of thumb is that if you're inside quotes you probably need pound signs, otherwise you probably don't (there are a few exceptions to this rule – like isDefined(), which takes a variable name as a string inside quotes).

So, if the pound signs aren't needed inside ColdFusion tags or functions, why are they needed in the <cfoutput> tag block? In <cfset> you're using the expression as part of the tag and it is safe to assume you are creating ColdFusion expressions. In the <cfoutput> block the expression is not part of the tag, only part of the text between the open and close tag. ColdFusion needs a way to distinguish between the two, and that is the reason for the pound signs. In most cases you can use pound signs only around simple expressions. To output the value of a complex expression, store the result to a variable and output the variable using a simple expression.

If you are creating a dynamic variable name using <cfset>, you'll need to use pound signs. Dynamic variable naming is beyond the scope of this article, but more information can be found at http://livedocs.macromedia.com/coldfusion/6.1/htmldocs/exprea24.htm. You may also need to use pound signs if you want to display the value of a variable inside a string literal (which is when you're inside quotes). More information on this can be found at http://livedocs.macromedia.com/coldfusion/6.1/htmldocs/exprea16.htm. You specify the attributes of many CFML tags as strings. When using them, I find it a good practice to ask myself whether I'm referencing a variable or a string. As mentioned earlier, if it's a variable, then use pound signs. Otherwise, specify the literal value without pound signs.

Using Operators to Create Complex Expressions

All of the expressions we've looked at up to this point have been simple expressions. Simple expressions can be tied together with one or more operators to form complex expressions. The format of an expression would be something like this:

```
Expression Operator Expression
```

It is worth noting that each expression can contain another expression, thus creating very complex expressions.

Operators fall into four main types: arithmetic, strings, decision, and Boolean. Arithmetic operators are used to perform math functions; string operators are used to combine multiple strings into a single string; decision operators are used to produce Boolean values based on the results of a comparison; and Boolean operators are used to perform logical operations. I'll save a discussion of Boolean and decision operators for the next column. Here are some of the more commonly used string and arithmetic operators:

- ***+:*** The plus operator is an arithmetic operator used to perform addition of two numbers. The plus operator can also be used as a unary operator to define a number as a positive number.
- ***-:*** The minus operator is an arithmetic operator used to perform subtraction of two numbers. The minus operator can also be used as a unary operator to define a number as a negative number.
- ****:*** The multiplication operator is an arithmetic operator used to multiply two numbers together.
- ***/:*** The division operator is an arithmetic operator used to divide a number by a second number.
- ***MOD:*** The mod command is an arithmetic operator that will divide a number by a second number and return the integer remainder.
- ***\\:*** The \ is an arithmetic div operator. It divides a number by a second number and returns an integer result without a remainder.
- ***^:*** The caret symbol is an arithmetic operator used to define the exponents of a number (to raise a number to a power).
- ***&:*** The ampersand operator is the only string operator. It is used to concatenate two strings together. The second string is placed at the end of the first string and the result is returned as a single string.

With all these operators, you will need to know the order of operations. Thankfully, the order of operations that ColdFusion uses is very similar to the one you probably learned in an elementary math class. First, the unary + and unary – operators are applied to specify which numbers are positive and which are negative. After the unary operators, the exponent symbol is evaluated. Next multiplication and division are performed, from left to right. Then the div operator is executed. Next comes the MOD operator. Following that, the subtraction and addition operators are applied from left to right. The string concatenation operator comes next, followed by all the decision and Boolean operators that are beyond the scope of this article. Normally you wouldn't want to mix string and arithmetic operators, but it is possible. Just as in math class, you can control the order of operations using parentheses, and evaluation always begins with the innermost set when parentheses are nested.

Examples of Complex Expressions

Here is an example of some complex expressions in action:

```
<cfset FirstName = "Jeffry">
<cfset LastName = "Houser">
<cfset FName = FirstName & LastName>
<cfset Age = 27+1>
```

The code starts out simply by setting two variables, FirstName and LastName. Both of these lines use simple expressions. The next line is more interesting. It creates a variable called FName by concatenating the FirstName variable with the LastName variable. The final piece of code uses the addition operator to add two numbers together. We can display these values using this code:

```
<cfoutput>
 #FName#<br>
 #Age#<br>
</cfoutput>
```

The display code should be familiar to you based on our other examples. We have a <cfoutput> block and a mix of CFML expressions and HTML line breaks. If you put both code segments into a browser and execute the code, you should see something like this:

```
JeffryHouser
28
```

But wait. Maybe you don't want to butt the FirstName and LastName up against each other. We can use a string literal and the concatenation operator to modify the <cfset> tag to add a space between the two in the FName value. Replace the third line of cfset code with this:

```
<cfset FName = FirstName & " " & LastName>
```

Then when you execute the display code you will see the full name correctly separated.

Conclusion

Expressions are an important part of ColdFusion development. Next month we'll take a look at making decisions in your code using the <cfif> tag. That article will also include an examination of Boolean and decision operators.

About the Author

Jeffry 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@farcryfly.com

LOOK FOR YOUR FREE...

SYS-CON MEDIA

›Linux ›Java ›Web Services ›.NET ›XML ›Wireless ›Storage ›Security

The Premier Resource for Today's Corporate & IT Decision Makers

VOL 1 ISSUE 1 SPRING 2004

IT solutions ›GUIDE

WWW.SYS-CON.COM/IT

» Delivering **Software as Service**

» Leveraging **Linux/Open Source**

» Moving to a **Service-Oriented Architecture**

» Desktop Software: **Migrating from Server to Client**

» Using **Developer Tools to Drive Cost Out of Software**

» Application **Integration**

» Storage & **Security**

TECHNOLOGIES YOU NEED NOW!

How to Manage Your Ideas Using Today's *i*-Technologies

Reaching 135,000 Corporate and IT Decision Makers

$5.99US $7.99CAN

Coming this SPRING!

Adventures in Encapsulation PART 2

Talking OO: Learning to think in ‘objects’

By Hal Helms

It’s past midnight and I’m in my car, prowling Atlanta for a good cup of coffee. (Such are the challenges of members of the Cult of the Midnight Programmer.) My cell phone rings.

HAL: Hello?
BEN: HB? This is Ben.
HAL: Ben! What’s up?
BEN: I’ve been thinking about a more effective way to use polymorphism in Mach-II. Got some time? I want to bounce some ideas off you.
HAL: Sure, Ben. I was think—

My phone shows a call waiting. Surely, normal people don’t have to interrupt one call in the middle of the night for another?

HAL: Hello?
JOHN: HB? This is John.
HAL: John, what’s up?
JOHN: I’m working on a project and I need to talk OO with you.
HAL: John, I’ve got Ben Edwards on the other line. Let me give you a call back in a bit.

An hour later Ben and I hang up and I call John. I still haven’t found any place open that serves good coffee.

HAL: Okay, John. Fire at will. So tell me about your OO project.
JOHN: Okay. I’m working on a tool to help clients determine their requirements for a Web application. I have the concept of a storyboard. Each storyboard may have multiple pages, and each of these pages may have multiple elements attached to it.
HAL: Elements?
JOHN: That’s just what I’m calling them. An element describes some discrete part of the page. You might have a form element and another element that deals with articles, then another element that represents an image. You see where I’m going with this?
HAL: Yes.
JOHN: So, I want to make this an OO application and use CFCs. It would give me a chance to try “thinking in objects.”
HAL: I like it.
JOHN: So I began by asking what sort of “things” make up my little universe and whether those things would make good CFCs.
HAL: What did you come up with?
JOHN: I have the following CFCs: Storyboard, Page, Element.
HAL: Hmmm… Just one Element CFC?
JOHN: Yes, but it has a field to say what kind of element it represents. I was going to call it elementType.
HAL: Go ahead.
JOHN: Well, we have a classic one-to-many join between Storyboard and Page and between Page and Element (see Figure 1).
HAL: Actually, we have a classic relationship between those, not a join.
JOHN: Why not a join?
HAL: Because a join is a very specific thing: it describes how tables in a database are related.
JOHN: It’s the same thing, isn’t it?
HAL: Not at all. The relationship exists in reality, whether we model it into a relational database schema or not.
JOHN: I think we’re just quibbling over semantics.
HAL: Okay. Well, let’s continue. So you’ve identified these various CFCs. Where did you go from there?
JOHN: I started with Storyboard, and I figure I need a getPages method.
HAL: What sort of thing will getPages return?
JOHN: Well, I was reading your article in the last ***CFDJ*** about ArrayLists and I thought an ArrayList of Page objects would be good.
HAL: That sounds good to me.
JOHN: Only the order the pages are returned to me in is important. They need to be alphabetized by the pageTitle column.
HAL: The pageTitle column?
JOHN: Yes, that’s a column in my Page table. So I figured that getPages should just go to the database, run a query, and then return the pageIDs of the pages that belong to that storyboard. That way, I can do an ORDER BY on my SQL and get the pageIDs in the right order.
HAL: But now you have a list of pageIDs, not Page objects.
JOHN: Yeah… but I guess when I need info on a specific page, I can drill down for that page then. Sure, that’ll work fine.
HAL: Let me ask you about adding a page to a storyboard. What happens?
JOHN: I save the information the client needs to the database.
HAL: And then getPages just queries the database.
JOHN: Right.
HAL: Did you find the design at all challenging? I mean, this is your first OO project.
JOHN: Not at all. It just seemed so natural to me.
HAL: I can believe that.

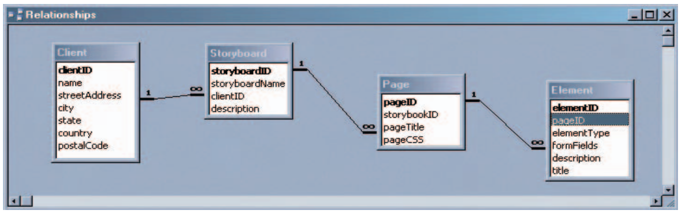


Figure 1: John’s database schema

JOHN: Wait a minute. That tone of voice… you don't like it!

HAL: John, exactly how is this "OO" project different from any other project you've ever done?

JOHN: Because we identify CFCs first?

HAL: And having identified them, they don't do much but hold code to go into and out of the database. That's not exactly an earth-shaking change. Your objects are definitely second- or third-class citizens. The real power in the land is the database.

JOHN: And that's not good.

HAL: I won't say that, but I will say that it's not object oriented. OO is all about encapsulation. But you say your Storyboard CFC has SQL to get the pages.

JOHN: Yes, the pageIDs.

HAL: And this Storyboard CFC is well encapsulated?

JOHN: Sure it is. It knows its pages; it knows its owner; it knows how to store and retrieve that info.

HAL: And, like many well-encapsulated components, it should be reusable?

JOHN: Yes. At least some of them are. Client, for example. And Element, because we may have the same element on several different pages. So really, I should model that as a many-to-many join…

HAL: So how can you – writing this code at one point in time – know what kind of database I'll be using at a later date when I reuse your code? Or am I forced to use one that you specify so that it conforms with your persistence model? Does that seem encapsulated or abstracted to you?

JOHN: No, I see your point, but—

HAL: But what?

JOHN: I guess I could have methods like getPagesFromOracle and getPagesFromSqlServer but that's too ugly to be right.

HAL: I agree.

JOHN: So what is the solution?

HAL: The solution is to stop thinking about databases at all. Purge your mind of the relational calculus. Enter into ObjectLand. Now, let's start writing the CFC for Storyboard (see Listing 1).

JOHN: That looks pretty good. I think you made a mistake, though, on the getClient and setClient methods. You say the type is Client, but isn't it really a string called clientID?

HAL: What is clientID?

JOHN: It's a column in the datab—

HAL: Exactly. Purge your mind of all databasey thoughts, Grasshopper. If I'm a Storyboard and you want to know who owns me, will it really be helpful if I tell you "211154"? Nope, you want me to pull my owner out of my pocket when you say getClient.

JOHN: Yes, but Storyboard though you may be, you don't have any pockets.

HAL: Sure I do. I call them instance variables and they're saved in the variables scope.

JOHN: So if I want to know the client, I don't ask you, the Storyboard, for a primary key, but for the—?

HAL: For the object itself.

JOHN: And what does this object look like?

HAL: Well, it's an instance of the Client CFC, right? So, here's the code for Client.cfc (see Listing 2).

JOHN: And where is the information about the client's street address, city, state, zip – that sort of thing?

HAL: Why, in Address.cfc, where it belongs (see Listing 3).

JOHN: So you have a separate object for an address?

HAL: Sure, why not? Now that I have this handy Address CFC, I can use it for all sorts of CFCs that have addresses: suppliers, venues, employees, etc.

JOHN: How do I associate a Client object with an Address object?

HAL: It's simple: look at the init method of Client. It accepts—

JOHN: name, which is a string, and address, which is an actual Address object. Is there something special about the init method?

HAL: No. Because CFCs lack the idea of a constructor – think of it as a special method that automatically runs whenever an object is created – many developers have decided to place in an init method whatever would be in a true constructor.

JOHN: But it isn't called automatically?

HAL: No, you have to remember to call it. You asked how you would associate an Address object with a Client object. First, let me ask you how you're planning to get the information on a new client.

JOHN: They'll fill out a form that asks for their info.

HAL: Great. Then for simplicity's sake, let's say that you just want their name and their address info. Now, let's look at what the form-processing page might have in it:

```
<cfset address = CreateObject('component', 'Address').init(form.street,
form.city, form.state, form.zip) />
```

HAL: That returns an Address object. Now for the Client object:

```
<cfset client = CreateObject('component', 'Client').init(form.name,
address) />
```

HAL: That's it. Now, let's say that this particular client has logged onto your system. Put the client object into a persistent scope – session might be good here – and you're good to go.

JOHN: Okay, but my situation is a little more complicated. I have storyboards, pages, clients, and elements.

HAL: You might represent it like this (see Figure 2).

HAL: Each box consists of three sections. The top section is the name of the CFC. The middle lists the instance variables – the data belonging to the object – including the data type for

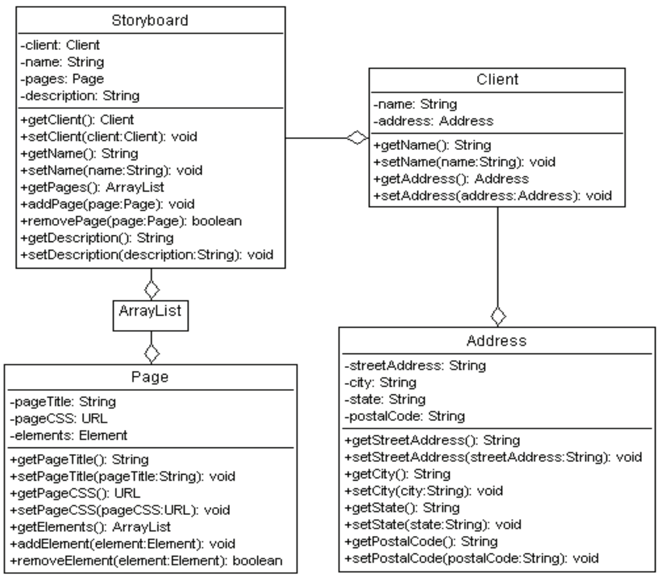

Storyboard
-client: Client
-name: String
-pages: Page
-description: String
+getClient(): Client
+setClient(client:Client): void
+getName(): String
+setName(name:String): void
+getPages(): ArrayList
+addPage(page:Page): void
+removePage(page:Page): boolean
+getDescription(): String
+setDescription(description:String): void
Client
-name: String
-address: Address
+getName(): String
+setName(name:String): void
+getAddress(): Address
+setAddress(address:Address): void
ArrayList
Page
-pageTitle: String
-pageCSS: URL
-elements: Element
+getPageTitle(): String
+setPageTitle(pageTitle:String): void
+getPageCSS(): URL
+setPageCSS(pageCSS:URL): void
+getElements(): ArrayList
+addElement(element:Element): void
+removeElement(element:Element): boolean
Address
-streetAddress: String
-city: String
-state: String
-postalCode: String
+getStreetAddress(): String
+setStreetAddress(streetAddress:String): void
+getCity(): String
+setCity(city:String): void
+getState(): String
+setState(state:String): void
+getPostalCode(): String
+setPostalCode(postalCode:String): void

Figure 2: An object-based diagram of relationships

each variable. The bottom shows the methods for each CFC.

