INTEGRATING FLEX APPS WITH COLDFUSION PG 5

COLDFUSION Developer's Journal

ColdFusionJournal.com

April 2006 Volume:8 Issue:4

Presorted
Standard
US Postage
PAID
St. Croix Press

ColdFusion Hosting is our Complete Focus

POWERFUL HOSTING PLANS
FREE SQL server access | FREE account setup | Unlimited email accounts Generous disk space & data transfer
30 day money-back guarantee | Great value

RELIABLE NETWORK
99.99% average uptime | State-of-the-art data center with complete redundancy in power, HVAC, fire suppression, bandwidth and security | 24/7 network monitoring

FANTASTIC SUPPORT SERVICES
24/7 suppport services | Knowledgeable phone support | We focus on your individual needs

CFDynamics

866.233.9626 CFDYNAMICS.COM

For years we have been involved in the Cold Fusion community and have come to know what developers and project managers look for in a web host. The combination of our powerful hosting plans, reliable network, and fantastic suppport sets us apart from other hosts.
Real service. Real satisfaction. Real value. Real support. Real Freedom.

macromedia ALLIANCE PARTNER

BlueDragonCertified V.I.P. Hosting Partner macromedia®

One little box. A whole lot of power.

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

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

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

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

› Make rapid J2EE development a reality.

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

› New mobile applications are a go.

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

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

For answers call us at 1-866-EDGEWEB
3 3 4 3 9 3 2

When calling your web host for support you want answers, not an annoying song stuck in your head from spending all day on hold. At Edgewebhosting.net, we'll answer your call in two rings or less. There's no annoying on-hold music, no recorded messages or confusing menu merry-go-rounds. And when you call, one of our qualified experts will have the answers you're looking for. Not that you'll need to call us often since our self-healing servers virtually eliminate the potential for problems and automatically resolve most CF, ASP, .NET, SQL, IIS and Linux problems in 60 seconds or less with no human interaction.

Sleep soundly, take a vacation, and be confident knowing your server will be housed in one of the most redundant self-owned datacenters in the world alongside some of the largest sites on the Internet today and kept online and operational by one of the most advanced teams of skilled Gurus, DBAs, Network and Systems Engineers.

By the Numbers:

- 2 Rings or less, live support
- 100 % Guarantee
- 99.999% Uptime
- 2.6 GBPS Redundant Internet Fiber Connectivity
- 1st Tier Carrier Neutral Facility
- 24 x 7 Emergency support
- 24 Hour Backup
- Type IV Redundant Datacenter

EDGE WEB HOSTING

For a new kind of easy listening,
talk to EdgeWebHosting.net

http://edgewebhosting.net

CFDJ Readers' Choice Award · MXDJ Readers' Choice Award

2003 - 2006

- Shared Hosting
- Managed Dedicated Servers
- Managed Colocation
- Semi-Private Servers
- ColdFusion
- BlueDragon
- ASP
- .NET
- .Linux
- .Java
- SQL Server
- .MySQL
- Self-Healing Servers

editorial

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

technical editor
Raymond Camden raymond@sys-con.com

executive editor
Nancy Valentine nancy@sys-con.com

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

production

lead designer
Abraham Addo abraham@sys-con.com

art director
Alex Botero alex@sys-con.com

associate art directors
Louis F. Cuffari louis@sys-con.com
Tami Beatty tami@sys-con.com

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

What I Love About ColdFusion

By Simon Horwith

Last month *CFDJ* held a contest in which I asked readers to write an essay about how ColdFusion MX 7 has made them a hero in the office.

You can read the winning entries and find out about our next contest in this month's issue. Deciding on the contest topic and reading the contest submissions made me reflect on my own experiences with CF over the past decade. What is it about ColdFusion that's kept me hooked for so long?

When ColdFusion was first released there was one main feature, more than any other feature, that made it very popular very fast: ColdFusion makes it ridiculously easy to create Web pages that can talk to databases and display database information. Now, almost 11 years later, this is still one of ColdFusion's primary strengths and most popular features.

Making it easy to talk to databases may be useful and popular, but it is certainly not enough to stay on top for over 10 years. A rapidly growing development community, global technological innovations, customer demand, and competing products require that new features are added to the ColdFusion server every so often. Thinking back over the years I remember the introduction of what seemed like several hundred functions and the ability to write simpler expressions, loop, etc., when CF 2 was released. I remember how great it was to have custom tags in ColdFusion 3. In versions 4, 4.01, and 4.5 we saw the addition of many more functions, server security features, and tags aimed at making the existing functionality more robust as well as many tags for leveraging external resources. It was at this point that the server clearly began addressing enterprise needs. Version 5 introduced charting, user-defined functions, the ability to access most memory scopes as structures, and Query of Queries…and then we had MX.

I remember so clearly just how overwhelming ColdFusion MX was and how much fun I had with it. SOAP support, XML support, J2EE deployment, and ColdFusion Components to name a few – this was not only a rewrite from the ground up, it was a serious feature release in its own right. Then, just over one year ago, ColdFusion MX 7 was officially released. I thought nothing could outdo the features introduced in ColdFusion MX – I was wrong. Flash Forms and Flash Paper, PDF generation, a killer new reporting engine and report builder, the Administrator API, and Event Gateways…dare I not forget event gateways…this was by far the most feature-packed release of ColdFusion to date.

What has made ColdFusion successful over all these years is the commitment that both Allaire and Macromedia had to making CF full of more features that are easier to use than anything else on the Web. I am confident that Adobe shares this commitment and cannot wait to see what comes next. In the interim before ColdFusion 8, we are getting a sneak peek at how Adobe is making ColdFusion better over at Adobe Labs, where the "Mystic" beta is adding a new remoting gateway and Flex/CF integration to the server.

Getting back to my original thought: What has kept me hooked on ColdFusion for so

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

BALANCE

Designer/developer, front-end/back-end, clients/sanity. . .web development is a balance and we can help you maintain it. Join now and experience a wealth of training resources tailored to the tools you use every day.

www.communitymx.com

Visit www.communitymx.com/trial/ for your free 10 day trial.

COMMUNITY MX

SYS-CON MEDIA

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

group publisher
Jeremy Geelan jeremy@sys-con.com

advertising

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

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

advertising director
Robyn Forma robyn@sys-con.com

advertising manager
Megan Mussa megan@sys-con.com

associate sales manager
Kerry Mealia kerry@sys-con.com
Lauren Orsi lauren@sys-con.com

sys-con events

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

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

customer relations

circulation service coordinator
Edna Earle Russell edna@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
Wayne Uffelman wayne@sys-con.com

online editor
Roger Strukhoff roger@sys-con.com

accounting

financial analyst
Joan LaRose joan@sys-con.com

accounts payable
Betty White betty@sys-con.com

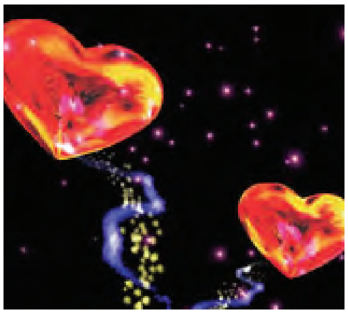
accounts receivable
Gail Naples gailn@sys-con.com

subscriptions

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

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

long? I can answer this with a single word – fun. It is fun building applications with ColdFusion. It hides complexity by making trivial that which it can. This leaves me to concentrate on the things that I want to focus on, all the while letting the server do its "dirty work" behind the scenes. I can develop extremely robust complex business logic – and in an object-oriented manner. I can talk to over a dozen different types of databases. I can deliver content in any one of several slick document formats in addition to HTML. I can create charts and graphs and Flash Paper and PDFs, and I can do intelligent text searching not only of database content but of file content in over 30 different document formats as well. I can talk to other servers and let other servers leverage the ColdFusion code that I write via SOAP. I can create files, generate reports, parse XML, and secure directories. I can talk to cell phones and instant messengers, Java devices and server ports. In other words, there isn't much I can't do. And I can do it all much faster than developers in any other language or on any other platform. My applications are secure, they scale, they are feature rich, and I have fun building them. I can look any client in the eye and honestly tell them that I will exceed their every expectation. Yes, it will be on time and under budget, and the application will be modern and full of great features that take advantage of the latest technologies.



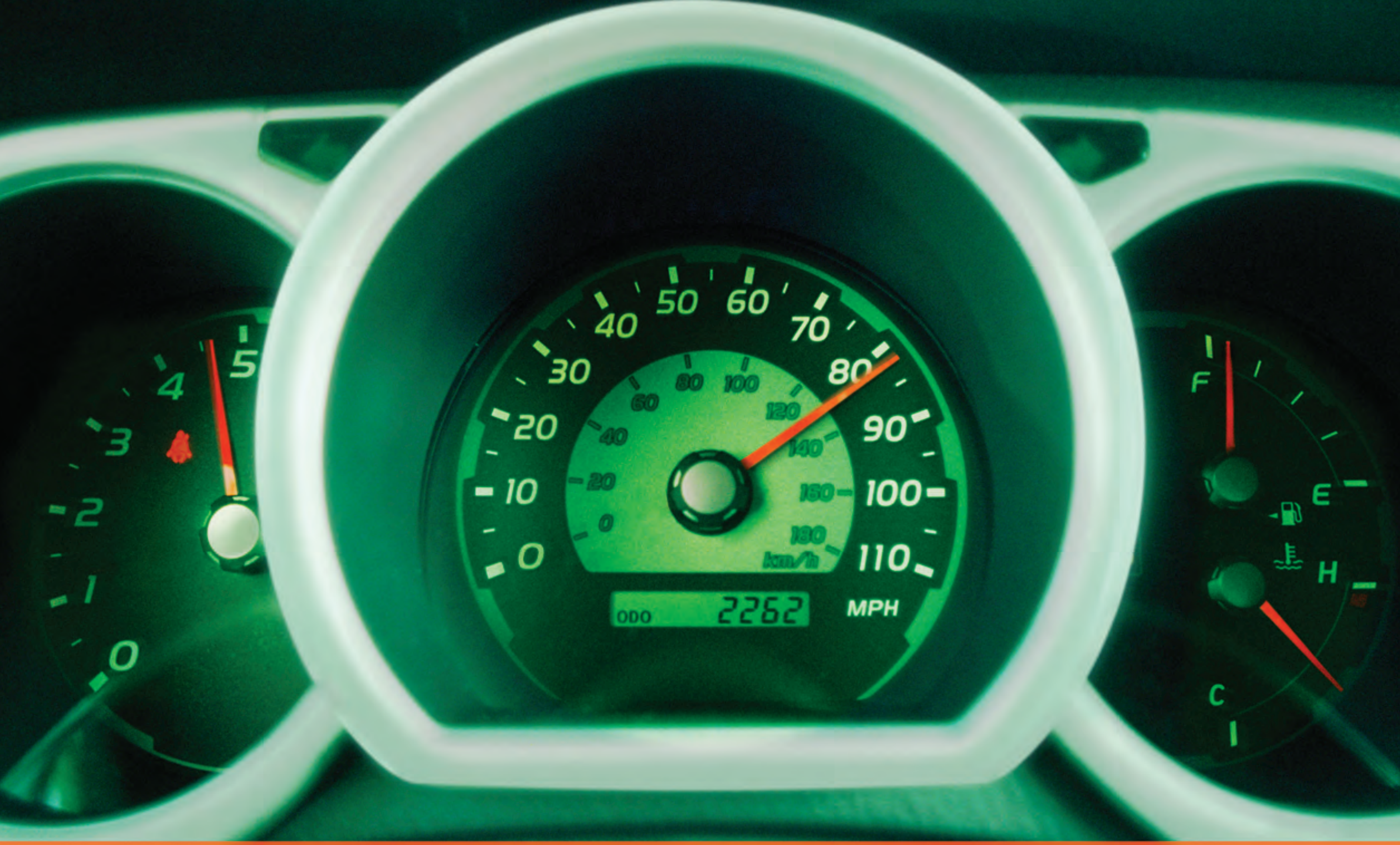
What's so great about ColdFusion is that none of the above is difficult to do, none of it. CF can do all of that and so much more right out of the box, often times with a single call to a tag. In this day and age in which technology is rapidly changing and companies value the ability to quickly change with it, it's a breath of fresh air to work with such an agile product.

For me, the fact that ColdFusion makes many tasks trivial is not just about rapid development, it's about resource allocation. Because I don't have to spend 10 hours figuring out how to create a PDF document from an HTML string, maybe I can spend more time learning Java, or studying design patterns, or learning Flex, or doing anything else I choose. For me, ColdFusion's ease of use enables me to spend more time evaluating and mastering other technologies and industry trends, and allows me to spend more time concentrating on those aspects of CFML development that I prefer to focus on. It's easy to become complacent with one's CF skills though, and every time a new feature-rich version of ColdFusion is released, it's important to stop everything and take time to master those new features. I wonder how many CF developers, even those who claim to really know CF 7, have actually really learned the ins and outs of the report generator, or XML forms, or using the Asynchronous Gateway? There are roughly 50 new features in ColdFusion MX 7 – so very few of us have actually mastered them all. My advice to all of our readers is a bit of advice I frequently give newbie students.

There's a lot in CFML and you can't expect to learn it all overnight. Pick one new tag, one new feature, or one new category of functions each week from the CF documentation and spend an hour or two each day mastering it. You'll be shocked how much you learn in a reasonably short amount of time. In the case of ColdFusion MX 7, even if you were completely fluent in CF MX 6.1 prior to its release, I expect that if you follow my advice you'll find that it'll take between six months and a year to learn everything that's new or changed in 7.

One feature introduced in ColdFusion MX and improved in CFMX 7 that I didn't mention earlier is the support for Flash Remoting. The significance of this feature wasn't fully realized by most developers initially, but Flex 2 is poised to change that. In extremely simple terms, Flex 2 is aimed at doing for Flash programming what ColdFusion has done for traditional server-side Web development – make it faster and easier to build better applications. So far it's living up to this promise, and I have no doubt that soon we'll see many CF applications getting "UI facelifts" and delivering better experiences than ever before. In this age of smart clients, Flex offers compelling advantages over all of its competitors, and Adobe has made sure that integrating Flex applications with ColdFusion is seamless, fast, and easy. I've said this before in recent editorials and I'll say it once more – there has never been a better time to be a ColdFusion developer.

TABLE OF CONTENTS



Rev Up Your Flash Video

Stay Ahead of the Competition With VitalStream and the Enhanced Video Features in Flash 8

With over two years of experience in delivering much of today's most popular media, VitalStream® is the first and most experienced Flash™ video streaming service provider.

Enhanced Flash 8 Video Features:

- New VP6 codec delivers higher quality video at the same bit rate
- 8-bit alpha channel transparency enables you to blend video with other elements
- Improved live video capabilities

VitalStream Complete Toolset for Flash:

- MediaConsole®
- MediaOps™ SDK
- Flash Authentication
- Reporting Dashboard

Integrate Streaming Media Into Your Flash Projects

Take Advantage of the Enhanced Video Features in Macromedia Flash 8
Call (800) 254-7554 or Download Tutorials at www.vitalstream.com/go/mxdj

macromedia® FLASH VIDEO STREAMING SERVICE

Call (800) 254-7554
Visit www.vitalstream.com

VitalStream®

Isn't It About Time to Dump CF5?

Going from CF5 to CFMX7

By Jeffry Houser

A new project just landed on my desk. This is not a maintenance project; it's brand new, start from scratch, development. The client asked me to build a site in ColdFusion 5. It's my fourth project in the past year that's based on a legacy version of CF. I could have saved the client both time and money by building it in CFMX, since I have a lot of the functionality pre-built as ColdFusion Components. But CFCs don't work in CF5, so those pre-built components are pretty useless. There's no better time to upgrade than right now; in this article I thought I'd show you the process I'd take when upgrading a server from CF5 to CFMX7.