JOHN: That one is a little more complicated.

HAL: Let's look at one. The Client CFC has a good one:

```
setAddress(address: Address): void.
```

HAL: You read it like this: the setAddress method accepts a single argument, called address of data type, Address. The method returns nothing.

JOHN: Is that UML?

HAL: That's a particular diagram type in UML called a class diagram. You use this diagram to build up an object model. More important, though, do you see how we've shown the relationship that exists within CFCs without even thinking about how this will be stored in a database?

JOHN: Yes.

HAL: You might have a session object – call it myStoryboard. Once the object is built, you just ask the object for the info you need. Want to display the number of pages in the storyboard?

JOHN: And we're not going to the database?

HAL: Why would we need to? We have all the information right here. All I need is a way to find out how many elements are in the pages ArrayList. Luckily, I have a length method in ArrayList.

```
There are currently #session.myStoryboard.getPages().length()# pages in
your storyboard.
```

HAL: The point is that databases are meant to persist data – to make sure it isn't lost when you turn your machine off. But as a way of providing structure to an application, they're not very good.

JOHN: But does your database have to match up with your object model? Like, would I have to move the client's address information, which right now is part of the Client table, to a separate Address table?

HAL: No. You might choose to do that, but is it necessary? Not at all. Otherwise, we'd be stuck going forward since we're often writing new apps that need data from older databases.

JOHN: Okay, but remember that I need to have my ArrayList of…Page Objects—

HAL: Right.

JOHN: I need those returned in a sorted order.

HAL: How about a handy-dandy ArrayListSorter CFC?

JOHN: They have one of those?

HAL: "They" don't, but maybe you can create one. It would be nice to have such a thing whenever you need to sort an ArrayList, wouldn't it?

JOHN: Yes. I guess it would need… what? A sort method?

HAL: Sure. What arguments does it accept?

JOHN: I'll need an ArrayList, of course. Then… hmmm… some way of indicating what we want to sort on.

HAL: Of course an ArrayList can contain all sorts of data types – queries, arrays, structs, simple values, and – what we're concerned with – objects. Now, we don't have time to discuss a comprehensive solution to sorting, but we can put together a narrower solution that will work for an ArrayList with either simple values or objects in it.

JOHN: So my ArrayListSorter will need to work if the ArrayList has either simple values or objects. But what if my ArrayList has both simple values and objects?

HAL: But how would that make sense? If you have an ArrayList that has – let's say, BaseballPlayers and numbers of fans in attendance – what would that ArrayList represent?

JOHN: I don't know.

HAL: Me either. So it sure wouldn't make sense to try to sort such a collection of things.

JOHN: Right. Okay, so shall I take a crack at an ArrayListSorter?

HAL: Sure. Just remember that you'll need to accept an ArrayList and something to sort by.

JOHN: But in some cases – like where the ArrayList has simple values – I won't need a property to sort by.

HAL: So, your CFC will need to accommodate both situations. If you want, I can help you write the code. It doesn't look like I'm going to find—

Suddenly, I spot it: "Mac's 24-Hour Really Good Coffee" on – where else? – Peachtree Street.

HAL: You know, on second thought, John, it would probably be a good practice for you to try it yourself.

JOHN: Okay. I'll e-mail it to you when I finish.

Mac was as good as his word. And when – after 3 a.m. – I arrived home, John's e-mail was waiting for me:

From: john
To: hal
Subject: ArrayListSorter
Hal,
Here's my code. I don't know if it's the most efficient sort mechanism, but it works. But your UML drawing? You forgot to add in an Element CFC. Anyway, I'm starting to get this OO stuff and I like it a lot! Thanks for all the help, John

John's code seemed to work fine (see Listing 4). But I hadn't, in fact, forgotten the Element CFC. But that discussion will involve polymorphism and will have to wait until next time.

About the Author

Hal Helms (www.halhelms.com) is a Team Macromedia member who provides both on-site and remote training in ColdFusion, Java, and Fusebox. Hal is cofounder of the Mach-II project.

hal.helms@teamallaire.com

Listing 1: Storyboard CFC

```
<cfcomponent displayname="Storyboard">
   <cfset variables.client = "" />
   <cfset variables.pages = "" />

   <cffunction name="init" access="public" returntype="Storyboard" out
   put="false">
      <cfargument name="client" type="Client" required="true" />
      <cfset setClient(arguments.client) />
```

Now More Than Ever Before...

JAVA DEVELOPER'S JOURNAL

JDJ

Means Business

EXCLUSIVE: WHERE IS TECH GOING IN 2004?
JAVA DEVELOPER'S JOURNAL
JDJ
JANUARY 2004
VOLUME:9 ISSUE:1
SPECIAL REPORT: SCOTT McNEALY 2004 PREDICTIONS
PLUS...
Viewpoint: Deutsch's Fallacies, 10 Years After
Feature: Multiplatform Frames
Data Integration: Which EII Solution Is Right for You?
EDGE 2004 EAST International Conference & Expo BOSTON, MA FEBRUARY 24-26, 2004
TAKE A RIDE WITH THE CEO OF BEA
RETAILERS PLEASE DISPLAY UNTIL MARCH 31, 2004
$5.99US $6.99CAN
THE WORLD'S LEADING i-TECHNOLOGY MAGAZINE WWW.JAVADEVELOPERSJOURNAL.COM

Subscribe Now!

$69.99 1 YEAR/ 12 ISSUES

www.SYS-CON.com/JDJ

or call

1-888-303-5282

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

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

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

SYS-CON MEDIA

The World's Leading i-Technology Publisher

```
      <cfset setPages(CreateObject('component', 'ArrayList')) />
      <cfreturn this />
   </cffunction>

   <!--- Accessors --->
   <cffunction name="getClient" access="public" returntype="Client" out
   put="false">
      <cfreturn variables.client />
   </cffunction>
   <cffunction name="setClient" access="public" returntype="void" out
   put="false">
      <cfargument name="client" type="Client" required="true" />
      <cfset variables.client = arguments.client />
   </cffunction>

   <cffunction name="getPages" access="public" returntype="ArrayList"
   output="false">
      <cfreturn variables.pages />
   </cffunction>
   <cffunction name="setPages" access="public" returntype="void" out
   put="false">
      <cfargument name="pages" type="ArrayList" required="true" />
      <cfset variables.pages = arguments.pages />
   </cffunction>
</cfcomponent>
```

Listing 2: Client.cfc

```
<cfcomponent displayname="Client">
   <cfset variables.name = "" />
   <cfset variables.address = "" />

   <cffunction name="init" access="public" returntype="Client"
   output="false">
      <cfargument name="name" type="string" required="true" />
      <cfargument name="address" type="Address" required="true" />
      <cfset setName(arguments.name) />
      <cfset setAddress(arguments.address) />
      <cfreturn this />
   </cffunction>

   <!--- Accessors --->
   <cffunction name="getName" access="public" returntype="string" out
   put="false">
      <cfreturn variables.Name />
   </cffunction>
   <cffunction name="setName" access="public" returntype="void"
   output="false">
      <cfargument name="name" type="string" required="true" />
      <cfset variables.name = arguments.name />
   </cffunction>

   <cffunction name="getAddress" access="public" returntype="Address"
   output="false">
      <cfreturn variables.address />
   </cffunction>
   <cffunction name="setAddress" access="public" returntype="void" out
   put="false">
      <cfargument name="address" type="string" required="true" />
      <cfset variables.address = arguments.address />
   </cffunction>
</cfcomponent>
```

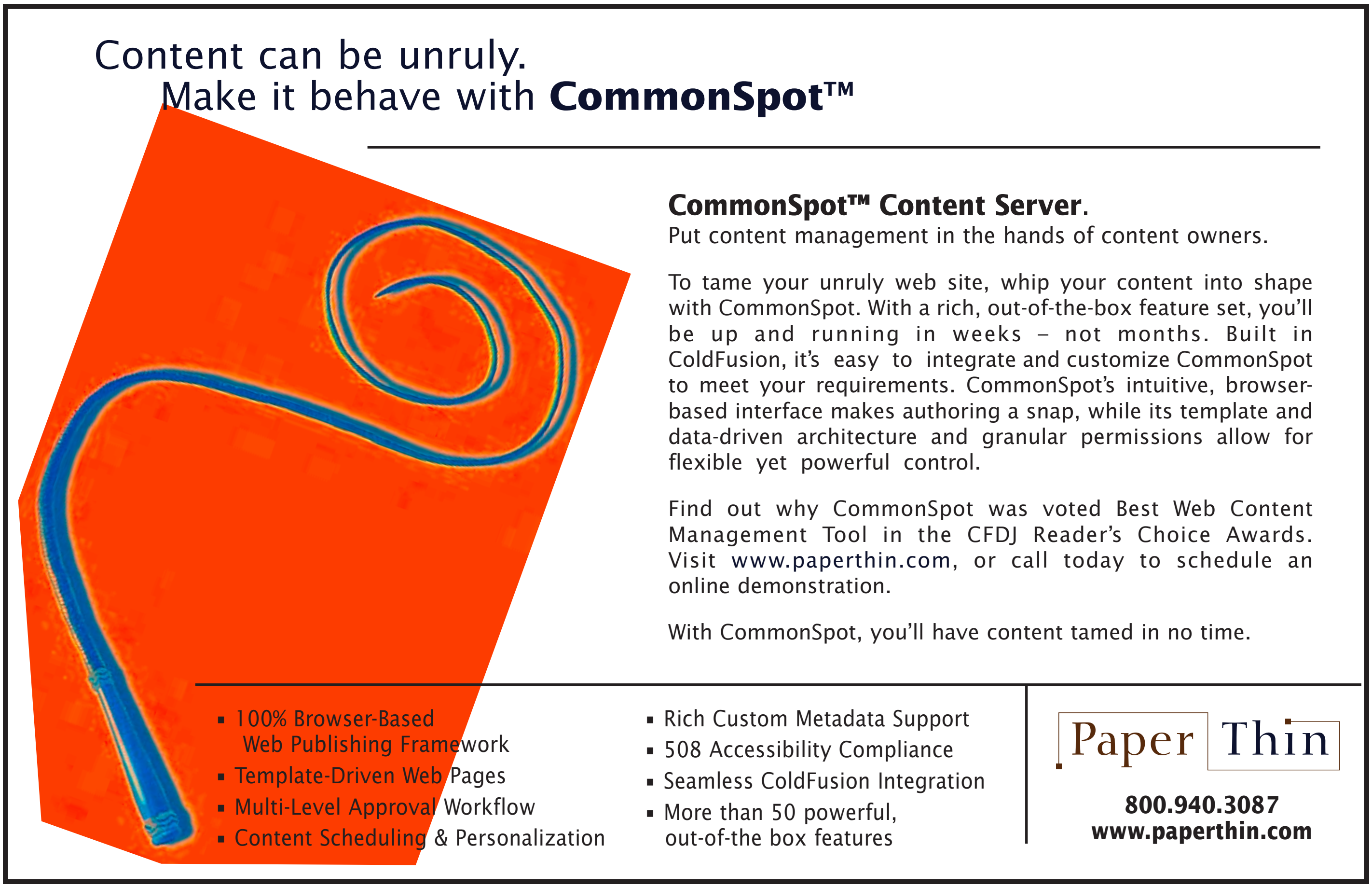
Listing 3: Address.cfc

```
<cfcomponent displayname="Address">
   <cfset variables.streetAddress = "" />
   <cfset variables.city = "" />
   <cfset variables.state = "" />
   <cfset variables.postalCode = "" />

   <cffunction name="init" access="public" returntype="Address"
   output="false">
      <cfargument name="streetAddress" type="string" required="true" />
      <cfargument name="city" type="string" required="true" />
      <cfargument name="state" type="string" required="true" />
      <cfargument name="postalCode" type="string" required="true" />

      <cfset setObjectID(CreateUUID()) />
      <cfset setStreetAddress(arguments.streetAddress) />
      <cfset setCity(arguments.city) />
      <cfset setState(arguments.state) />
      <cfset setPostalCode(arguments.postalCode) />
      <cfreturn this />
   </cffunction>

   <cffunction name="getObjectID" access="public" returntype="string"
   output="false">
      <cfreturn variables.objectID />
   </cffunction>
   <cffunction name="setObjectID" access="private" returntype="void" out
   put="false">
      <cfargument name="objectID" type="string" required="true" />
      <cfset variables.objectID = arguments.objectID />
   </cffunction>

   <cffunction name="getStreetAddress" access="public"
   returntype="string" output="false">
      <cfreturn variables.streetAddress />
   </cffunction>
   <cffunction name="setStreetAddress" access="public" returntype="void"
   output="false">
      <cfargument name="streetAddress" type="string" required="true" />
      <cfset variables.streetAddress = arguments.streetAddress />
   </cffunction>

   <cffunction name="getCity" access="public" returntype="string" out
   put="false">
      <cfreturn variables.city />
   </cffunction>
   <cffunction name="setCity" access="public" returntype="void"
   output="false">
      <cfargument name="city" type="string" required="true" />
      <cfset variables.city = arguments.city />
   </cffunction>

   <cffunction name="getState" access="public" returntype="string" out
   put="false">
      <cfreturn variables.state />
   </cffunction>
   <cffunction name="setState" access="public" returntype="void" out
   put="false">
      <cfargument name="state" type="string" required="true" />
      <cfset variables.state = arguments.state />
   </cffunction>

   <cffunction name="getPostalCode" access="public" returntype="string"
   output="false">
      <cfreturn variables.postalCode />
   </cffunction>
   <cffunction name="setPostalCode" access="public" returntype="void"
   output="false">
      <cfargument name="postalCode" type="string" required="true" />
      <cfset variables.postalCode = arguments.postalCode />
   </cffunction>
</cfcomponent>
```

Listing 4: ArrayListSorter.cfc

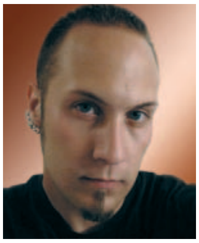
```
<cfcomponent displayname="ArrayListSorter">

   <cffunction name="init" access="public" returntype="ArrayListSorter"
         output="false">
      <cfreturn this />
   </cffunction>

   <cffunction name="sort" access="public" returntype="ArrayList" out
   put="true">
      <cfargument name="arrayList" type="ArrayList" required="true" />
      <cfargument name="sortBy" type="string" required="false"
   default="" />
      <cfset var tmp = ArrayNew(1) />
      <cfset var it = arguments.arrayList.getIterator() />
      <cfset var sortedArrayList = CreateObject('component',
   'ArrayList').init() />
      <cfloop condition="#it.hasNext()#">
         <cfset next = it.next() />
         <cfif Len(arguments.sortBy)>
            <cfset setVariable('next', Evaluate('next.get' & argu
            ments.sortBy & '()')) />
         </cfif>
         <cfset ArrayAppend(tmp, next) />
      </cfloop>

      <cfset ArraySort(tmp, 'text') />

      <cfloop from="1" to="#ArrayLen(tmp)#" index="i">
      <cfset it = arguments.arrayList.getIterator() />
         <cfloop condition="#it.hasNext()#">
            <cfset next = it.next() />
            <cfset nextValue = next />
            <cfif Len(arguments.sortBy)>
               <cfset setVariable('nextValue', Evaluate('next.get' &
                arguments.sortBy & '()')) />
            </cfif>
            <cfif nextValue EQ tmp[i]>
               <cfset sortedArrayList.add(next) />
               <cfbreak>
            </cfif>
         </cfloop>
      </cfloop>

      <cfreturn sortedArrayList />
   </cffunction>

</cfcomponent>
```

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



Confronting JRun – Who's Afraid of J2EE?

Open the doors to some powerful possibilities



By Benjamin Blackwell

Let's start with a few basic truths. First, the average ColdFusion developer is not a Java developer. Second, most ColdFusion applications do not utilize Java. In all the hype and excitement surrounding CFMX, many developers and their applications may have been left behind.

The New Paradigm

ColdFusion MX provides many benefits for developers but also brings with it many questions, the answers to which are not readily apparent. Unfortunately, in some cases CFMX 6.1 lacks documentation regarding these questions, and, for the non-Java developer, reading JRun documentation can be like reading ancient Greek. Articles highlighting the basics appear only infrequently on Macromedia.com and the ColdFusion blogs. Unless you were already familiar with the world of JRun and J2EE, you may have quickly become lost in this new paradigm.

The ColdFusion MX Standard edition and the stand-alone installation of CFMX Enterprise are similar to the ColdFusion Server product developers were once familiar with. JRun is at the core of these products but is essentially hidden from the developer. Both the Standard edition and the stand-alone installation of Enterprise benefit from what the J2EE architecture has done for the ColdFusion language, but neither allows the developer to take full advantage of other features that J2EE offers.

CFMX Enterprise deployed on JRun allows developers to build applications that utilize the strengths of the J2EE platform. You can deploy multiple instances of the same application or deploy many separate applications for heightened stability and security. You can employ Java servlets, JavaServer Pages, and Enterprise JavaBeans in your applications. You can also cluster your applications for failover and high availability.

In order to use any of these features, it helps to have an understanding of what's going on under the hood. To answer some of the questions, and to bring a better understanding of JRun to the average ColdFusion developer, I will be focusing on a full J2EE installation of CFMX 6.1 Enterprise Edition for Windows with the included JRun 4 Application Server. We will also look at a similar installation on Red Hat Linux 7.2.

The JRun Launcher

The JRun Launcher is a GUI tool used for starting and stopping JRun server instances under Windows. In a production environment, you will most likely use the JRun Management Console (JMC) in place of the JRun Launcher for controlling server instances. However, the JRun Launcher is a convenient tool for starting server instances on a development machine or for starting the Admin server instance, which controls the JMC.

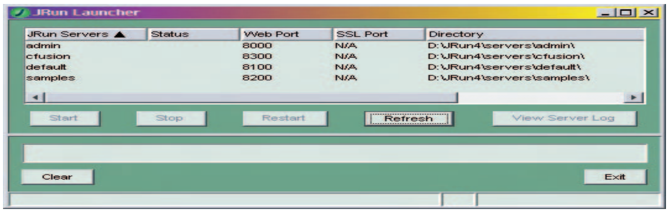
To open the JRun Launcher on your Windows machine, go to Start > Programs > Macromedia > Macromedia JRun 4 > JRun Launcher. You should then see a command window and after a few seconds, the JRun Launcher application (see Figure 1). If you close the command window, it will also close the Launcher and you will need to restart it.

The Launcher window lists the available servers that you may control. It also defines the servers' status, port number, and directory where each server exists.

In order to access the JRun Management Console, the Admin server must be running. If it is not, select it in the Launcher window and click the Start button (it may already be running if JRun was installed as a Service under Windows). Be patient; it may take several seconds for the server to start. Once the Admin server has started, you may click Exit in the JRun Launcher. Both the Launcher and the system message window should close.

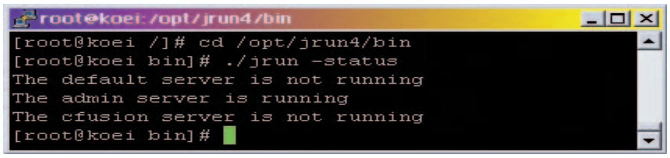
Launching JRun via the Command Line

Instead of using a GUI tool like the JRun Launcher, you can view the server status and control JRun from a command line in both Windows and Linux. Open a command shell and navigate to your JRun installation's /bin directory, /opt/jrun4/bin in our case on Linux, and D:/jrun4/bin on Windows. Under Windows the command



JRun Servers ▲	Status	Web Port	SSL Port	Directory
admin		8000	N/A	D:\JRun4\servers\admin\
cfusion		8300	N/A	D:\JRun4\servers\cfusion\
default		8100	N/A	D:\JRun4\servers\default\
samples		8200	N/A	D:\JRun4\servers\samples\

Figure 1: JRun Launcher window



```
[root@koei /]# cd /opt/jrun4/bin
[root@koei bin]# ./jrun -status
The default server is not running
The admin server is running
The cfusion server is not running
[root@koei bin]#
```

Figure 2: JRun server status on Linux

```
jrun -status
```

will display the available servers information. The command

```
./jrun -status
```

on Linux will do the same (see Figure 2).

Unlike the JRun Launcher, this gives you only a list of all server instances and whether they are running or stopped, but what more do you really need? To start the Admin server instance (if it is not already running) type

```
./jrun -start -nohup admin
```

for Linux or

```
jrun -start -nohup admin
```

for Windows.

The JRun Management Console

After the Admin server has started, open a Web browser to the address http://localhost:8000 (you may substitute the IP address or URL for localhost if you are working on a remote server). If the Admin server is running, you will be presented with the JMC login screen. Enter the username and password you defined during the installation process to log in. Under Windows, you may also go to Start > Programs > Macromedia > Macromedia JRun 4 > JRun Management Console to access the JMC.

Log in to the JMC and you will be presented with a list of available servers. These will be the same servers listed in the JRun Launcher window (or listed from the –status command). By default, the installer configures four different servers (see Figure 3).

This view lists basic information for each server instance and basic controls for each. Under the Actions column you will see four icons, each of which has a pop-up tool tip to tell you its purpose. The icons are Edit, Start/Stop, Restart, and Delete. The Host column defines the name of the physical server on which the instance is running. The JNDI Port column can be safely ignored for your ColdFusion installation. The HTTP port is the default port for accessing each server through a Web browser. The Proxy Port column is the port number on which JRun communicates with an external Web server. And finally, the Status column defines whether the server is currently running or stopped.

So what are these servers that are created by default when you install ColdFusion deployed on JRun, and what do they do?

- ***Admin:*** Contains the JMC application and must be running in order to access it. This server is necessary only when you need to make changes to JRun through the JMC, such as adding or removing a server instance. In a production environment it is safe to start this server only when needed.
- ***Cfusion:*** Supports the default CFMX application environment. This is the server that controls your ColdFusion applications; it is also the server with which you will have the most interaction. Of the four servers, this is the only one necessary for deploying and accessing your ColdFusion applications (it must be running to do so).
- ***Default:*** The default server for deployment of Java Web archives (WARs) and Enterprise JavaBeans (EJBs) under a normal JRun installation. This server can be safely disabled or removed in a production environment if it is not being used to deploy Java applications.
- ***Samples:*** This one should be self-explanatory. This is a server configured to run the sample applications that ship with JRun and should be removed or disabled in a production environment.

In a typical ColdFusion installation, you will rarely, if ever, need to interact with any server other than cfusion, though you may have ColdFusion deployed on more than one JRun server instance on a machine. Any ColdFusion-related configuration and administration tasks are handled in the ColdFusion Administrator, which runs under the cfusion server instance in a default installation. In more advanced ColdFusion environments, however, the JMC allows you many options for controlling and customizing your server environment.

Controlling Servers

The Available Servers panel in the JMC allows you to control all server instances under JRun. It gives you the ability to start and stop or restart any configured server, just as the JRun Launcher does. But it also gives you the ability to edit or delete any server instance.

Start the cfusion server by clicking its Start icon. Once it has started, open a new browser window. If you installed JRun with the built-in Web server, point your browser to http://localhost:8300/CFIDE/administrator/index.cfm. If you configured an external Web server, such as Apache or IIS, point your browser to http://<your web server root>/CFIDE/administrator/index.cfm. You should now see your ColdFusion Administrator login screen.

It is possible to control server instances without having to use the JMC or the Admin server instance. Under Windows, the default servers may have been installed as services. You may check by going to Start > Settings > Control Panel > Administrative Tools > Services. The Services Management Console (see Figure 4) should display all JRun server instances that were installed as Windows services. From here you can start and stop instances, as well as setting whether they automatically run at system start.

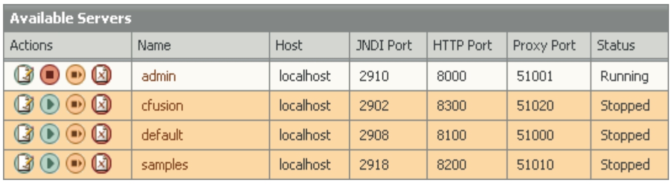
You can also control any server instance through a command shell. From your JRun /bin directory, the command

```
./jrun -<start | stop | restart> -nohup <instance name>
```

will allow you to start and stop any instance under Linux. On Windows the command

```
jrun -<start | stop | restart> -nohup <instance name>
```

will have the same results.



Available Servers

Actions	Name	Host	JNDI Port	HTTP Port	Proxy Port	Status
	admin	localhost	2910	8000	51001	Running
	cfusion	localhost	2902	8300	51020	Stopped
	default	localhost	2908	8100	51000	Stopped
	samples	localhost	2918	8200	51010	Stopped

Figure 3: Default JRun servers

Adding and Deleting Servers

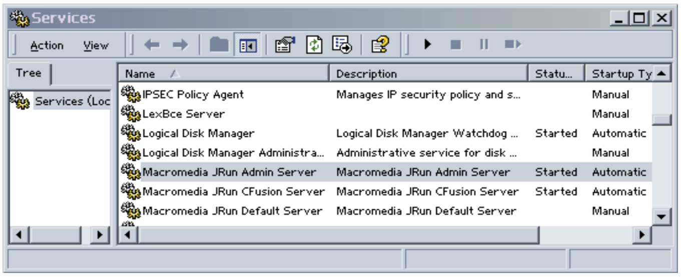
The JMC allows you to create new server instances and to delete instances. From the Available Servers panel in the JMC, select the Delete icon for the Samples server. You will be asked if it is "OK to delete this server". Click OK. You will be presented with the following choices:

- Remove the entire server directory for this server
- Remove the server without removing any associated files or directories
- Return to the server list

Removing the entire directory will remove any and all files and applications that exist in that server's directory under JRun. If you simply wish to remove the server without losing your applications files, select the second choice. Since it is the Samples server we wish to delete, it is safe to delete all of the related files (and as a security precaution, not a bad idea). Select the first choice "Remove the entire server directory for this server". The server and its files have now been removed.

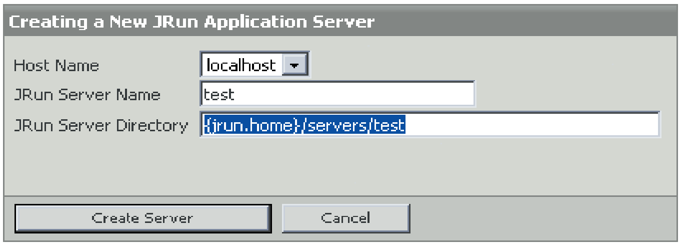
Adding a new server instance is just as simple. From the JMC's main page, select "Create New Server" located at the top of the screen, above the JRun logo. You will then be presented with a pop-up window asking you for some basic information.

For this example, type the name "test" into the JRun Server Name field. When you move to the JRun Server Directory field, it will automatically populate with a path to the JRun root's /servers directory. Leave this value as it is and click Create Server (see Figure 5).



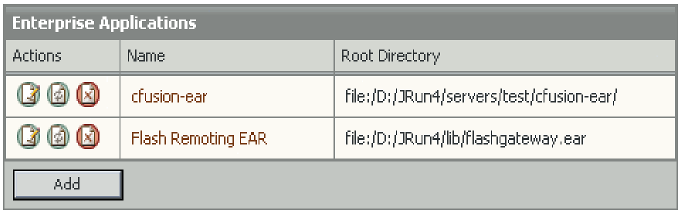
Name	Description	Statu...	Startup Ty
IPSEC Policy Agent	Manages IP security policy and s...		Manual
LexBce Server			Manual
Logical Disk Manager	Logical Disk Manager Watchdog ...	Started	Automatic
Logical Disk Manager Administra...	Administrative service for disk ...		Manual
Macromedia JRun Admin Server	Macromedia JRun Admin Server	Started	Automatic
Macromedia JRun CFusion Server	Macromedia JRun CFusion Server	Started	Automatic
Macromedia JRun Default Server	Macromedia JRun Default Server		Manual

Figure 4: The Services Management Console


Creating a New JRun Application Server

Host Name: localhost
JRun Server Name: test
JRun Server Directory: {jrun.home}/servers/test

Figure 5: New Server dialog


Enterprise Applications

Actions	Name	Root Directory
	cfusion-ear	file:/D:/JRun4/servers/test/cfusion-ear/
	Flash Remoting EAR	file:/D:/JRun4/lib/flashgateway.ear

Figure 6: Enterprise applications on test

The pop-up window will now display information on port numbers for the new server to use. Leave these at the defaults for now and click Finish. Your new server instance should now appear in the Available Servers list.

At this point you've learned how to control, delete, and create server instances on your JRun server. So how does this help you as a ColdFusion developer?

Multiple Server Instances

In the old days (before CFMX) ColdFusion was an application server. One copy of ColdFusion running on your machine served many different CFML applications to the world through your chosen Web server. Times have changed. ColdFusion is no longer an application server. It is now an application being served by a J2EE application server.

JRun is a J2EE application server, which means that it serves J2EE applications, as opposed to a Web server, which serves Web sites. In the case of CFMX 6.1, JRun delivers the ColdFusion application (the cfusion server) and the JRun Management Console application (the Admin server), as well as the default and sample applications.

Each of these applications runs in its own instance (think of each instance as a completely separate J2EE application server). Each instance exists in its own protected space, much like a desktop application does. This helps to insulate each application and keep it safe from errors in other applications. It also allows the developer to run multiple instances of the same application, each one separate but technically identical.