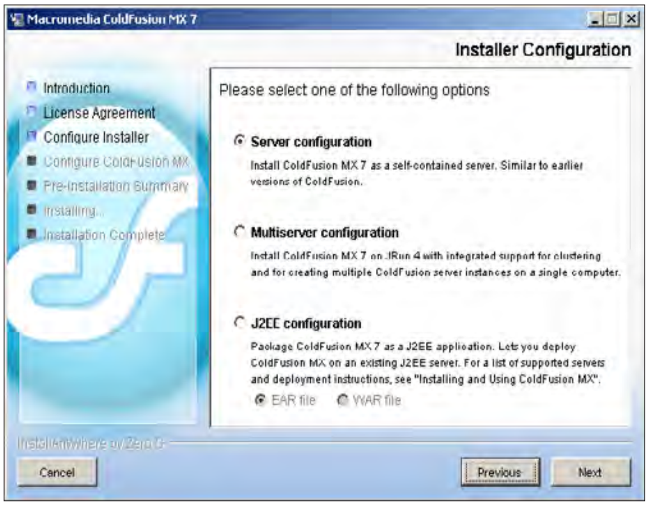
The Upgrade

There are compelling reasons to get off CF5. For developers, it's the use of CFCs that let you build more maintainable code. For business folk, it's support. It's harder to find CF5-based developers and I suspect that when CFMX8 comes out, support for CF5 will drop. If they follow an 18-month release schedule, then I'm sure it will happen within the next year. Other niceties like PDF generation and event gateways make the upgrade more appealing. So let me step you through the process.

First download the latest version of ColdFusion. You can do so from the Adobe Web site. The J2EE version is a slightly different beast so I'll just step you through the standard install. It's closest to the CF5 installation. The J2EE version, if you already have a J2EE server, can create multiple isolated ColdFusion instances. This is great if you have to isolate apps from each other, but I don't usually use this. You can get the same functionality from the multi-server install (and it's easier to manage).

Launch the install program, and follow these steps:

1. Select your language and click Okay. Read the introduction note. There are no decisions to make on this screen so click Next. Read through the license agreement and select "I agree" then click Next. These were the easy steps.
2. Enter your license number or select the trial/developer edition. For this example, the developer edition is okay. Select it, and click Next.
3. This screen in Figure 1 lets you select which version to install. I've already mentioned versions, but you can read more details about them in the livedocs at http://livedocs.macromedia.com/coldfusion/7/htmldocs/00000003.htm#1211441. Select "server" and click Next.
4. This screen lets you select the sub-components you want to install. There are three options, the documentation, the search services, and the ODBC services. I won't install the tutorials or documentation on a production server, but I usually keep the other two. On a dev server I usually install it all. Click Next.
5. Select the directory that you want to install to. I usually keep the default. Click Next.
6. A previous version was detected, as shown in Figure 2. You can use the built in the Web server or reconfigure the existing Web server. We want to reconfigure. Click Next. The installer will detect your Web server. You're given the option of configuring multiple Web sites. On Windows XP, IIS doesn't let you have multiple Web sites, so select Update All and click Next. Enter your Web root and click Next.
7. Enter the password for the ColdFusion Administrator. Click Next and enter the password for RDS. Click Next.
8. You'll get a summary of the install options, as shown in



Please select one of the following options

[✓ Server configuration]
Install ColdFusion MX 7 as a self-contained server. Similar to earlier versions of ColdFusion.

Multiserver configuration
Install ColdFusion MX 7 on JRun 4 with integrated support for clustering and for creating multiple ColdFusion server instances on a single computer.

J2EE configuration
Package ColdFusion MX 7 as a J2EE application. Lets you deploy ColdFusion MX on an existing J2EE server. For a list of supported servers and deployment instructions, see "Installing and Using ColdFusion MX".
[✓ EAR file] WAR file

Figure 1 Select the install version

WEB ALL-STAR seeks an Integrated Software Suite that's up for anything. Understand my need to create cool graphics one day, while tweaking CSS the next. Should be intuitive, multi-platform and not phased by my turbo-charged pace. High-maintenance is a no-no.

Different people. Different needs. One suite solution.

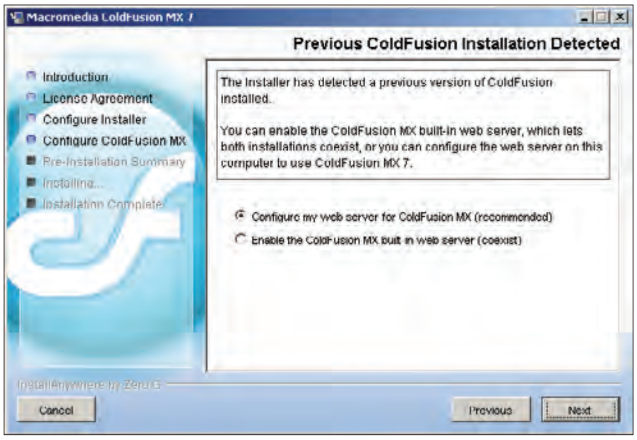
With the latest versions of Macromedia Dreamweaver®, Flash® Professional, Fireworks®, Contribute™, and FlashPaper™, the new Studio 8 is quite a catch. To meet Studio 8 and find all the tools you need to design, develop and maintain online experiences, visit www.macromedia.com/go/studio8_mxdj

macromedia®
STUDIO 8

Figure 3. Review them and click Install. ColdFusion will start installing. Go top off your drink of the day (coffee, water, or vodka).

9. When you come back, CF should be installed.

Now that ColdFusion is installed, you have to configure it. This happens the first time you open the ColdFusion Admin-


Macromedia ColdFusion MX 7

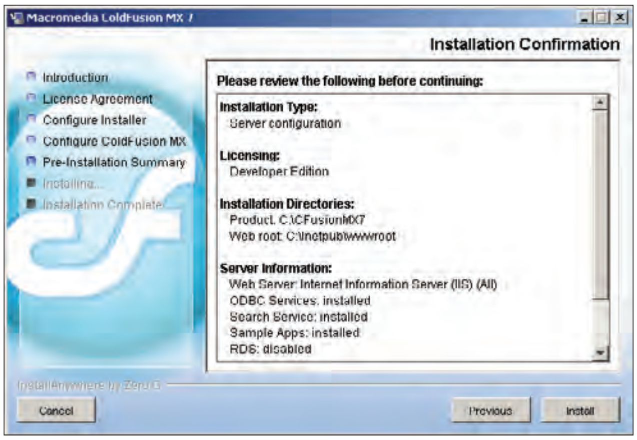
Previous ColdFusion Installation Detected

The Installer has detected a previous version of ColdFusion installed.

You can enable the ColdFusion MX built-in web server, which lets both installations coexist, or you can configure the web server on this computer to use ColdFusion MX 7.

(•) Configure my web server for ColdFusion MX (recommended)
( ) Enable the ColdFusion MX built in web server (coexist)

Figure 2: Previous ColdFusion installation detected


Macromedia ColdFusion MX 7

Installation Confirmation

Please review the following before continuing:

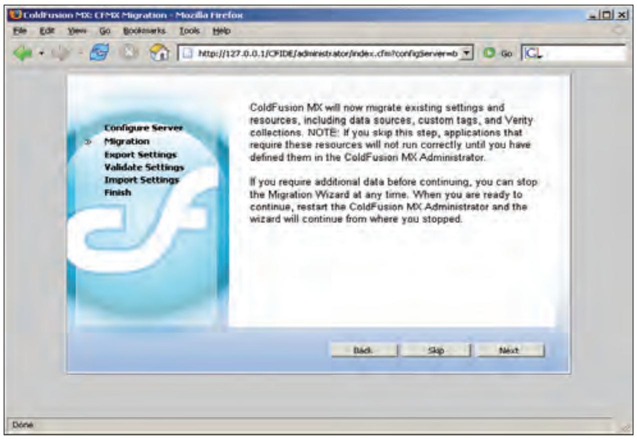
Installation Type:
Server configuration

Licensing:
Developer Edition

Installation Directories:
Product: C:\CFusionMX7
Web root: C:\Inetpub\wwwroot

Server Information:
Web Server: Internet Information Server (IIS) (All)
ODBC Services: installed
Search Service: installed
Sample Apps: installed
RDS: disabled

Figure 3: Summary of options

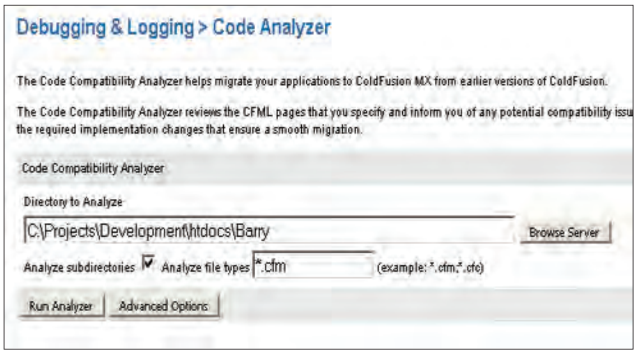

ColdFusion MX: CFMX Migration - Mozilla Firefox

http://127.0.0.1/CFIDE/administrator/index.cfm?configServer=b

ColdFusion MX will now migrate existing settings and resources, including data sources, custom tags, and Verity collections. NOTE: If you skip this step, applications that require these resources will not run correctly until you have defined them in the ColdFusion MX Administrator.