Now what does this mean to the ColdFusion developer? First, and most fundamental, is the ability to run multiple ColdFusion environments. Yes, you may now run many different instances of the ColdFusion application from one Enterprise ColdFusion license. This means that you can configure several instances of ColdFusion, each with its own ColdFusion Administrator. This also means that since each application runs as a separate instance, a badly coded application running in one server instance will not bring down applications running in a different instance. Multiple instances of ColdFusion also mean tighter security, as CFML applications in different instances do not share variables scopes with one another.

On the more advanced side of the fence, running multiple instances of ColdFusion has an even greater significance. You may run different versions of the ColdFusion application in different instances and even upgrade certain instances, while leaving others at an earlier version. You may also take advantage of clustering multiple instances of the same application on one box for failover protection and enhanced performance.

Installing a Separate ColdFusion Instance

If you navigate to the /servers directory in your JRun root folder (D:\JRun4\servers in our Windows example and /opt/jrun4/servers in our Linux example), you will see a folder for each server instance configured in JRun, including "test" – the server instance we just created. Open the cfusion folder and you should see a folder named cfusion-ear. This folder is the ColdFusion enterprise application and it contains everything necessary for JRun to interpret your CFML pages.

Copy the cfusion-ear folder and place it in, for example, D:\tmp on Windows or /tmp on Linux. Now open the folder "test" in the /servers directory under the JRun root. This is the folder that will contain the J2EE applications for the test server instance we just added. Copy the cfusion-ear folder from the temporary location you just placed it in and delete the folder named default-ear. Default-ear is the default enterprise application added to the new instance by JRun. For our example default-ear will not be needed and it is necessary to remove it so that applications within the archives do not conflict over the context root.

The context root is the HTTP path where the application's files can be accessed. Each deployed application under a single instance in JRun must have a separate root. A context root of "/" tells JRun that the HTTP root folder (e.g., www.domain.com/) should be used to process your application's pages. The Web application installed by JRun under default-ear is configured with a context root of "/".

Return to the JMC and click Start to start the test server instance. It will take several seconds, so be patient. Once the test server is running, click its name (or the Edit button) to go to the J2EE Components page. You should see a table listing the enterprise applications installed on test (see Figure 6).

Select the cfusion-ear application and click its name or the Edit icon. The next page will present an overview of the cfusion-ear enterprise application and its contents, including any WARs or Enterprise JavaBeans (EJBs) installed. There should be one Web application listed by the name of "Macromedia ColdFusion MX". Click the name or the Edit icon.

Now you're presented with an overview of the Macromedia ColdFusion MX Web application. Scroll down to the General Settings table and in the "Document Root" field enter the directory where you would like to place your ColdFusion documents. We're going to use JRun's internal Web server, so for our example Document Root, let's input

```
D:\test
```

on Windows or

```
/var/www/html/test
```

on Linux. For the field labeled "Context Path" enter a forward slash "/". This is the context root that tells JRun which path you would like to use to access your application via the Web. Now click Apply.

Create a new folder for your Document Root, equal to what you entered in the Document Root field for the ColdFusion MX Web application (D:\test or /var/www/html/test in our exam-

ple). In this folder create a simple CFML page, something like:

```
<cfset myVar = “It’s Alive!!!”>
<cfoutput>#myVar#</cfoutput>
```

Now surf back to the Home page of the JMC. See which HTTP port number is listed for your test server instance. Open up a browser window and surf to http://localhost:port. You should see your sample message displayed. Congratulations! You’ve successfully installed a working ColdFusion instance.

EAR vs WAR Deployment

The ColdFusion application may be deployed on JRun as either an EAR or a WAR. The method of deployment is up to the developer, but an explanation of the differences between the two is hard to come by.

First and foremost, EARs can contain multiple WARs, EJBs, XML deployment descriptors, and other J2EE components. A WAR contains all of the Web files for a single Web application, such as HTML and JSP pages, tag libraries, and JavaScript and CSS files.

You can deploy ColdFusion as a WAR under JRun, but for ColdFusion, deployment as an EAR is generally more beneficial. EAR deployment provides many more options to ColdFusion, such as the use of Web services and the ability to deploy a ColdFusion application over multiple machines.

The deployment descriptors for an EAR also include much more information than that available to a WAR (directory paths and configuration settings). For example, you must manually configure the context root when installing ColdFusion as a WAR, but installation of the ColdFusion EAR will automatically configure it in the context root of your choosing during the install process. This extra information within an EAR allows you to replicate a ColdFusion instance and all applications contained within it much quicker and easier.

Advanced Configuration

Deploying ColdFusion on JRun gives you the ability to run CFML pages and to access the CF Administrator, but some features are not configured by default. Some of these features have platform-specific dependencies or server-specific environment settings and must first be configured in order to utilize them. To configure these features, you must modify arguments to the Java Virtual Machine (JVM).

JRun stores JVM arguments in the jrun_root/bin/jvm.config file. The arguments specified in the jvm.config file will be common to all sites configured under JRun. Open this file in a text editor and you will see several different arguments defined. We are most concerned with the arguments java.library.path (the JVM library path), java.class.path (the JVM classpath), and java.args (arguments to the VM). In these arguments you will manually add paths to enable the advanced features.

To enable support for C++ CFX tags:

- Add WEB-INF/cfusion/lib to the JVM classpath

To enable Verity:

- Add WEB-INF/cfusion/lib to both the JVM classpath and library path
- Add WEB-INF/cfusion/lib/platform/bin to the JVM library path (platform will be either _ilnx21, _nti40, or _solaris depending on your server’s OS)

To add support for COM objects under Windows:

- Add WEB-INF/cfusion/jintegra/bin to the JVM library path
- Add WEB-INF/cfusion/jintegra/bin/international to the JVM library path
- Add -DJINTEGRA_NATIVE_MODE -DJINTEGRA_PREFETCH_ENUMS to the java.args JVM argument

To enable the cfreport tag and the Microsoft Access for Unicode database driver under Windows:

- Add WEB-INF/cfusion/lib to the JVM classpath

To enable charting:

- Add WEB-INF/cfusion/lib to the JVM classpath
  - For Windows, add: Xbootclasspath/a:$cf_root/WEB-INF/cfusion/lib/webchartsJava2D.jar to the java.args argument
  - For Unix, add: Djava.awt.graphicsenv=com.gp.java2d.ExGraphicsEnvironment for JDK version 1.4.0 and earlier or Djava.awt.graphicsenv=com.gp.java2d.ExHeadlessGraphicsEnvironment for JDK version 1.4.1 or later

You will notice that the classpath argument is the same for most of these features. In order to enable the binary files JRun simply needs to be made aware of the location in which ColdFusion stores them. Be aware that enabling some of these features may have the effect of enabling others.

Conclusion

In the end, working with JRun is not quite as difficult as it looks. You are given several simple tools for performing administration and configuration duties. The JRun Launcher is a convenient GUI tool for starting and stopping servers, as well as getting a quick glimpse of server status. The JRun Management Console is a tool for starting and stopping servers, deploying J2EE applications (such as the ColdFusion server environment), and creating server instances.

You’ve also seen that using these tools is not necessary. Anything that you can accomplish through them can also be accomplished through a command interface or through the editing of appropriate files. In the majority of ColdFusion installations, you will probably never touch the Launcher or JMC and may in fact never need to.

JRun opens the doors to some very powerful possibilities should you decide to take advantage of them. I’ve only scratched the surface here, but hopefully this peek under the hood has made JRun seem a little less ominous and the answers to some of your questions a little clearer.

About the Author

Benjamin Blackwell is an application developer with too many projects to list here. He is a Certified ColdFusion MX Developer and has been playing with code since 1994. Benjamin was a technical editor on the book Inside ColdFusion MX *for New Riders Publishing.*

wired@studiomachine.com

JavaScript Remote Scripting with ColdFusion MX

Build Win32-like functionality into your Web forms without resorting to Flash

By Grant Szabo

Have you ever wanted to write an HTML-based form that would behave more like a Win-32 “fat client” application than a Web page? Something like a combo box–like control where the user can type in a text field and have a list-box containing matching records auto-filter based on what was typed?

Maybe you’re not thinking of anything that fancy. What about simple drill-down selectors where a user chooses a manufacturer, and, based on that selection, a second drop-down list instantly populates, containing just the products offered by that chosen manufacturer? What’s more, you want to accomplish this without having to refresh the browser page and you aren’t planning on implementing Flash Remoting.

Enter JavaScript Remote Scripting (JSRS). JSRS is a client-side JavaScript library that uses dynamic HTML elements to make hidden remote procedure calls back to the Web server. JSRS was written by Brent Ashley in 2000 (www.ashleyit.com/rs/). It is open source and free of charge to anyone who wants to use it. It is no longer supported by Brent, but there is a community of active developers who use the library, and help is often available from these folks on Brent’s forums. JSRS works asynchronously and, according to Brent, is known to work on Win9x, WinNT/2000, WinXP, Unix/Linux/BSD, and Mac with IE4+, NS4.x, NS6.x, Mozilla, Opera7, and Galeon. There are server-side implementations for ASP, ASP.NET, ColdFusion, PerlCGI, PHP, Python, and JSP (servlet).

This article will provide concrete steps on how to implement JSRS with ColdFusion MX so you can begin taking advantage of this great tool in your ColdFusion projects.

I’ve targeted this article for intermediate to advanced ColdFusion developers who have intermediate skills with JavaScript and DHTML. If you haven’t done a ton of work with JavaScript or DHTML, don’t sweat it. There should be enough code here to help you get your first JSRS script working even without that knowledge. Working on a JSRS implementation will help you develop these skills and you’ll have fun at the same time!

Examples on the Web

First, let’s start with some examples so you can see this technology in action before you start implementing it. Every bit of JSRS code that I’ve implemented has been for corporate intranets or secure extranet applications, so unfortunately I can’t show you anything running in production. I’ve got a couple of examples on my home page for you to check out, though.

Navigate to www.quagmire.com/whiteboard.aspx to see the examples.

- ***Drill-Down Example:*** Choose “Northwind v2” from the tree view. In this example, you’ll see two drop-down lists. The first list contains employees. The second list contains regions. Pull down the first list and start randomly selecting employees. You’ll see that regions belonging to that employee automatically appear without refresh in the second drop-down list.

- ***List Filtering Example:*** Choose "Autofilter" from the tree view. This is a working example of an auto-filtering select box. The select box content changes dynamically, and without refresh, based on the criteria typed into the text box. Select a product in the list box and click Submit. Product details display. No refresh occurs. This is a pretty dramatic use of JSRS technology and should get you pumped up to do your first JSRS implementation!

Getting Started

The first thing you'll need to do is download the JSRS package for ColdFusion MX and review the source code. It's important to understand the concept of how JSRS works before attempting to implement anything. There are a few moving parts to get familiar with.

Download the JSRS source code example from www.sys-con.com/coldfusion/sourcec.cfm. The CFMX package is available without the Autofilter example at www.quagmire.com/cfm/jsrsCFMX.zip.

The following core files are included in the distribution. Let's get familiar with what they are and what they do:

- ***jsrsClient.cfm:*** The jsrsClient.cfm file is where everything starts and finishes. The critical elements that you must include in this CF template are:
  – Two JavaScript functions: a calling function and a callback function for each JSRS call
  – At least one HTML form element that uses a JavaScript event to fire your calling function
  – A <DIV> tag to which you will write the results of your server-side call
- ***jsrsSvr.cfm:*** This template processes your server-side call. This script executes in a different thread than jsrsClient.cfm.
- ***jsrsServer.cfc:*** This ColdFusion component is used to write the results of jsrsSvr.cfm to your callback function in jsrsClient.cfm. You instantiate this object and invoke the jsrsDispatch() method to do this.
- ***jsrsClient.js:*** This is the core JavaScript library written by Brent Ashley and must be included for any of this to work. You include this file in the <head> section of jsrsClient.cfm.
- ***License.txt:*** Brent Ashley's No Nonsense Copyright and License for JSRS.

How JSRS Works

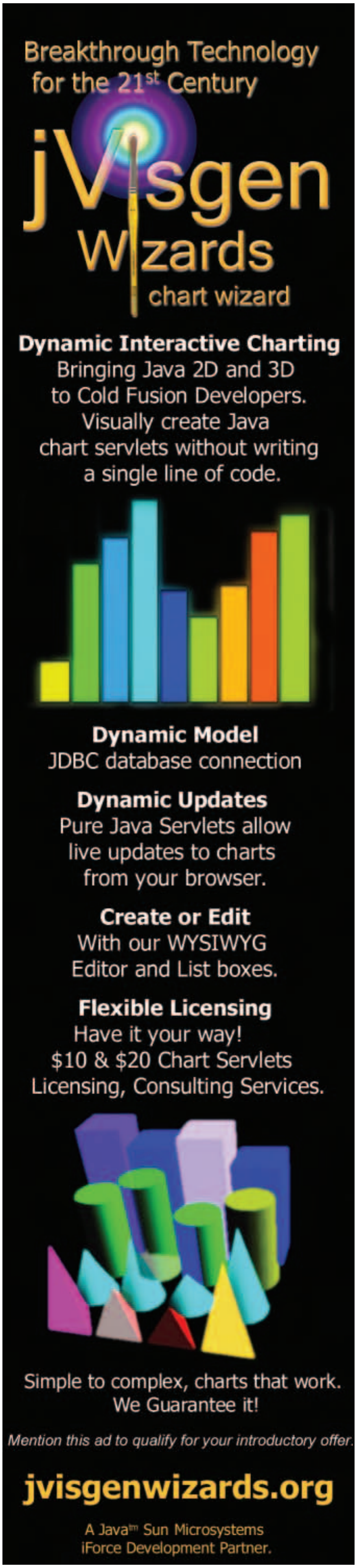
1. Within jsrsClient.cfm, an HTML form element, such as a text box is wired to a JavaScript event, such as an onKeyUp() event. This onKeyUp() event fires when the user types something in the text box. The contents of the text box are passed to the JavaScript function that is executed when the onKeyUp() event fires.
2. The JavaScript function executes a method in Brent Ashley's jsrsClient.js JSRS library named jsrsExecute(). One of the parameters passed into jsrsExecute is the name of the JavaScript callback function. You'll write your callback function in JavaScript and place that function directly beneath the function you used to call jsrsExecute(). Both of these functions will reside in jsrsClient.cfm.

3. jsrsClient.js is a blackbox. It processes the information passed in via jsrsExecute(), then does an HTTP Get or Post (depending on the client browser) to the jsrsSvr.cfm file. jsrsClient.js exists on the client (it's sitting in the browser cache). It does a post to the server. This is key to understanding what is happening with JSRS. This post is a new thread that the user executed without even knowing it.

4. jsrsSvr.cfm is a server-side processing form that you write to handle the event. jsrsSvr.cfm is a standard CFML template with some special code for handling the information it receives from jsrsClient.js. Here's where you can make a database call, execute a stored procedure, or call into one of your business objects (a CFC, for example) to do something, like return a recordset. You then need to convert this recordset into a delimited list. Once you have this list constructed, you invoke the jsrsDispatch() method of the jsrsSvr.cfc object.

5. jsrsSvr.cfc is a CFC that I wrote that takes your delimited list and writes it back to your callback function (see step 2). The process flow is now back where it all started – on jsrsClient.cfm, but now it's in your callback function.

6. jsrsClient.cfm callback function uses the JavaScript split function to split your list into a JavaScript array. You



then take these values from the array and write them into an HTML string. A little secret about JavaScript that you should know – JavaScript processes arrays about 10x faster than it processes lists. Thus, you should use arrays wherever possible. It's easy to take the array and convert it into a string by using the join method of the array object. See the jsrsClient.cfm code you downloaded for how to do this in the callback function.

7. Once you have your HTML string ready for display, simply write this to a <DIV> tag using the innerHTML property.

Putting It All Together

First, place jsrsClient.js somewhere on your Web site. I usually create a folder named "javascript" off of my webroot and place all .js files in there. That's a great place to put jsrsClient.js.

Next, place jsrsSvr.cfc in your components directory – the directory where you are storing all of your CFCs for your application.

jsrsClient.cfm

Now, start working on your jsrsClient.cfm page (you don't need to name it jsrsClient.cfm – name it whatever you want). First, set an include reference in the <HEAD></HEAD> section of your template like so:

```
<html>
<head>
<title>jsrsClient</title>
<script language="javascript" src="/javascript/jsrsClient.js"></script>
</head>
```

Next, build your Web form. Here is the HTML from the Autofilter jsrsClient.cfm file:

```
<form>
<table>
<tr>
<td class="font1Bold">Product Name: </td>
<td><input type="text" size="15" name="productNameTextBox"
class="font1" onKeyUp="fcnGetProduct(this)"></td>
</tr>
<tr>
<td></td>
<td><div id="divListBox">
<select name="ProductId" size="5" class="font1">
<cfoutput query="RS">
<option value="#ProductId#">#ProductName#</option>
</cfoutput>

</select>
</div></td>
</tr>
</table>
<div id="divDetail"></div>
</form>
```

Next, you'll write your calling and callback functions (nest these inside <script language="javascript"></script> tags). The Autofilter example makes two JSRS calls so there are two calling functions and two callback functions. The first RPC call fires with the onKeyUp() event of the text box. The second RPC call fires when the user selects a product in the list box.

```
//autofilter JSRS Calling Function (Fires from onKeyUp() event)
function fcnGetProduct(obj) {
var aParams = new Array(1);
aParams[0] = obj.value;
jsrsExecute( '/jsrs/jsrsSvrMX.cfm', fcnCallbackGetProduct,
'getProduct', aParams, 0);
                                                  }
```

Note the parameters being passed to jsrsExecute. These parameters are:

1. ***'/jsrs/jsrsSvrMX.cfm':*** This is the template that will handle the server-side processing for JSRS.
2. ***'fcnCallbackGetProduct':*** This is the name of the callback function.
3. ***'getProduct':*** This is the <cfcase> action argument that will be used in jsrsSvr.cfm.
4. '***aParams':*** jsrsExecute() requires that parameters be passed in via a JavaScript array.
5. ***'0':*** This is a bit field set to either 0 or 1. If you set it to 1, JSRS will make an iFrame visible that it places on jsrsClient.cfm, providing a "window" to execute your jsrsSvr.cfm template. When you are debugging your jsrsSvr.cfm page, you'll set this value to 1 so that you can see any errors that your jsrsSvr.cfm page might be throwing. Once you have everything debugged, set this value back to 0 and effectively hide all the back-end processing under the blanket.

Now you can start putting together your callback function. You'll probably need to spend some time in jsrsSvr.cfm at this point as well to build your server-side logic, as your callback function will receive its data from there. There is no right order to this process – just work the way that makes the most sense to you.

```
//autofilter JSRS Callback Function
function fcnCallbackGetProduct(optsStr) {
    aCallback = optsStr.split(delim); //split takes the string/list
    //returned from the RPC and converts it into an array
    var strHTMLArr = new Array(); //instantiate array to hold HTML
    var strHTML = ""; //initialize strHTML
    strHTMLArr[0] = '<select id="ProductId" name="ProductId" size="5"
    class="font1">';
    for (i=1; i<=aCallback.length; i++) {
          if(i % 2 == 0) {
            //value of loop is even
                        strHTMLArr[i] = aCallback[i-1] + '</option>';
          }
          else {
            //value of loop is odd
                        if (i == 1) {
                        //highlight the first item in the array
                        strHTMLArr[i] = '<option value="' + aCallback[i-
                        1] + '" SELECTED>';
                        }
```

A LIMITED TIME SAVINGS OFFER FROM SYS-CON MEDIA

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

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

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $\$275^{00}$ Pay only $175 for a 1 year subscription plus a FREE CD

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

6-Pack

Pick any 6 of our magazines and save up to $\$350^{00}$ Pay only $395 for a 1 year subscription plus 2 FREE CDs

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

9-Pack

Pick 9 of our magazines and save up to $\$400^{00}$ Pay only $495 for a 1 year subscription plus 3 FREE CDs

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

CALL TODAY! 888-303-5282

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

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

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

☐ **MX Developer's Journal**

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

☐ **Linux World Magazine**

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

☐ **Java Developer's Journal**

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

☐ **Web Services Journal**

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

☐ **.NET Developer's Journal**

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

☐ **XML-Journal**

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

☐ **WebLogic Developer's Journal**

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

☐ **ColdFusion Developer's Journal**

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

☐ **Wireless Business & Technology**

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

☐ **WebSphere Developer's Journal**

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

☐ **PowerBuilder Developer's Journal**

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

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

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

SYS-CON MEDIA

```
                    else{
                                strHTMLArr[i] = '<option value="' +
                                aCallback[i-1] + '">';
                    }
        }
    }
    strHTMLArr[i++] = '</select>';
    strHTMLArr[i++] = '<input type="button" value="View Detail"
    class="submit" onClick="fcnGetProductDetail()">';
    strHTML = strHTMLArr.join(""); //join() is the opposite of split().
    //It converts an array to a list using a delimiter, in this case ""
    //(no delimiter)
    document.getElementById("divListBox").innerHTML = strHTML; //write
    //out the completed html string to the div tag
}
```

Let's walk through the callback function. The first line is receiving our delimited list back from jsrsDispatch() (see jsrsSvr.cfc). We take that value, which I call "optsStr", and use the JavaScript split method to convert the list to a JavaScript array. Next, I initialize two JavaScript vars, strHTMLArr, and strHTML. strHTMLArr is a JavaScript array that will be used to hold HTML strings.

Now, use a JavaScript FOR LOOP to iterate through the callback array, writing out HTML along the way. Once the loop is finished, I write in an HTML button (that will call the next JSRS calling function) and then use the join method of the callback array object. By specifying "" as the parameter of the join method, I am effectively concatenating the JavaScript Array into a string as opposed to a list.

The remaining JavaScript functions are not presented here, but are available for download from www.sys-con.com/coldfusion/sourcec.cfm.

jsrsSvr.cfm

Now you'll write your jsrsSvr.cfm file. You can put this file in the same directory as your jsrsClient.cfm file *or* you can place it out in some global directory where you will place *all* of your JSRS methods (such as /jsrs located off of your webroot). If you use the latter method, you can use just this one single jsrsSvr.cfm file to handle requests from all of your jsrsClient.cfm files used throughout your application – you just include a new <cfcase></cfcase> statement for each JSRS RPC.

I would suggest that you simply use the jsrsSvr.cfm file included in the package you downloaded and modify the code inside the <case></case> statements. Everything else can be left alone – and should be left alone unless you really know what you're doing. I've optimized the code in this file to run as fast as possible – using CF arrays.

Here's the <cfcase> for getProduct. Remember "getProduct" was a parameter that was passed into jsrsExecute() in the calling function.

```
<cfcase value="getProduct">

    <cfscript>
    //get recordset
    objNorthwind = CreateObject('component',
'quagmireV3.components.northwind');
    RS = objNorthwind.getProduct(aParams[1]);

    //build return string
    for(i=1; i LTE RS.RecordCount; i=i+1) {
           request.rStr = request.rStr & RS.ProductId[i] &
           variables.delim & RS.ProductName[i] & variables.delim;
    }

    //remove trailing delimiter
    variables.rStrLen = Len(request.rStr);
    if(variables.rStrLen GT 0)
           request.rStr = RemoveChars(#request.rStr#,
#variables.rStrLen#,  1);
    </cfscript>

</cfcase>
```

There is no requirement to use CFScript. If you're not comfortable with CFScript, use regular CFML. I prefer CFScript syntax over regular CFML so I tend to use it a lot in my ColdFusion development.

Walking through this code, I'm first getting a recordset of matching products using the "getProduct" method of my northwind.cfc. Here's that function:

```
<cffunction name="getProduct" access="public" returntype="query">
<cfargument name="ProductName" type="string" required="yes">
<cfstoredproc procedure="spNW_Sel_Products_findByProductName" data-
source="#this.DSN#">
<cfprocparam type="In" cfsqltype="CF_SQL_VARCHAR"
variable="@ProductName" value="#arguments.productName#" null="No">
<cfprocresult name="RS1">
</cfstoredproc>
<cfreturn RS1>
</cffunction>
```

Next, I build a return string by using a FOR LOOP (you may use <cfoutput></cfoutput> instead if you wish). What's important is that you concatenate the delimited list.

Last, I whack off the trailing delimiter. The variable request.rStr is now ready to be handed off to jsrsSvr.cfc so that it can be posted back to our callback function.

Conclusion

I hope that this concise overview of implementing JavaScript Remote Scripting will help advance your ability to develop UIs for the Web using ColdFusion. JSRS is a very powerful and effective tool for building Win32-like functionality into your Web forms without resorting to Flash.

About the Author

Grant Szabo is a senior software architect for Sterling Creek Software (www.sterlingcreek.net) where he delivers Microsoft .NET and ColdFusion solutions for select clients. Grant worked for Allaire, (later Macromedia) as director, worldwide professional services in 2000–2001. He is Macromeda Certified in CFMX and CF5, and holds several Microsoft Certifications (MCSD, MCDBA, MCSE, and MCSA).

szabogi@yahoo.com

SYS-CON MEDIA

304,187 of the World's Foremost IT Professionals

DIRECT MAIL, EMAIL OR MULTI-CHANNEL

Target CTOs, CIOs and CXO-level IT professionals and developers who subscribe to SYS-CON Media's industry leading publications

Java Developer's Journal... The leading publication aimed specifically at corporate and independent java development professionals

LinuxWorld Magazine... The premier monthly resource of Linux news for executives with key buying influences

Web Services Journal... The only Web Services magazine for CIOs, tech, marketing & product managers, VARs/ISVs, enterprise/app architects & developers

XML-Journal... The world's #1 leading XML resource for CEOs, CTOs, technology solution architects, product managers, programmers and developers

SPECIAL INSERT: WEB SERVICES EDGE CONFERENCE & EXPO
JAVA DEVELOPERS JOURNAL
HyperThreading Java
LINUXWORLD MAGAZINE
Why Migrate to Linux?
Facing Microsoft's Fury...
Taking the Geek Out of Linux
A comprehensive solution for enterprise IT
FREE
WEB SERVICES EDGE WEST SEPT. 30 – OCT. 2
WebServices JOURNAL
Future-Proofing Solutions
XML JOURNAL
Enterprise Solutions • Content Management • XML Labs
Next-Generation Web Presentation
POWERBUILDER AND WEB SERVICES
PowerBuilder Developer's Journal
CALLING A PB COMPONENT AS A WEB SERVICE
JOCELYNE ATTAL LOOKS AT THE FUTURE OF WEBSPHERE
WebSphere DEVELOPER'S JOURNAL
Migrating WebSphere...
SPECIAL INSERT: WEB SERVICES EDGE CONFERENCE & EXPO
COLDFUSION Developer's Journal
Two great tastes together!
Windows Server 2003 and CFMX 6.1
MOVING AHEAD FOR THE 21ST CENTURY
BEA WebLogic DEVELOPER'S JOURNAL
Java Connector Extensibility Architecture
XML DOCUMENTATION: SLAYING THE DOCUMENTATION DRAGON
.NET .NET JOURNAL
Bringing the Monkey to the iPAQ
WWW.DOTNETDEVELOPERSJOURNAL.COM
INSIDE SECTION
Wi-Fi • MMS • ARPU • 802.11 • VAS • PKI • RFID • 3G
wireless BUSINESS TECHNOLOGY
The Wireless Weatherman

PowerBuilder Developer's Journal... The only PowerBuilder resource for corporate and independent enterprise client/server and web developers

WebSphere Developer's Journal... The premier publication for those who design, build, customize, deploy, or administer IBM's WebSphere suite of Web Services software

ColdFusion Developer's Journal... The only publication dedicated to ColdFusion web development

WebLogic Developer's Journal... The official magazine for BEA WebLogic application server software developers, IT management & users

.NET Developer's Journal... The must read iTechnology publication for Windows developers & CXO management professionals

Wireless Business & Technology... The wireless magazine for key corporate & engineering managers, and other executives who purchase communications products/services

Recommended for a variety of offers including Java, Internet, enterprise computing, e-business applications, training, software, hardware, data back up and storage, business applications, subscriptions, financial services, high ticket gifts and much more.

NOW AVAILABLE!
The SYS-CON Media Database
304,187 postal addresses

e-POSTDIRECT®
Your iMarketing Source

For email information:
contact Frank at 845-731-3832
frank.cipolla@epostdirect.com
epostdirect.com 800-409-4443 fax845-620-9035

For postal information:
contact Kevin at 845-731-2684
kevin.collopy@edithroman.com
edithroman.com 800-223-2194 fax845-620-9035

List Brokerage & Management
EDITH ROMAN
Data Processing • Email Marketing

CFX-Advanced VPS Hosting from CFXHosting

A flexible and cost-effective hosting solution

Selene Bainum

There comes a time in the life of almost every developer when he or she needs to host a Web site. Unfortunately, not all of us can have a T1 running to our basement with a dual-processor Pentium 4 running our site. No, we developers – and businesses – must look elsewhere for help. Unless you have a lot of money, that typically means you'll be part of a shared hosting environment.

In a traditional shared hosting environment, your Web site will be one of perhaps hundreds running on the same server. You won't get access to the ColdFusion Administrator and don't even think about playing around in the operating system. Disk space and data transfers are limited, and additional data transfer rates can quickly raise your monthly fees.

Many advanced developers and small-to-medium companies often feel restricted by shared hosting environments, but the next step up – buying dedicated servers – is often out of the question. VPS hosting can help bridge that gap.

New Options from an Old Friend

CFXHosting, one of the leading ColdFusion hosting providers, has added another item to its menu of options: CFX-Advanced VPS Hosting. VPS stands for Virtual Private Server, and it is so virtual you won't be able to tell the difference from the real thing. And for $189 a month, you won't even try to look.

Unlike traditional shared hosting environments, VPS hosting allows you full access to your virtual server from Terminal Server, access to the file system, and access to the Microsoft Management Console. You can even reboot your virtual server when you like.

How is this possible? CFXHosting uses their own software to allow multiple instances of an operating system on the same physical server. This allows you to have your own operating system that is shared by no one else. An individual physical server will never have more than four customers or operating system instances installed.