If you require additional data before continuing, you can stop the Migration Wizard at any time. When you are ready to continue, restart the ColdFusion MX Administrator and the wizard will continue from where you stopped.

Figure 4 The ColdFusion configuration wizard.


Debugging & Logging > Code Analyzer

The Code Compatibility Analyzer helps migrate your applications to ColdFusion MX from earlier versions of ColdFusion.

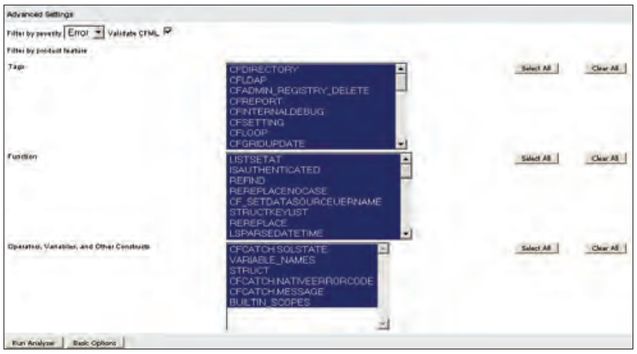
The Code Compatibility Analyzer reviews the CFML pages that you specify and inform you of any potential compatibility issu the required implementation changes that ensure a smooth migration.

Code Compatibility Analyzer

Directory to Analyze: C:\Projects\Development\htdocs\Barry

Analyze subdirectories ☑ Analyze file types *.cfm (example: *.cfm;*.cfc)

Figure 5 Compatibility analyzer


Advanced Settings

Filter by severity: Error Validate CFML ☑

Filter by product feature

Tags: CFDIRECTORY, CFLDAP, CFADMIN_REGISTRY_DELETE, CFREPORT, CFINTERNALDEBUG, CFSETTING, CFLOOP, CFGRIDUPDATE

Function: LISTSETAT, ISAUTHENTICATED, REFIND, REREPLACENOCASE, CF_SETDATASOURCEUERNAME, STRUCTKEYLIST, REREPLACE, LSPARSEDATETIME

Operators, Variables, and Other Constructs: CFCATCH.SQLSTATE, VARIABLE_NAMES, STRUCT, CFCATCH.NATIVEERRORCODE, CFCATCH.MESSAGE, BUILTIN_SCOPES

Figure 6 Compatibility analyzer advanced options

istrator, or you could select the checkbox on the final install screen to do the configuration manually. Click Done (and check the box) to enter the configuration wizard, as shown in Figure 4.

1. Enter your ColdFusion Administrator password and click Login.
2. ColdFusion will chug for a bit and then give you the option of migrating your settings. You probably want to do this, so click Next. If you want to start CFMX new and configure it as needed you can skip this.
3. You'll see a list of items that can be migrated. Datasources, verity collections, and mappings are just some of them. Click Next.
4. ColdFusion will chug some more. It will let you know if there are any problems. Click Next to start the migration.
5. Once again, the setup wizard chugs and lets you know that things were done. Click Ok to open the Getting Started experience (if you installed he samples / documentation).

The Code Analyzer

Before upgrading a production machine, you might want to test your apps on a development machine first. I strongly recommend it, especially when moving from ColdFusion 5 to ColdFusion MX. ColdFusion 5 was built on a C++ architecture, whereas CFMX is on top of J2EE. These changes in underlying structure introduced a few oddities to the language. Most of your code will work without a problem, and there are many upgrade success stories. But, for those few gotchas it's good to test.

To help alleviate the migration problem, the ColdFusion administrator includes a code analyzer. It will examine your code and help isolate problems or potential problems. (The code analyzer also helps if you're just upgrading from CFMX to CFMX7, although you're less likely to have serious issues). Once ColdFusion's installed, launch the administrator and sign in. Select Code Analyzer under the "Debugging and Logging" menu. You'll see a screen like Figure 5.

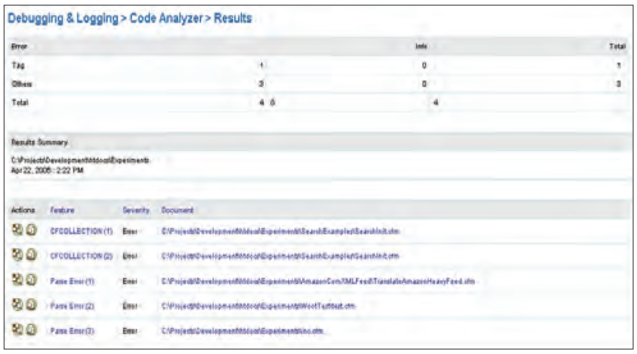
Enter the name of your directory, the type of files to analyze, and click the run analyzer option. For the file types, most likely you'll be analyzing cfm files and therefore want to use the default "*.cfm" as the type. If you've used some tricks on the Web server to make ColdFusion process files with alternate extensions such as HTML then you'll want to add those extensions here. Click the advanced options feature, and you'll see Figure 6. This lets you specify which tags, functions, or issues to look for in your code. Normally, I'd just use the basic option, but if you need this kind of granular control it's there.

Click Run Analyzer and you should a list similar to Figure 7. (Or maybe you code perfectly and get nothing returned.) You can sort the listing by clicking on the column headers. Click the first icon to see a detailed report of the errors. Click on the second to see a file specific.

I don't recommend using the code analyzer as a substitute for doing quality assurance tests on your apps, but it's a good starting point to figure out what areas may need code changes.

Here are some of the most common issues I've faced when upgrading:

- ***Security Model:*** The security model changed in moving off the C code base. Many tags and functions related to the CF5 way of doing things have been rendered obsolete. Tags such as cfauthenticate and cfimpersonate are gone. Functions such as isAuthorized, isAuthenticated, and AuthenticatedUser are gone. Take a look at cflogin and cfloginuser for information on ColdFusion's new security features. (You didn't use that old security framework anyway, did you?)
- ***Monitor Errors:*** CF5 had an error type for the cferror called monitor. Monitor was so you could follow specific error types. It trumped cftry / cfcatch blocks. This kind of error handling is gone.
- ***Graphing:*** cfgraph and cfgraphdata are gone. Instead look at cfchart, cfhartdata, and cfhartseries.
- ***Dynamic Database Access:*** ColdFusion 5 lets you use connections strings to connect to the database. This was useful in many cases, especially shared hosting environments and preventing other people on your server from guessing your DSN and getting access to all of your data. Connection strings were added to BlueDragon.net, but with ColdFusion the best you're going to get is a security sandbox that restricts database access.
- ***Registry Access:*** The cfregistry tag no longer works on Unix machines. Unix doesn't have a registry in the same way that ColdFusion does; so that's not a surprise.
- ***Periods in the Variable Name:*** In CF5, the period was valid in a variable name. In CFMX, a structure will be created. The text before the period will be the structure name and the text after the period will be the key. This is called dot notation and is similar to how you access many values in CF, such the results of a query set.
- ***ODBC:*** This is more of a configuration issue than a coding issue. In CF5, ColdFusion could access datasources created in the ODBC control panel by default. Since CFMX is Java-based, it uses JDBC instead of ODBC. You can access control panel databases uses an ODBC bridge, or connect to the database directly using the JDBC drivers included.



Debugging & Logging > Code Analyzer > Results

Error		Info	Total
Tag	1	0	1
Others	3	0	3
Total	4 6	4	

Results Summary

C:\Projects\Development\Hdocs\Experiments
Apr 22, 2006 - 2:22 PM

Actions	Feature	Severity	Document
	CFCOLLECTION (1)	Error	C:\Projects\Development\Hdocs\Experiments\SearchExamples\SearchInit.cfm
	CFCOLLECTION (2)	Error	C:\Projects\Development\Hdocs\Experiments\SearchExamples\SearchInit.cfm
	Parse Error (1)	Error	C:\Projects\Development\Hdocs\Experiments\AmazonCom\XMLFeed\TranslateAmazonHeavyFeed.cfm
	Parse Error (2)	Error	C:\Projects\Development\Hdocs\Experiments\WoolTest.cfm
	Parse Error (3)	Error	C:\Projects\Development\Hdocs\Experiments\inc.cfm

Figure 7 Compatibility analyzer results

There's some documentation included with CFMX6 that talks about migrating from 5 to MX. It's worth a read since it goes into more depth than this article. Check it out at http://livedocs.macromedia.com/coldfusion/6/Migrating_ColdFusion_5_Applications/cf_migration_guide.htm.

Final Thoughts

It seems weird to be writing about this so long after the initial release of CFMX. I look forward to the day when I can remove CF5 from my machine and clients will stop asking me for version 5-compatible code (I've started turning down requests for 4.5 development). Until then, I keep ColdFusion 5, ColdFusion MX, and BlueDragon running side-by-side. This is beneficial when it comes to migrating apps between versions or testing code for cross-version compatibility. I described this set up in an article I co-wrote with Charlie Arehart at http://coldfusion.sys-con.com/read/42069.htm. Although it's a bit dated, all the techniques still apply.

About the Author

Jeff Houser has been working with computers for over 20 years. He owns a DotComIt, a web consulting company, manages the CT Macromedia User Group, and routinely speaks and writes about development issues. You can find out what he's up to by checking his Blog at www.jeffryhouser.com.

jeff@instantcoldfusion.com

CFDJ Essay Contest Results

It's show time!

By Simon Horwith

Last month I announced a *CFDJ* essay writing contest; the topic for the essays was "How ColdFusion MX 7 made me a hero in the office." I received many great entries and thank each of you for your submissions.

I chose two winning entries, and the authors of each will receive free entry to CFUnited, the largest ColdFusion event of the year. The winning submissions were not graded on writing style or grammar, but on the merits of their content. I'd like to thank Michael Smith and Liz Frederick from TeraTech for working with me on this and for donating these great prizes.

Our first winner, Mike Markowski, is a Macromedia/Adobe Certified Professional who works for the Air Protection Division at the Environmental Protection Agency. Mike is relatively new to ColdFusion, having built his first application in 2004. His essay not only describes how ColdFusion and many of the features introduced in ColdFusion MX 7 made him a hero, but also illustrates ColdFusion's ability to allow relatively inexperienced developers to quickly achieve results.

Our second winner is John Baldwin. John is the Director of Information Technology at Riley Children's Foundation, a non-profit committed to improving the health and well-being of the children of Indiana. In his essay, John describes how ColdFusion has helped to automate and increase the processing of online transactions (donations) and its reporting and PDF generation has been essential for generating donor statement letters as Acrobat PDF documents.

Both of these wining entries have been reprinted in *CFDJ* this month. They are a good sample of the wide spectrum of entities using CF, from large government organizations to smaller charitable associations. Everywhere, in every size business and in every industry, more and more companies are relying on ColdFusion to drive mission-critical applications and to take the needs and data within organizations onto the Web.

There's good news for those of you who had high hopes of winning a free entry to CFUnited but didn't win our contest this month. CFDJ is holding one more essay contest to win free admission to CFUnited. I apologize in advance as I do realize that this issue may not reach many of our print readers until just prior to the actual conference start date. News of this contest certainly should reach those of you who read *CFDJ* online (http://www.coldfusionjournal.com/) and/or who follow my blog (http://www.horwith.com).

Our next contest is quite simple. It's another e-mail/essay contest and one that pretty much all of our readers should be qualified to enter. Write me an essay, or even just an e-mail (simon@horwith.com), describing why you'd like me to send you to CFUnited. That's all. You don't have to write a novel – even something as short as a single paragraph will do if you feel that you can explain enough in one paragraph. I'm not judging entries based on length or grammar. The winning entry (or entries) will be selected based, once again, on the merit of their content. The author of the winning entry for this contest will receive free admission to CFUnited, but he or she will also be assigned some homework. I expect this essay contest winner to write me an article, immediately following the close of CFUnited, describing their experience and thoughts about the event. I'll be reviewing essays and e-mails for this contest until May 22, at which point I'll make a decision about the winner. In addition, I am also extending last month's topic until the 22 for anybody who would still rather write about how ColdFusion MX 7 made them a hero.

Good luck.

Enter the premier ColdFusion Conference

Charlie Arehart
New Atlanta

John Paul Ashenfelter
Transitionpoint

Selene Bainum
INPUT

Simeon Bateman
Simb Development

Thomas Burleson
Universal Mind

Raymond Camden
Mindseye

Sandy Clark
Constella Group

Sean Corfield
Adobe Systems, Inc.

Michael Dinowitz
House of Fusion

Nahuel Foronda
Universal Mind

Ben Forta
Adobe Systems, Inc.

Shlomy Gantz
BlueBrick Inc.

Rob Gonda
Editor-in-Chief of AJAX Developer's Journal

Hal Helms
Hal Helms Inc.

Simon Horwith
Editor-in-Chief, CF Developer's Journal

Jeffry Houser
DotComIt

Jeremy Kadlec
Edgewood Solutions

Kurtis D Leatham
KomputerMan.com

Adam Wayne Lehman
US Department of State

Tom Link
Universal Mind

Jeremy Lund
Univ. Health Care

Nate Nelson
Xententia, Inc.

Mike Nimer
Adobe Systems Inc.

Maxim Porges
CFI/Westgate Resorts

Steve Rittler
Countermarch Systems

Andrew Schwab
IEXP Software / FusionDox

Michael Smith
TeraTech

Matt Stevanus
Universal Mind

Jeff Tapper
Tapper.net Consulting

Kelly Tetterton
Duo Consulting

Glenda Vigoreaux
GVX Technology

Douglas Ward
The Centech Group

Dave Watts
Figleaf Software

Register Today!

CFUNITED is the only conference of its kind that is run by developers, for developers. What this means is that speakers aren't dictated what they can and cannot talk about and all of the speakers are encouraged to base their sessions on real world experience. There are real users in the case studies so you can see what your colleagues at other organizations are doing. It also means that the marketing fluff stops at the door – so you can learn more from other developers!

www.cfunited.com

"I attended CFUNITED along with a number of my colleagues from the university. Some were experienced developers, some were novices, some were IT managers or server admins. The consistent thread of all our post-conference conversations regarding CFUNITED was the number of ideas we came away with. Whether it was a new solution to an old problem or just having the possibilities of ColdFusion opened up before us, we all felt it was well worth the trip. Of course our blood sugar levels after eating that CF birthday cake might have contributed some to that feeling as well :-)"
- Bob Flynn, Indiana University MMUG

Accessibility/Usability - Section 508, CSS and disabled access
Advanced CF - Advanced ColdFusion topics
Boot Camp - Basic ColdFusion and Flash Topics
Deployment/Platform - Integrating with SQL Server, Windows and .NET
Flex - Learn Flex 2 by experts at Universal Mind & Adobe
Manager/Emp Programming - Fusebox and Project management topics

8th Annual Event
Over 1,000 attendees

Pre-Conference Classes: June 26th & 27th
Main Event: June 28th - July 1st
Washington DC Area

Adobe
New Atlanta COMMUNICATIONS
TeraTech Programming
HostMySite.com
Microsoft

Implementing HTTP Basic Authentication

Route around many of the common limitations of traditional forms-based authentication

By Patrick Correia

Most ColdFusion applications that require users to be authenticated follow the pattern laid out in the official ColdFusion documentation and in the ColdFusion MX Developer's Guide.