The configuration comes with 3GB of disk space free for your data, 40GB of data transfers per month, 256MB of dedicated RAM, and one static IP address. You'll also get free support for the first 60 days, with six free support incidents per month thereafter.

Impressive Software Support

VPS hosting supports a number of different software options, starting with Windows 2000 or Windows 2003 Servers at the core, running either ColdFusion 5.0 or ColdFusion MX (6.1) Professional. All Macromedia Studio MX products are supported, including the newest member of the Studio MX family – Macromedia Contribute. Since the operating system is yours alone, you get full ColdFusion tag support (with the exception of CFEXECUTE and CFREGISTRY), as well as the ability to use any CF or CFX custom tag you wish to install. You can even use ColdFusion RDS to access your code.

If you care enough about your site to pay for VPS hosting, chances are you'll want a database more robust than Microsoft Access. That's good, because Microsoft SQL Server can be set up for you. With SQL Server and ColdFusion control, you can create as many databases and data sources as you like. And of course, Microsoft Access is supported as well.

You'll also get 50 e-mail aliases, e-mail forwarding, and catchall e-mail addresses. The server is WAP, CGI, VBScript, and Java enabled; ASP support is optional. You even have access to XML Web services and the raw data logs.

Monitoring Your Site Traffic

Included in the VPS hosting package is LiveStats Statistic log analyzer (see Figure 1). This product offers some cool features, such as the ability to analyze your Web traffic in real time and to see who is currently on your site. The interface to LiveStats isn't quite as nice as WebTrends, but since you have the ability to install software on your virtual server, you have the option of installing it yourself.

Uptime Guaranteed

One nice aspect of CFXHosting is their 99.9% uptime guarantee. If for any reason your Web site is not available for HTTP access by third parties, you'll qualify for a credit. In fact, if the uptime falls below 89.9%, you'll receive a full credit for that month. Of course there are a few restrictions, like war and insurrection, but I think I'll take my chances.

First-Rate Hosting Facility

Worried that your site isn't safe in a hosting environment? You needn't be with CFXHosting. While it is impossible to guarantee against all hackers and probes, the Cisco Secure PIX Firewall between your site and the rest of the world will certainly help you sleep better at night.

But not all attacks come over the network; your site isn't safe if the physical location at which it is stored is not safe. But again, CFXHosting has it covered. Between 24/7 security personnel, video cameras, electromagnetic access cards, and biometric hand scanners, you can be assured your site is physically safe. There is also a redundant power grid so the center will stay up even if an entire power grid goes down. The temperature and humidity of the facility is closely monitored and the FM-200 Fire Suppression system will kick in, in the unlikely event of a fire.

Backup and Restore

Unlike many developers who sometimes neglect to make backups of their precious applications, CFXHosting has a high-speed, high-volume backup system that backs up every server in their network, though these services will cost you extra. Daily and weekly backup options are available, with the setup and monthly fees dependent on how much data is being backed up. These fees can quickly exceed your monthly hosting rates, but if your database contains rapidly changing, vital information, it may well be worth it.

À La Carte Add-Ons

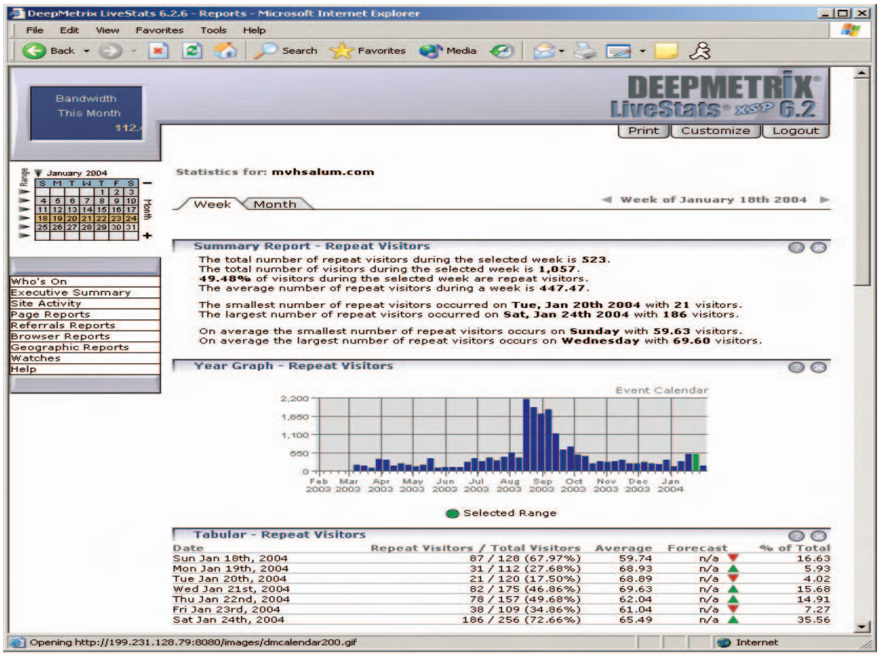
Backups aren't the only additions you can make to your service. You can also purchase additional Web space, additional bandwidth, additional SQL Server space, custom-name servers, and support incident blocks. Again, these extras can quickly add up, but since they are all billed on a monthly basis, you only have to pay for them when you need them.

Getting Started

The process of getting set up for VPS hosting is very simple and fairly quick. At press time, there was a 30-day free trial period, after which there is no obligation to continue, however, many customers choose to. To get started, fill out the free trial form and you'll be contacted by CFXHosting within 24 hours. After answering a few questions, their install team gets to work setting up your environment. Setup takes between 3 and 5 business days, depending on their current request load, and can even take as little as one business day. After that, you are ready to go.

Sign Me Up, Scotty

VPS Hosting is certainly one of the best alternatives to restrictive shared-hosting environments and dedicated or colocated servers. With the ability to control the operating system and software and to administer ColdFusion and SQL Server, you'll forget that you don't have your own server. The peace of mind in knowing that you are the only one who has access to the operating system on which your code resides is very important as well. If you are looking for a flexible, yet cost-effective hosting solution, VPS Hosting from CFXHosting is the way to go!


DeepMetrix LiveStats 6.2.6 - Reports - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Bandwidth This Month 112.
DEEPMETRIX LiveStats XSP 6.2
Print Customize Logout
Range January 2004 Month
S M T W T F S
Who's On
Executive Summary
Site Activity
Page Reports
Referrals Reports
Browser Reports
Geographic Reports
Watches
Help
Week Month
Week of January 18th 2004
Event Calendar
2,200
1,650
1,100
550
0
Feb 2003 Mar 2003 Apr 2003 May 2003 Jun 2003 Jul 2003 Aug 2003 Sep 2003 Oct 2003 Nov 2003 Dec 2003 Jan 2004
Selected Range
Internet

Statistics for: **mvhsalum.com**

Summary Report - Repeat Visitors

The total number of repeat visitors during the selected week is **523**.
The total number of visitors during the selected week is **1,057**.
49.48% of visitors during the selected week are repeat visitors.
The average number of repeat visitors during a week is **447.47**.

The smallest number of repeat visitors occurred on **Tue, Jan 20th 2004** with **21** visitors.
The largest number of repeat visitors occurred on **Sat, Jan 24th 2004** with **186** visitors.

On average the smallest number of repeat visitors occurs on **Sunday** with **59.63** visitors.
On average the largest number of repeat visitors occurs on **Wednesday** with **69.60** visitors.

Year Graph - Repeat Visitors

Tabular - Repeat Visitors

Date	Repeat Visitors / Total Visitors	Average	Forecast	% of Total
Sun Jan 18th, 2004	87 / 128 (67.97%)	59.74	n/a ▼	16.63
Mon Jan 19th, 2004	31 / 112 (27.68%)	68.93	n/a ▲	5.93
Tue Jan 20th, 2004	21 / 120 (17.50%)	68.89	n/a ▼	4.02
Wed Jan 21st, 2004	82 / 175 (46.86%)	69.63	n/a ▲	15.68
Thu Jan 22nd, 2004	78 / 157 (49.68%)	62.04	n/a ▲	14.91
Fri Jan 23rd, 2004	38 / 109 (34.86%)	61.04	n/a ▼	7.27
Sat Jan 24th, 2004	186 / 256 (72.66%)	65.49	n/a ▲	35.56

Opening http://199.231.128.79:8080/images/dmcalendar200.gif

Figure 1: LiveStats sample report

About the Author

Selene Bainum created and maintains webtricks.com, a premier ColdFusion tutorial site. She is currently a senior Web developer with INPUT.

selene@webtricks.com

Product Information

CFX-Advanced VPS Hosting from
CFXHosting
P.O. Box 22789
Louisville, KY 40252-0789

Phone: 866 CFX-HOST
Fax: 502 568.3934
E-mail: feedback@cfxhosting.com
Web: www.cfxhosting.com

Test Environment
HP Pavilion zt1180, 1.2GHz P-III, ColdFusion MX, 512MB RAM, Windows XP Professional, MS SQL Server 2000

Pricing
After a 30-day free trial, the monthly fee is $189. There is also a $189 setup fee.

Product Snapshot

Target audience
Developers and organizations looking for a better alternative to the restrictions of shared hosting environments without the costs of a dedicated or colocated server.

VPS Features

- ColdFusion Administrator access
- Terminal services and Telnet access
- SSL Secure Server available
- 3GB manageable disk space
- 40GB data transfer included
- 256MB dedicated RAM
- Ability to host up to 50 Web sites, including 10 free domains
- 100MB Microsoft SQL Server 200 database space
- Unlimited DSNs
- 99.9% guaranteed uptime
- 24/7/365 support

Supported Software

- ColdFusion 5.0 or MX Professional
- Windows 2000, 2003 Server
- Macromedia Studio MX
- LiveStats Statistics
- Real Server/Windows Media
- Custom tags
- ColdFusion RDS
- Custom software installs

COLDFUSION DEVELOPER'S JOURNAL READERS' CHOICE AWARDS

And the Winners Are…

Best Book

Winner

ColdFusion MX Web Application Construction Kit

Ben Forta/Macromedia Press

The original and still best-selling book on ColdFusion.

www.forta.com/books/0321125169

Best CF Web Service

Winner

CF Tip-of-the-Day

Ben Forta

Ben Forta's ColdFusion Tip-of-the-Day as a Web service.

www.forta.com/cf/tips/syndhowto.cfm

Best Consulting Service

Winner

TeraTech Consulting

TeraTech

TeraTech is the leading ColdFusion consulting company, helping companies, government agencies, and associations with ColdFusion since version 1.5 over seven years ago. Winner of the 2002 *CFDJ* award for best consulting company, TeraTech uses modern open methods such as Fusebox to deliver superior quality software in a timely way.

www.teratech.com

Best Content Management Tool

Winner

Macromedia Contribute

Macromedia

With Macromedia Contribute now anyone can easily update and publish content to existing Web sites without knowing HTML. Save time and money by having content owners update and manage Web content themselves, leaving Web professionals free to focus on Web site design, usability, and application functionality. Maximize productivity by updating content offline, then publishing changes when reconnected to the Internet. Macromedia Contribute uses the same code-generation engine as Dreamweaver MX, so changes made by contributors won't insert extraneous code or reformat existing code.

www.macromedia.com

Best Custom Tag or Component

Winner

soEditor

SiteObjects

soEditor turns an ordinary <textarea> form field into a dynamic WYSIWYG browser-based HTML editor. Both the Pro and Lite versions of soEditor add comprehensive word processor–like features to your Web sites and Web applications. soEditor makes it possible for anyone to create, edit, and maintain Web content without the need for HTML knowledge or programming expertise.

www.siteobjects.com

Best E-Business Software

Winner

FuseTalk

FuseTalk

FuseTalk is the leading provider of premium discussion forums and online collaboration applications for Macromedia ColdFusion environments. FuseTalk now delivers Section 508, digest subscriptions, reporting, offline capabilities, XML export, private topics, subcategories, external authentication, spell check, and virtual meeting capabilities (chat, video/audio, and whiteboard).

www.fusetalk.com

Best Education & Training

Winner

CFMX Exam Buster

CentraSoft

CFMX Exam Buster is the ultimate testing tool specifically designed to help ColdFusion MX developers earn their ColdFusion MX Certification through the Macromedia Certified Professional Program. CFMX Exam Buster has over 750 questions designed to help you prepare for all aspects of the actual ColdFusion MX Certification exam. CFMX Exam Buster allows you to take 11 different tests, includes topic tests for your weak areas, and offers the ability to create random tests.

www.centrasoft.com

Best Web Application

Winner

FuseTalk

FuseTalk

FuseTalk is the world's leading discussion forum and collaboration software solution for ColdFusion environments. The new FuseTalk Community Edition now delivers streaming online interaction with Virtual Meeting modules for video/audio, white-boarding, and chat.

www.fusetalk.com

Best Web Development Tool

Winner

Dreamweaver MX

Macromedia

Dreamweaver MX is the professional choice for building Web sites and Internet applications. Now every member of your development team – designers, developers, and programmers – can work in a single integrated environment to create, build, and manage Web sites and Internet applications. Macromedia Dreamweaver MX combines its renowned visual layout tools with the rapid Web application development features of Dreamweaver UltraDev and the extensive code-editing support of Macromedia HomeSite. So the world's best way to create professional Web sites is now the easiest way to build powerful Internet applications.

www.macromedia.com

Best Web Hosting

Winner

Edge Web Hosting

Edge Web Hosting

Edge Web Hosting is a Macromedia Alliance hosting partner and featured CF5 and CFMX

hosting facility focusing on the needs of CF developers.
www.edgewebhosting.net

Best Web Site or Community

Winner
Macromedia DevNet
Macromedia
Macromedia DevNet, formerly known as the Designer & Developer Center, is a developer Web portal that provides over 400 articles – with 20–30 new articles, sample applications, and resources added each month – as well as tips, tutorials, and discussion forums to provide a comprehensive set of tools, skill-building resources, and best practices to broaden the knowledge and abilities of development teams.
www.macromedia.com

Most Innovative CF Application

Winner
BlueDragon
New Atlanta Communications
Redeploy, integrate, and extend CFML on J2EE and .NET.
www.newatlanta.com

BEST BOOK

Winner
ColdFusion MX Web Application Construction Kit
Ben Forta/Macromedia Press
www.forta.com/books/0321125169

First Runner-up
Advanced ColdFusion MX Application Development
Ben Forta/Macromedia Press
www.forta.com/books/0321127102

Second Runner-up
Fusebox: Developing ColdFusion Applications
Jeff Peters and Nat Papovich/New Riders
www.thirdwheelbikes.com/book

Third Runner-up
ColdFusion MX: The Complete Reference
Jeffry Houser/Osborne
www.instantcoldfusion.com

BEST CF WEB SERVICE

Winner
CF Tip-of-the-Day
Ben Forta
www.forta.com/cf/tips/syndhowto.cfm

First Runner-up
ColdFusion MX Hosting
CFM-Resources.Com
www.cfm-resources.com

Second Runner-up
BillingOrchard
BillingOrchard
www.billingorchard.com

Third Runner-up
DailyReading
Aspiring Geek
www.aspiringgeek.com

BEST CONSULTING SERVICE

Winner
TeraTech Consulting
TeraTech
www.teratech.com

First Runner-up
Fig Leaf Consulting Services
Fig Leaf Software
www.figleaf.com

Second Runner-up
PabloSoft.Com
PabloSoft.Com
www.pablosoft.com

Third Runner-up
Vshift Consulting
Vshift
www.vshift.com

BEST CONTENT MANAGEMENT TOOL

Winner
Macromedia Contribute
Macromedia
www.macromedia.com

First Runner-up
CommonSpot Content Server
PaperThin
www.paperthin.com

Second Runner-up
ActivEdit
ActivSoftware
www.cfdev.com/activedit

Third Runner-up
soEditor
SiteObjects
www.siteobjects.com

BEST CUSTOM TAG OR COMPONENT

Winner
soEditor
SiteObjects
www.siteobjects.com

First Runner-up
ColdFusion Tool Box
Level 2 Labs
www.coldfusiontoolbox.com

Second Runner-up
MyAccessAdmin 2.0
Aspiring Geek
www.aspiringgeek.com

Third Runner-up
intelliCALENDAR
oprea.org
www.oprea.org

BEST E-BUSINESS SOFTWARE

Winner
FuseTalk
FuseTalk
www.fusetalk.com

First Runner-up
AbleCommerce 4.3 for ColdFusion MX
AbleCommerce
www.ablecommerce.com

Second Runner-up
CFX_XML Parser
ActivSoftware
www.cfdev.com/xml/xml-parser

Third Runner-up
inFusion Mail Server (iMS)
On-Line Data Solutions
www.coolfusion.com/products/ims

BEST EDUCATION & TRAINING

Winner
CFMX Exam Buster
CentraSoft
www.centrasoft.com

First Runner-up
TeraTech Training
TeraTech
www.teratech.com

Second Runner-up
EasyCFM.Com
EasyCFM.Com
www.easycfm.com

Third Runner-up
Macromedia DevNet
Macromedia
www.macromedia.com

BEST WEB APPLICATION

Winner
FuseTalk
FuseTalk
www.fusetalk.com

First Runner-up
soEditor
SiteObjects
www.siteobjects.com

Second Runner-up
ActivEdit
ActivSoftware
www.cfdev.com/activedit

Third Runner-up
eWebEditPro
Ektron
www.ektron.com

BEST WEB DEVELOPMENT TOOL

Winner
Dreamweaver MX
Macromedia
www.macromedia.com

First Runner-up
Macromedia Studio MX
Macromedia
www.macromedia.com

Second Runner-up
CFXGraphicsServer
TeraTech
www.cfxgraphicsserver.com

Third Runner-up
Macromedia Flash MX
Macromedia
www.macromedia.com

BEST WEB HOSTING

Winner
Edge Web Hosting Cold Fusion Shared and Dedicated Hosting
Edge Web Hosting
www.edgewebhosting.net

First Runner-up
CFXHosting ColdFusion Hosting
CFXHosting
www.CFXHosting.com

Second Runner-up
ColdFusion Web Hosting
CrystalTech Web Hosting
www.crystaltech.com

Third Runner-up
CFM-Resources.Com
CFM-Resources
www.cfm-resources.com

BEST WEB SITE OR COMMUNITY

Winner
Macromedia DevNet
Macromedia
www.macromedia.com

First Runner-up
EasyCFM.Com
EasyCFM.Com
www.easycfm.com

Second Runner-up
House of Fusion site and lists (including Fusion Authority)
House of Fusion
www.houseoffusion.com

Third Runner-up
CFConf
TeraTech
www.cfconf.org

MOST INNOVATIVE CF APPLICATION

Winner
BlueDragon
New Atlanta Communications
www.newatlanta.com

First Runner-up
CommonSpot Content Server
PaperThin
www.paperthin.com

Second Runner-up
PhotoFolio
Codesweeper
www.codesweeper.com

Third Runner-up
CFM-Resources.Com - CFM Advanced Security v1.0
CFM-Resources
www.cfm-resources.com

continued from page 10

Without a strict separation of your application text from text presentation and CF code, you will end up with one set of source code per locale – a management nightmare. Translation management is another often overlooked G11N task that benefits from the full separation of application text; handing a translator a file full of CF code is just asking for trouble. The normal (i.e., Java) method for handling this is through the use of resource bundles (per locale files holding key/value pairs of application text). You can see an example of this technique in Ray Camden's article, "Add Localization to Your Web Site" (*CFDJ*, Vol. 6, issue 2).

Calendars

Most CF developers are familiar with the Gregorian calendar (though perhaps unaware of its designation). While it's used in most English-speaking countries, it's certainly not the only calendar in use. A fully globalized CFMX application must handle non-Gregorian calendars such as Buddhist, Chinese, Hijri, Japanese, and Hebrew calendars. Besides day and month names varying from calendar to calendar, year numbering will also vary. For instance, the Gregorian calendar year of 2004 is 2547 in the Buddhist Era in Thailand. The day a week starts with will also vary from calendar to calendar, as will the actual length of a "month," making date calculations problematic. If this all sounds a bit too much, I'm in complete agreement. I frequently use and strongly recommend using IBM's ICU4J (http://oss.software.ibm.com/icu4j) to handle calendars as well as holidays (calculating the Easter holiday, for instance, isn't trivial).

Time Zones

Your application should store date/time data as UTC. This gives your application flexibility if you need to distribute it across many time zones or you have the need to "cast" data to time zones besides the CF server's.

Conclusion

The Internet's future growth will most likely take place in regions outside North America, which is fast approaching total saturation. The user base in these growth areas is not one homogeneous locale but consists of dozens – perhaps hundreds – of locales. A globalized CFMX application will therefore need to efficiently handle locales and provide locale-specific functionality to be successful in these "new" international markets.

"There are $3.6 trillion in online transactions going on outside the U.S. – and that amount is growing"

The next article in this series will examine char encodings and CFMX in detail. It aims to deal with the different kinds of encoding, BIDI (bidirectional text), and the use of CSS in G11N applications, and to go into further detail on why we should all just use Unicode.

Resources

- *Forrester Projects $6.8 Trillion for 2004:* www.forrester.com/ER/Press/Release/0,1769,277,FF.html
- *Internet World Usage Stats:* www.internetworldstats.com/stats.htm
- *Bilingual ads hit airwaves* (San Antonio Express-News*):* http://news.mysanantonio.com/story.cfm?xla=saen&xlb=110&xlc=1078719
- *NITLE Weblog Census:* http://www.blogcensus.net
- *Character sets and codepages:* www.microsoft.com/typography/unicode/cscp.htm
- *What is Unicode:* www.unicode.org/standard/WhatIsUnicode.html
- *National Weather Service Climate Glossary:* www.cpc.ncep.noaa.gov/products/outreach/glossary.html
- *What is Universal Time?:* http://aa.usno.navy.mil/faq/docs/UT.html

About the Author

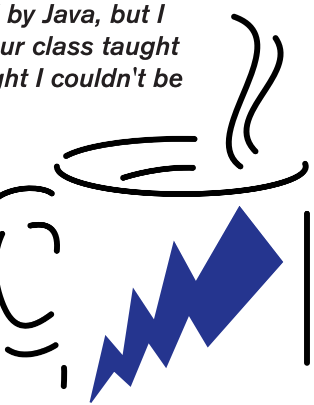
Paul Hastings, who after nearly 20 years of IT work is now a perfectly fossilized geologist, is CTO at Sustainable GIS, an agile consulting firm specializing in Geographic Information Systems (GIS) technology, ColdFusion Internet and intranet applications for the environment and natural resource markets, and of course globalization. Paul is based in Bangkok, Thailand, but says that's not nearly as exciting as it sounds.

paul@sustainableGIS.com

CFDJ Advertiser Index

ADVERTISER	URL	PHONE	PAGE
21C JVISGENWIZARDS	WWW.JVISGENWIZARDS.ORG	650-641-1744	37
ACTIVEPDF	WWW.ACTIVEPDF.COM	866.GoToPDF	4
CFDYNAMICS	WWW.CFDYNAMICS.COM	866.233.9626	19
CFXHosting	WWW.CFXHOSTING.COM	866-CFX-HOST	6
CTIA	WWW.CTIASHOW.COM	301-694-5243	Cover III
EDGE WEB HOSTING	WWW.EDGEWEBHOSTING.NET	1.866.EDGEWEB	Cover II
HAL HELMS, INC	WWW.HALHELMS.COM		47
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM/CFDJ	877.248.HOST	21
INTERMEDIA.NET	WWW.INTERMEDIA.NET	800.379.7729	Cover IV
MACROMEDIA	WWW.MACROMEDIA.COM/INTO	415-252-2000	11
MACROMEDIA	WWW.MACROMEDIA.COM/GO/2004	415-252-2000	17
PAPERTHIN	WWW.PAPERTHIN.COM	800.940.3087	29
TERATECH	WWW.TERATECH.COM	301-424-3903	3
WEBCORE TECHNOLOGIES	WWW.WEBCORETECH.COM	877-WCT-HOST	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.



ColdFusion User Groups

For more information go to...
http://www.macromedia.com/cfusion/usergroups

New England

New Hampshire
Northern N.E. MMUG
www.mmug.info

Massachusetts
Boston, MA CFUG
www.cfugboston.org

Rhode Island
Providence, RI CFUG
www.ricfug.com

Vermont, Montpelier
Vermont CFUG
www.mtbytes.com/dfug/index.htm

Midatlantic

New Jersey, Raritan
Central New Jersey CFUG
www.freecfm.com/c/cjcfug/index.cfm

New York
Albany, NY CFUG
www.anycfug.org

New York
Long Island, NY CFUG
www.licfug.org

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

New York
Rochester, NY CFUG
www.roch-cfug.org

New York
Syracuse, NY CFUG
www.cfugcny.org

Pennsylvania, Harrisburg
Central Penn CFUG
www.centralpenncfug.org

Pennsylvania
Philadelphia, PA CFUG
www.pacfug.org

Pennsylvania
Pittsburgh, PA CFUG
www.orbwave.com/pghcfug

Pennsylvania
State College, PA CFUG
www.cfug-sc.org

Southern

Alabama
Birmingham, AL CFUG
www.bcfug.org

Alabama
Huntsville, AL CFUG
www.nacfug.com

Delaware, Dover
Delaware CFUG
www.decfug.org

Delmarva, Dover
Delmarva CFUG
www.delmarva-cfug.org

Florida
Gainesville, FL CFUG
http://plaza.ufl.edu/aktas

Florida
Orlando, FL CFUG
www.cforlando.com

Florida
Tallahassee, FL CFUG
www.tcfug.com

Florida
Tampa, FL CFUG
www.tbcfug.org

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

Georgia, Atlanta
Atlanta, GA CFUG
www.acfug.org

Georgia, Atlanta
Georgia CFUG
www.cfugorama.com

Georgia
Columbus, GA CFUG
www.vcfug.org

Kentucky
Louisville, KY CFUG
www.loulexcfug.com

Louisiana
New Orleans, LA CFUG
www.nocfug.org

Maryland
Annapolis, MD CFUG
www.ancfug.com

Maryland
Baltimore, MD CFUG
www.cfugorama.com

Maryland
Broadneck H.S. CFUG
www.cfug.broadneck.org

Maryland, Bethesda
Maryland CFUG
www.cfug-md.org

Maryland
California, MD CFUG
http://smdcfug.org

North Carolina
Charlotte, NC CFUG
www.charlotte-cfug.org

North Carolina
Fayetteville, NC CFUG
www.schoollink.net/fcfug

North Carolina
Raleigh, NC CFUG
www.ccfug.org

Oklahoma
Oklahoma City, OK CFUG
http://idgweb4.ouhsc.edu/cfug

Oklahoma
Tulsa, OK MMUG
www.tulsacfug.org

Tennessee
Memphis, TN CFUG
http://cfug.dotlogix.com

Tennessee
Nashville, TN CFUG
www.ncfug.org

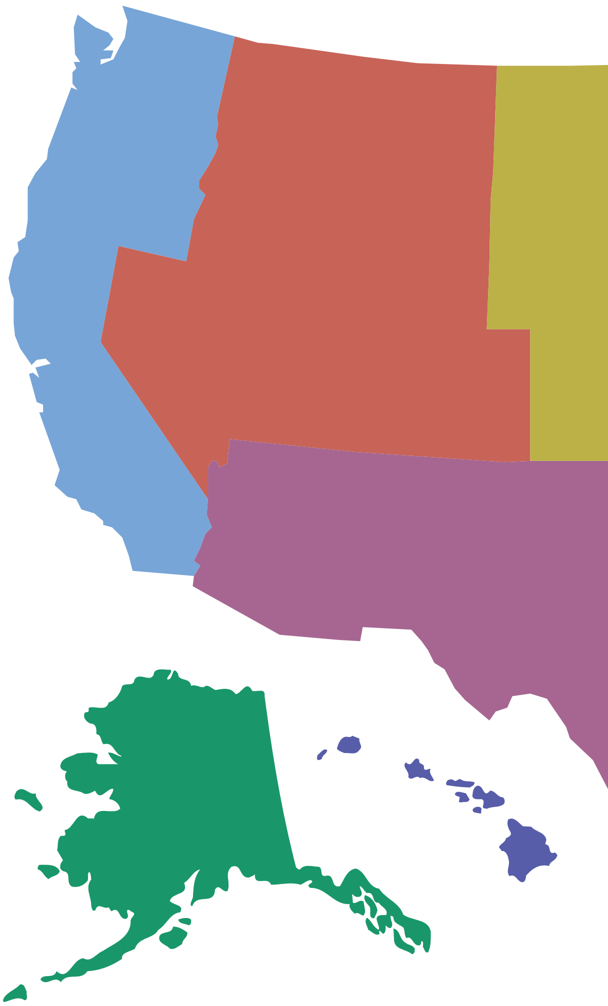
Virginia
Hampton Roads CFUG
www.hrcfug.org

Virginia
Northern Virginia CFUG
www.cfugorama.com

Virginia
Richmond, VA CFUG
http://richmond-cfug.btgi.net

Virginia, Roanoke
Blue Ridge MMUG
www.brmug.com

Washington D.C.
Washington, D.C. CFUG
www.cfugorama.com



Midwest

Northern Colorado
Northern Colorado CFUG
www.nccfug.com

Colorado Hamilton M.S. CFUG
http://hamilton.dpsk12.org/teachers/team-c/intro.html