This HTML forms-based method is perfectly serviceable, but as applications become more complicated it's not uncommon to run across issues with session expiration and deep linking/bookmarking. This article describes an alternative method of implementing user authentication that's based on the mature and time-tested authentication scheme laid out in the original HTTP specification. Without making the application significantly more complex, it can route around many of the most common limitations of traditional forms-based authentication.

An Overview of Forms-Based Authentication

To start off, let's take a look at some typical ColdFusion code that uses forms-based authentication from the point-of-view of a Web browser. A typical application might have code in the Application.cfm file that looks something like this:

```
<cfif NOT IsDefined("SESSION.login")>
    <cfif NOT IsDefined("FORM.login")>
            <cfinclude template="loginForm.cfm">
            <cfabort>
    <cfelse>
            <!--- Process login information --->
    </cfif>
</cfif>
```

Let's look at the series of requests and responses that occur between a Web browser (WB) and a server (S) when the user requests a page that's protected by this kind of authentication code:

WB: Please send me the page called /index.cfm.
S: OK, here's the page called /index.cfm. (The server sends a page containing a form with text fields for user name and password and a submit button.)
WB: (After the user fills in the form and clicks the submit button) Here's a POST to the page called /index.cfm.
S: OK, I processed the submission and here's the page called /index.cfm.

Notice how the server claimed the two pages it sent were both /index.cfm? That's an example of one problem with forms-based authentication: it misrepresents the resources identified by URLs. To illustrate another problem, take a look at what happens later, after the user takes a break from using the application and the session expires:

WB: Please send me the page called /administrator/addUser.cfm.
S: OK, here's the page called /administrator/addUser.cfm. (The server sends a page containing a form with text fields for user name and password and a submit button.)

Since the session expired, the server responded to the request by providing the login form again. Even if the form provides some information about why the user needs to log in again, the user may become confused, and at the very least is inconvenienced.

There are other ways to implement forms-based authentication (such as using <cflocation> to redirect the browser to the login form instead of using <cfinclude> to send it in place of the requested page), but the basic issues are the same:

1. Forms-based authentication misrepresents the resource requested by the browser. Depending on the implementation, it claims the login form is the resource that was requested, or it sometimes claims that the resource requested is located somewhere else (that "somewhere else" being the location of the login form).
2. It forces the user to understand technical issues like session expiration.
3. Since it's dependent on session cookies, it's not easily scalable to cluster environments and doesn't work if the browser has cookies disabled.
4. Since it's dependent on the end user to read the login form and understand that it's not the resource the user requested, it's not compatible with alternative user agents

More Detail: URLs and HTTP Status Codes

What's all this talk about "misrepresenting the resource," you say? It boils down to URLs and status codes.

Each time the browser requests a page, it specifies the exact resource (page) it's requesting. This resource is identified by the URL and a basic tenet of HTTP is that two requests for the same resource (assuming there's no form submission happening) should return exactly the same result (this is closely related to the property of HTTP requests called "idempotency," the principle that GET requests shouldn't change the state of the resource being requested). When the server sometimes sends a login form and sometimes sends the actual page, it's breaking this one-to-one correspondence between the URL and the resource.

Each time the server responds to a request, it sends a status code that very succinctly tells the Web browser the "gist" of its response. Most of the time (when it's serving ColdFusion pages at least), the server sends the status code 200 ("OK"), which means "this page I'm sending you is exactly what you asked for." If instead of executing the page the user requested, the server is sends a login form, it's lying to the browser when it says it found what the user asked for.

When the ColdFusion server processes a <cflocation> tag, it sends the status code 302 ("Found"), which means "the page you're looking for is temporarily located somewhere else and I'm sending you its location." If the address the server sends isn't actually the page the user requested, but a login form, it's lying again.

These distinctions may seem pedantic. But every Web browser since NCSA Mosaic has been raised speaking HTTP as its native language and there's a surprising amount of potential to be leveraged in modern browsers when you respect the true meaning of the terms in that language. And sometimes when you ignore the rules of HTTP you get burned – a notable example of this was when Google released a beta of its Web Accelerator and lots of developers who had ignored the principle of idempotency found that their Web sites were breaking.

Basic Authentication to the Rescue

All the way back at the dawn of the Web (May 1996, to be specific), when Tim Berners-Lee codified the rules of the Hypertext Transfer Protocol (HTTP/1.0), he and his co-authors considered the problem of how to secure resources on the Web and their answer to the problem has come to be known as HTTP Basic Authentication. With Basic authentication, the initial dialogue between Web browser and server looks more like this:

WB: Please send me the page called /index.cfm.
S: Sorry, that page is part of an application called "MyApplication." You can't see that page without providing a user name and password. Here's a page to tell the user why he can't see the page he asked for.
WB: (Prompts the user for a user name and password.)

Efficient Web Content Management

CommonSpot™ Content Server is the ColdFusion developer's leading choice for efficient Web content management.

Our rich Web publishing framework empowers developers with out-of-the-box features such as template-driven pages, custom content objects, granular security, and powerful ColdFusion integration hooks (just to name a few), allowing you to rapidly build and efficiently deploy dynamic, high performance Web sites.

CommonSpot's open architecture and extensive APIs enable easy customization and integration of external applications. With CommonSpot, you can design and build exactly what you need. Best of all, CommonSpot puts content management in the hands of content owners. With non-technical users responsible for creating and managing Web content, developers are freed to focus on strategic application development.

Evaluate CommonSpot today.
To see your site *running* under CommonSpot, call us at 1.800.940.3087.

features.

- 100% browser-based
- Content object architecture
- Template driven pages
- 50+ standard elements
- Content reuse
- Content scheduling
- Personalization
- Flexible workflow
- Granular security
- Mac-based authoring
- 508 compliance
- Full CSS support
- Custom metadata
- Taxonomy module
- Extensible via ColdFusion
- Web Services content API
- Custom authentication
- Replication
- Static site generation
- Multilanguage support

1.800.940.3087
www.paperthin.com

Paper Thin

Please send me the page called /index.cfm – and here are the user's credentials to access that page.
S: OK, I processed the credentials and here's the page called /index.cfm.

Doesn't that look like a much more productive exchange than when the server was telling all sorts of lies about the resources it was sending to the browser? In fact it is more productive and here are a few reasons why:

1. HTTP status codes and URLs are used exactly as they were intended.
2. Once the browser knows the user's credentials for the application, it can continue to provide them when needed without prompting the user again (more on this below). Depending on your application, this might mean the end of session expiration!
3. It's not dependent on keeping login credentials in the session scope, so it even works when browsers have cookies disabled. (Your application may still need to use session-scope variables for other purposes.)
4. It allows alternative clients (such as command-line and automated user agents) to access your application, unlocking the potential for new and exciting uses of the data in your application.

Web Server-Integrated Basic Authentication

Basic authentication doesn't have to be implemented in application code. All the most common Web servers (including Apache and Microsoft's Internet Information Server) allow the administrator to protect resources with Basic authentication by setting some configuration parameters. In some cases, this may be sufficient, but if you want to validate the user login information against an existing user database (the one built into your application, for example), you'll almost certainly find it easier to keep the authentication code integrated with the rest of your application.

How to Implement Basic Authentication with ColdFusion

The first time a Web browser requests a page that's protected by Basic authentication, the Web server needs a way to tell it that it should stop and ask the user for login information. It does this by sending a response to the initial request with the status code 401 ("Access Denied"). In this response, it also sends information about what types of authentication it's willing to accept (this is the "WWW-Authenticate" header). In this article, we're only discussing Basic Authentication, but other more complex (and more secure) methods exist such as Digest authentication and Microsoft's proprietary NTLM authentication. Along with the 401 response the server also sends a user-friendly (or at least user-readable) explanation of why access is being denied.

When the browser gets the 401 response, before showing the user the "Access Denied" message, it prompts for a user name and password. If the user provides credentials, the browser repeats the request and includes the user name and password in an "Authorization" request header. If the credentials are sufficient, the server then sends the requested page with a 200 ("OK") status code. If the credentials aren't acceptable to the server, it simply sends back the same 401 response, which makes the browser prompt the user again. If the user declines to provide credentials (by clicking a "Cancel" button on the password prompt, for example), or fails to provide acceptable credentials a certain number of times (usually three, although it's up to the Web browser to decide), the browser stops prompting the user and shows the user the "Access Denied" message provided by the server.

Listing 1 shows the code needed to implement Basic Authentication. This code should be included at the top of any page that needs to be protected by authentication (an Application.cfm file is a good place, for example).

The first lines use the built-in ColdFusion function GetHttpRequestData() to retrieve complete information on the request headers sent by the client and ensure that our session-level user variable is initialized. Then the code checks to see if the client has provided the "Authorization" header that contains the login and password. The "Authorization" header consists of two parts separated by a single space: first, the authentication scheme (in this case "Basic"); and second, the user name and password separated by a colon and Base64-encoded to protect them from character set issues during transmission. The user name and password are decoded using the ToBinary() and ToString() functions, and the application verifies that they're correct (obviously this part will vary from application to application). If the user hasn't been logged in either by providing incorrect credentials or by not providing any at all, the response code is set to 401 and the "WWW-Authenticate" response header is added. Finally, a user-friendly "Access Denied" message is added to the response and execution is stopped so the server sends only the "Access Denied" message back to the client.

Even if the user has cookies disabled, so the user's session doesn't persist from request to request, the user won't be prompted again for a login and password. However, in this case, you won't be able to store any other data in the session scope without resorting to passing session identifiers as URL parameters.

Since the browser is getting accurate response codes throughout the transaction, there's no need to write code to redirect the user after a successful login. This is an additional benefit of using Basic authentication and means that your application fully supports deep linking and bookmarking without any extra code.

The Authentication Realm

There's one additional piece of information that the server sends in the "WWW-Authenticate" header: the authentication realm. The realm is defined as a string that uniquely identifies the set of resources on the server that all use the same user account source. The browser assumes that if it has prompted the user for a login and password for a resource on a given server, it can reuse the same credentials for another resource in the same realm on the same server.

Although the specification for Basic authentication discourages the use of the realm for any purpose other than matching it for equality against other realm values, in practice the realm actually has another important function. When most browsers prompt the user for login credentials, they show the realm as a description of the application the user is logging in to. So it's a good idea to set the realm in your authentication code to a brief description of your application (this description would replace the string "MyApplication" in Listing 1).

Caching of Credentials: The End of Session Expiration

When a browser gets a 401 ("Access Denied") message in response to a request for a given resource, it checks to see if the realm specified in the "WWW-Authenticate" header matches any realms it has already authenticated against on the current server for the current browser session. If it finds credentials that have been accepted for that realm, it retries the request using those credentials (prompting the user for credentials only if the cached credentials are rejected by the server). The browser also assumes that resources in subdirectories below the resources for which it has already been prompted to authenticate will be part of the same realm and preemptively sends the cached credentials the first time it requests these resources. These two facts contribute to one of the biggest advantages of Basic authentication: the end of session expiration messages.

Examine Listing 1 again and consider what happens if the user has been previously authenticated but has been inactive enough that the session expired. On the next request, a new session is initiated by the ColdFusion server and SESSION.loggedInUserName is set to an empty string. If the browser has previously requested the page, it will have preemptively sent the login credentials; if not, it will check its cache and see that it already has credentials for the specified realm and it will repeat the request, providing the cached credentials. In either case, the user has been automatically logged back into the application without being prompted! This provides a much smoother and less confusing experience for the user.

Keep in mind that this may have some affect on your application architecture. If you're using the session scope only to cache information that's already stored in a database somewhere, your application can probably be converted to Basic authentication with no issues. However, if you store temporary data in the session scope (such as shopping cart information), you may have to consider a different approach since the user may have multiple ColdFusion sessions over the course of what they consider a single session of interacting with the Web application. Just remember that a user session might be initialized on any page of your application.

Limitations of Basic Authentication

Basic authentication isn't without its limitations. First and foremost, it's important to realize that the user name and password is transmitted from the browser to the Web server in unencrypted clear text. (The Base64 encoding isn't a security measure – it's designed to be easily reversible.) If you consider the data in your application to be sensitive, you should definitely use

Other companies in this magazine spent a lot of time on pretty ads. As you can see, we did not. We spent our time hiring the best people and training them to deliver outstanding support for your website. We spent our time building a state of the art datacenter and staffing it with people who care about your website like it's their own. Compassion, respect, credibility, ownership, reliability, "never say no," and exceed expectations are words that describe our service philosophy. From the first time you interact with us, you'll see what a difference it really makes. And you'll also forgive us for not having a pretty ad.

HostMySite.com

WEB HOSTING • MANAGED DEDICATED SERVERS • COLOCATION • VPS • ECOMMERCE • BLOGGING • EMAIL an additional security layer such as SSL to protect the transmission of the user name and password (as well as the data). That being said, this lack of encryption is a limitation that's common to both forms-based and Basic authentication.

The other significant limitation of Basic authentication, and the only major drawback to using it over forms-based authentication, is that it provides much less opportunity for context around the login and password. A forms-based solution can show information around the login box explaining the format of the login and, in the case of password errors, can provide detailed information about what went wrong. Additionally, the login prompt can be customized to match the look-and-feel of the rest of the application. When implementing HTTP Basic authentication, you should provide an entry page that isn't authenticated that provides this information so the user is prepared to enter the appropriate login and password information.

Conclusion

Basic authentication provides an alternative to traditional forms-based authentication methods. By sticking closely to the HTTP specification, this method leverages features built in to the Web browser to prevent confusing session expiration problems and the need to code around other limitations in non-standard forms-based authentication methods.

About the Author

Patrick Correia is a Web developer for Clough Harbour & Associates LLP, an east coast multi-disciplined engineering firm. A Certified Advanced ColdFusion MX Developer based in Albany, New York, he has spent the last five years developing ColdFusion-based business process improvement solutions for the firm's numerous municipal and private clients.

pcorreia@cha-llp.com

Listing 1

```
<cfset reqData = GetHttpRequestData() />
<cfparam name="SESSION.loggedInUserName" default="" />
<cfif IsStruct( reqData )
   AND StructKeyExists( reqData, "Headers" )
   AND IsStruct( reqData.Headers )
   AND StructKeyExists( reqData.Headers , "Authorization" )
   AND REFindNoCase(
        "Basic [A-Za-z0-9+/=]+", reqData.Headers.Authorization )>

    <cfset credentials = ToString( ToBinary(
        ListLast( reqData.Headers.Authorization , " " ) ) )/>
    <cfset username = ListFirst( credentials, ":" ) />
    <cfset password = ListLast( credentials, ":" ) />

    <cfquery name="qryCheckCredentials">
        <!--- Check login information --->
    </cfquery>
    <cfif qryCheckCredentials.RecordCount IS 1>
        <cfset SESSION.loggedInUserName = username />
    </cfif>

</cfif>
<cfif SESSION.loggedInUserName IS "" >

    <cfheader statuscode="401" statustext="Access Denied" />
    <cfheader name="WWW-Authenticate"
        value="Basic realm=""MyApplication""" />
    <cfinclude template="accessDenied.cfm" />
    <cfabort/>

</cfif>
```

REGISTER NOW!
www.iTVcon.com

CALL FOR PAPERS NOW OPEN!

Welcome to the Future of Video on the Web!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2006

Coming in 2006 to New York City!

"Internet TV is wide open, it is global, and in true 'Web 2.0' spirit it is a direct-to-consumer opportunity!"

For More Information, Call 201-802-3023 or Email itvcon@sys-con.com

Welcome to the Future!

Did you already purchase your ".tv" domain name?
You can't afford not to add Internet TV to your Website in 2006!

2005 was the year of streaming video and the birth of **Internet TV**, the long-awaited convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company must now provide video content to remain competitive, not to mention live and interactive video Webinars and on-demand Webcasts.

20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today's well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully featured PVRs, and significant advances in streaming video technologies, **Internet TV** is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: **Internet TV** is about to change forever the $300BN television industry, too.

The Internet killed most of print media (even though many publishers don't realize it yet), Google killed traditional advertising models, and **Internet TV** will revolutionize television the way we watch it today. You need to be part of this change!