Illinois, Champaign
East Central Illinois CFUG
www.ecicfug.org

Illinois, Chicago
Chicago, IL CFUG
www.cfugorama.com

Indiana
Indianapolis, IN CFUG
www.hoosierfusion.com

Indiana, South Bend
Northern Indiana CFUG
www.ninmug.org

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

Michigan, Dearborn
Mmaniacs CFUG
www.mmania.org

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

Minnesota, Minneapolis
Twin Cities CFUG
www.colderfusion.com

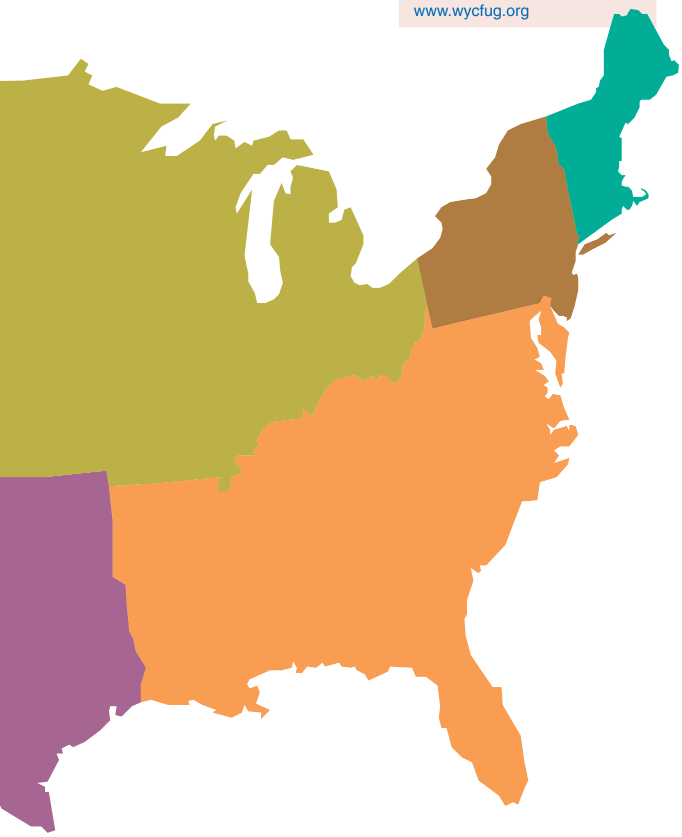
Rockies

Montana, Helena
Montana CFUG
http://montanacfug.site-o-matic.com

Nevada
Las Vegas, NV CFUG
www.sncfug.com

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

Wyoming, Jackson
Wyoming MMUG
www.wycfug.org



Missouri
Kansas City, MO CFUG
www.kcfusion.org

Missouri
St. Louis, MO CFUG
www.psiwebstudio.com/cfug

Nebraska
Omaha, NE CFUG
www.necfug.com

Ohio, Columbus
Ohio Area CFUG
www.oacfug.org

Ohio
Mid Ohio Valley MMUG
www.movcfug.org

Wisconsin
Milwaukee, WI MMUG
www.metromilwaukee.com/usr/cfug

Southwest

Arizona
Phoenix, AZ CFUG
www.azcfug.com

Arizona
Tucson, AZ CFUG
www.tucsoncfug.org

Texas
Austin, TX CFUG
http://cftexas.net

Texas
Dallas, TX CFUG
www.dfwcfug.org

Texas
San Antonio, TX CFUG
http://samcfug.org

West Coast

California
Bay Area CFUG
www.bacfug.net

California, Fresno
Central California CFUG
www.cccfug.com

California, Inland Empire,
Inland Empire CFUG
www.sccfug.org

California
Los Angeles CFUG
www.sccfug.org

California
Orange County CFUG
www.sccfug.org

California
Sacramento, CA CFUG
www.saccfug.org

California
San Diego, CA CFUG
www.sdcfug.org

Southern California
Southern California CFUG
www.sccfug.org

Oregon
Eugene, OR CFUG
www.EugeneCFUG.org

Oregon
Portland, OR CFUG
www.pdxcfug.org

Alaska

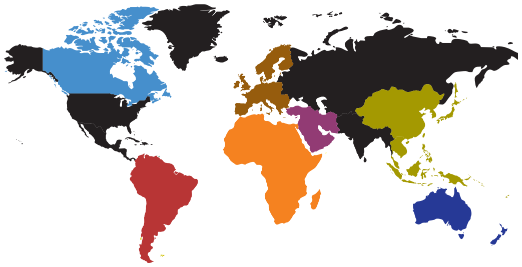
Alaska
Anchorage, AK CFUG
www.akcfug.org

Hawaii

Hawaii, Honolulu
Hawaii CFUG
http://cfhawaii.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.



Africa

South Africa, Cape Town
Cape Town, South Africa CFUG
www.coldfusion.org.za

South Africa, Johannesburg
Johannesburg, South Africa CFUG
www.coldfusion.org.za

Asia

Malaysia, Kuala Lumpur
Malaysia CFUG
www.coldfusioneer.com

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

Japan, Urayasu City Japan CFUG
http://cfusion.itfrontier.co.jp/jcfug/jcfug.cfm

Australia

Australia, Melbourne
Australian CFUGs
www.cfug.org.au

Canada

Canada
Edmonton, AB CFUG
http://edmonton.cfug.ca

Canada
Kingston, ON CFUG
www.kcfug.org

Canada
Montreal, QC CFUG
www.kopanas.com/cfugmontreal

Canada
Ottawa, ON CFUG
www.cfugottawa.com

Canada, Ottawa (HS group)
Ecole Secondary CFUG
www.escgarneau.com/gug

Canada
Toronto, ON CFUG
www.cfugtoronto.org

Canada
Vancouver, BC CFUG
www.cfug-vancouver.com

Canada
Vancouver Island CFUG
www.cfug-vancouverisland.com

Europe

Belgium, Brussels
Belgium CFUG
www.cfug.be

Central Europe, Munich
Central Europe CFUG
www.cfug.de

Finland, Helsinki
Finland CFUG
www.cfug-fi.org

France, Valbonne
France CFUG
www.cfug.fr.st

Germany, Frankfurt
Frankfurt CFUG
www.cfug-frankfurt.de

Ireland
Belfast, Ireland CFUG
www.cfug.ie

Ireland
Cork, Ireland CFUG
http://viewmylist.com/cork

Italy, Bologna
Italy CFUG
www.ingenium-mmug.org

Sweden
Gothenburg, Sweden CFUG
www.cfug-se.org

Switzerland
Martin Bürlimann, Switzerland CFUG
www.cfug.ch

United Kingdom, London
UK CFUG
www.ukcfug.org

United Kingdom
Northern England CFUG
www.cfug.org.uk

Middle East

Pakistan Edu, Lahore Cantt
Pakistan Educational CFUG
www.cfeugpakistan.org

Pakistan, Lahore
Pakistan CFUG
www.cfugpakistan.org

Saudi Arabia, Riyadh
CFUG Saudia
www.cfugsaudia.org

Turkey, Izmer
Turkey CFUG
www.cftr.net

South America

Brazil
Rio de Janeiro CFUG
www.cfugrio.com.br

PaperThin Announces New Nonprofit and Association Customers

(*Boston*) – PaperThin, Inc., a midmarket content management system vendor, has announced that several nonprofit organizations and associations have recently selected CommonSpot Content Server to manage their Internet, intranet, and extranet sites. With CommonSpot, nontechnical staff can now take control of the process of creating and publishing Web content. The timeliness and relevance of content on the Web site is increased while the costs to maintain it are reduced.

New customers in the nonprofit and association sectors include Academy for Education Development (Washington, DC), American Academy of Pediatrics (Elk Grove Village, Illinois), American Society for Engineering Education (Washington, DC), Association of Public Television Stations (Washington, DC), Memphis Arts Council (Memphis, Tennessee), and United Way of the MidSouth (Memphis, Tennessee). These organizations join a long list of associations and nonprofits that have successfully deployed CommonSpot to manage their Web sites.

In developing their new Web initiatives, several of these organizations have engaged PaperThin partners, including Beaconfire Consulting (Washington, DC), Edgewater Technology (Wakefield, Massachusetts), Fig Leaf Software (Washington, DC), Fusion Productions (Webster, New York) and Ringger (Memphis, Tennessee) to redesign and build their new sites, which the organizations will then maintain themselves.

Using CommonSpot's extensive out-of-the-box features, PaperThin partners are able to build dynamic Web sites quickly, according to PaperThin. Deploying standard features like templates and style sheets allows a designer to maintain a consistent look and feel throughout the site. Once the site is built, content owners, rather than in-house Web developers or outside consultants, can easily and cost-effectively create and publish dynamic, personalized Web content within a controlled environment. No special technical knowledge is required, and authors can publish documents, images, PDFs, video clips, and other content using a wizards-based interface.

CommonSpot can be easily integrated with association management systems to provide members with personalized Web site content. "Association staff members like CommonSpot because it is easy to implement and simple for nontechnical content owners to learn, while association executives appreciate CommonSpot's flexible pricing models that fit a wide range of budgets," noted Steve Drucker, CEO of Fig Leaf Software.

www.paperthin.com

Black Knight Transferred to Gestaltech

(*Atlanta*) – Montara Software has reached a definitive agreement to transfer all rights of its Black Knight product – including their customer list – to Gestaltech. Gestaltech will begin marketing an updated version of Black Knight in Q1 of this year.

Black Knight is the first ColdFusion extension that allows developers to leverage Microsoft's .NET platform with their Java-based CFML applications. Black Knight also provides for the porting of existing extensions – including Java extensions through Microsoft J# – frequently requiring only a single line of code to be changed before recompiling.

www.gestaltech.com

Macromedia Director MX 2004 Now Available

(San Francisco) – Macromedia has announced the immediate availability of Macromedia Director MX 2004, the latest version of the industry-standard multimedia authoring tool. Macromedia Director MX 2004 adds support for JavaScript, Macromedia Flash MX 2004 content, DVD-Video playback, and the ability to create projector files for both Mac and Windows platforms in one simple step. Director MX 2004 is available for immediate download from the Macromedia Online Store at www.macromedia.com/go/buy_dmx2004.

"Macromedia Director MX 2004 furthers the product's rich legacy by adding features that directly address developer requests to work more efficiently and utilize the latest technologies available," said Miriam Geller, director of product management, Macromedia. "Director MX 2004 greatly eases development and enables users to create richer content through a seamless workflow with other MX 2004 products and support for standards like JavaScript and DVD-Video."

Macromedia Director MX 2004 delivers multimedia content anywhere, whether it be to CD, DVD, intranets, kiosks, or the Internet. It supports most major video, audio, bitmap, 3D, and vector formats to give developers the broadest content palette from which to deliver sophisticated and compelling user experiences. Extensive video capabilities within Director MX 2004 allow developers to incorporate video files in DVD-Video, Windows Media, RealMedia, QuickTime, and Flash Video formats. DVD-Video functionality will enable developers to embed, control, and play back DVD-Video content within multimedia projects. Director also has a powerful Xtras plug-in architecture that enables developers to extend both the application and playback to add nearly limitless functionality.

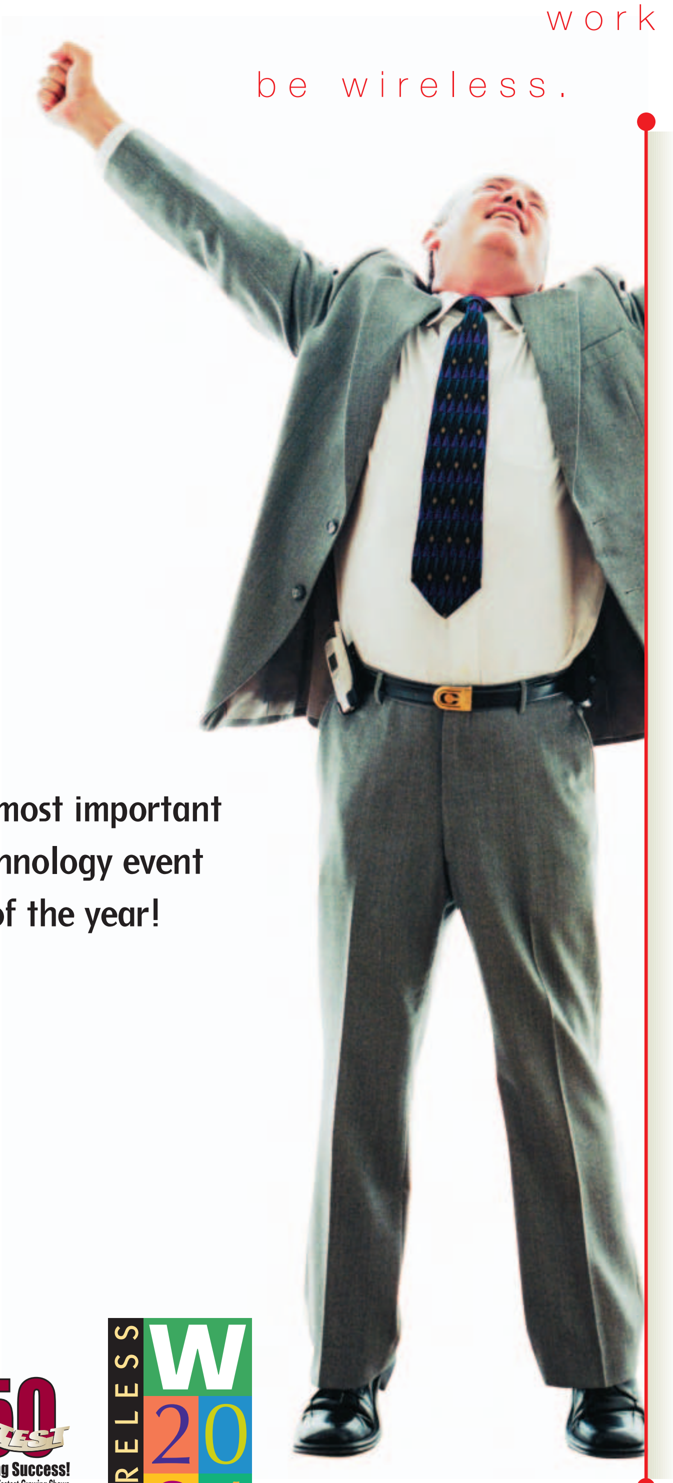
"Every year, I need to create at least eight different presentations that are sent around the world to showcase ESPN programming. Director MX 2004 delivers a whole new level of functionality enhancements that will make my job easier and more fun," said Brett E. Stewart, interactive designer, ESPN, Inc. "Being able to incorporate DVD-Video into my Director presentations will both improve the video quality and more effectively enable me to reach new markets."

Macromedia Director MX 2004 is tightly integrated with other products and servers in the Macromedia MX family. In addition to adding support for Flash MX 2004, Director also has the ability to launch and edit both Flash and Fireworks to enable a seamless workflow. Macromedia will be hosting online seminars to showcase the new features of Director MX 2004. For more information on seminars, visit www.macromedia.com/go/directoronlineseminar.

live wireless.

work wireless.

be wireless.



CTIA WIRELESS 2004

is the one show where wireless standards are created and the technological direction of the industry is set. With the largest gathering of wireless engineers and technologists, this is where you will find the tools you need to help build and advance the wireless industry.

Look at all CTIA WIRELESS 2004 has to offer the wireless engineer and technologist:

- *6 CTIA educational sessions dedicated to exploring wireless technology*
- *IEEE Wireless Communications Network Conference (WCNC) 2004 – the industry's foremost conference for developing wireless standards and engineering*
- *WiFi Summit – the CTIA Smart Pass program, a cutting edge look at WiFi strategy and security*
- *A 400,000 square foot exhibit floor displaying the latest in wireless technology and applications*

The most important technology event of the year!

CTIA Global

TSW 50 FASTEST
Celebrating Success!
Tradeshow Week's 50 Fastest-Growing Shows

Produced by
CTIA

WIRELESS 2004 CTIA®

welcome the wireless generation.

March 22-24, 2004 Georgia World Congress Center Atlanta, GA, USA www.ctiashow.com

John T. Chambers
President & CEO
Cisco Systems

Scott McNealy
Chairman & CEO
Sun Microsystems, Inc.

Russell Simmons
Chairman & CEO, Rush Communications,
Co-founder & Chairman, Def Jam Records