Jeremy Geelan

Jeremy Geelan
Conference Chair, iTVCon.com
jeremy@sys-con.com

PRODUCED BY
SYS-CON EVENTS

List of Topics:

- > Advertising Models for Video-on-demand (VOD)
- > Internet TV Commercials
- > Mastering Adobe Flash Video
- > How to Harness Open Media Formats (DVB, etc)
- > Multicasting
- > Extending Internet TV to Windows CE-based Devices
- > Live Polling During Webcasts
- > Video Press Releases
- > Pay-Per-View
- > Screencasting
- > Video Search & Search Optimization
- > Syndication of Video Assets
- > V-Blogs & Videoblogging
- > Choosing Your PVR
- > Product Placement in Video Content
- > UK Perspective: BBC's "Dirac Project"
- > Case Study: SuperSun, Hong Kong

Track 1	Corporate marketing, advertising, product and brand managers
Track 2	Software programmers, developers, Website owners and operators
Track 3	Advertising agencies, advertisers and video content producers
Track 4	Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers

Designing ColdFusion applications with UML

By Robert Blackburn

Have you ever watched actors improvise? It's quite fun; you never know what's going to happen. But imagine if a movie director decided a script wasn't necessary at all.

After all, the actors know the plot and know the most about their characters, so why not let them improvise. Wouldn't this lead to a more creative and realistic movie? I think we can all see that this would simply lead to a confusing story with no consistency. The actors are expected to impose their own talents on their roles, but they need a script to guide them along.

Developers are no different, yet I see "improv developing" all the time. A project manager will just describe what's needed – the outlines of a plot – and let the developers work it out. The results are often what you'd see with a movie. However, designing an application with UML can act as a script for developers. They're still free to use their own creativity and experience when implementing the design, but the UML documents provide them with a map of what the final product should be.

So it's no surprise that UML is quite popular in many software development communities, and you even see it used occasionally for ColdFusion. Yet how can ColdFusion developers really leverage UML to avoid improv development?

I'd like to show you some ways I've used UML to help me design ColdFusion applications. In this article I'll first explain what UML is, then how to read it, and finally how to create it.

What Is UML?

UML or Unified Modeling Language isn't a markup language like HTML or XML, and it's not a programming language like CFML. It's a unified method for modeling systems. This is usually used for software, but it can model any system. There are a number of benefits to designing an application. Like better team coordination, faster development cycles, and more scalable application designs.

Whether you're a freelance, corporate, or Open Source developer most of us need to work on a team. By designing an application upfront project managers can get a bird's eye view of a system before development starts. This lets them divide the work among multiple developers or groups. It also lets those developers know what to expect from the other part of the system without having to open the code.

Designing a system first also forces the designer to consider how the system will work before coding begins. Instead of putting those decisions off until later, they're made earlier. It's easier and faster to move a few blocks around a diagram then it is to refactor code. UML will help uncover faulty design decisions and reveal potential problem areas.

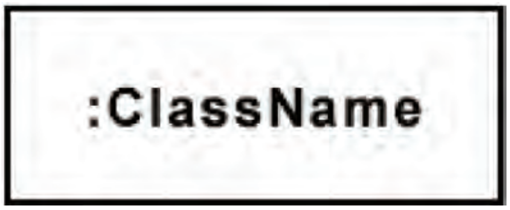
Lastly, most well-designed applications will pay close attention to the separation of concerns (SoC). MVC (Model-View-Controller) is one common way to achieve SoC. In fact, most ColdFusion frameworks are designed to enforce MVC (they often take the role of the controller). UML also helps enforce better application designs, like MVC. This creates more scalable, stable, and bug-free applications.

Reading UML

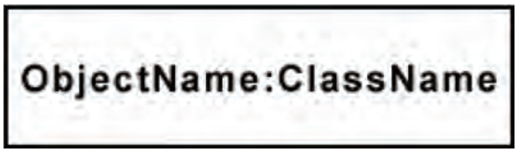
Most UML diagrams are fairly intuitive; you can glean most of what you need simply by studying them. But to write good UML you need to have a basic understanding of the notations used and what the different symbols mean. So before I show you how to use UML to create a ColdFusion application I'll explain a few of the elements you'll see in this article and how they relate to ColdFusion.

Class Symbol

At the heart of any OO application are "classes." In ColdFusion these are most directly related to components (CFC). The most basic symbol of a class in UML is just a box with the class's name in it like:

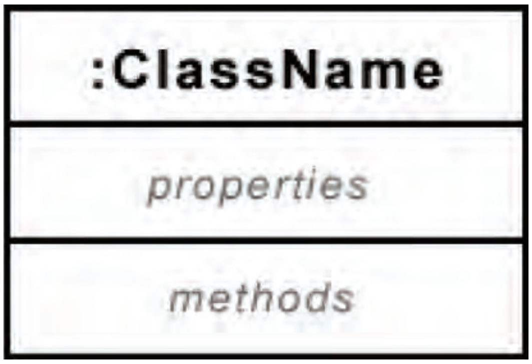


You might notice that there's a colon before the class name. Anything to the right of the colon is the class's name; anything to the left is the name of the specific object like:



This comes in handy when documenting an interaction between specific objects of a class, especially when two objects belong to the same class and you have to distinguish between them. But most of the time when designing an application you only see a class name.

If more details are needed about the class, this symbol can be expanded on. One or two more boxes can be added, the first containing the class's properties and the second its methods like:

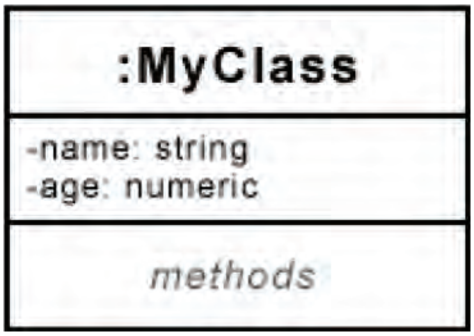


There's a special notation used to list properties and methods in these blocks.

Here's the notation for properties:

```
-propertyname: datatype
```

The first character is a plus or minus sign, which stand for a private (-), or public (+) property. In a CFC, this means that you'll either scope the properties with "VARIABLES" (private) or "THIS" (public). You'll almost always have private properties, so you'll almost always use the minus sign to prefix your properties. The property name is whatever name you intend to give this variable inside the code. The name is followed by a colon and the data type of the property. For example:



If I were to open the "MyClass.cfc" file, I'd expect to see these lines of code:

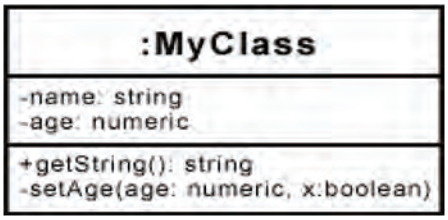
```
<cfpoperty name="name" type="string" />
<cfpoperty name="age" type="numeric" />
```

And I'd expect to see these variables referred to as VARIABLES.name, and VARIABLES.age.

The notation for methods (known as functions in ColdFusion) isn't all that different:

```
+methodname(argument: datatype, argument: datatype): datatype
```

As you can see, the methods start off the same way as properties, with a plus or minus to indicate public or private. Then the name of the method itself is listed. The real difference is what follows the method name: a set of parentheses containing a list of the method's arguments. Each argument follows the same format as the properties - name:type. After the arguments is whatever data type, if any, the method itself returns. Let's expand our previous example:



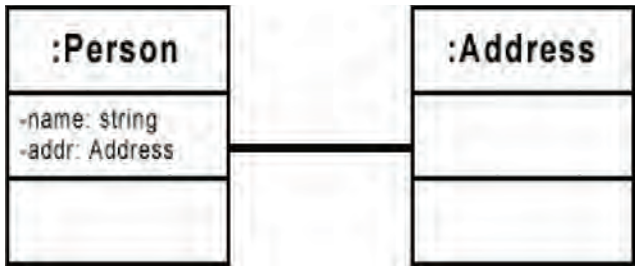
Now if I open the MyClass.cfc file I'd expect to see this code:

```
<cffunction name="getString" access="public" returntype="string">
    ... code ...
</cffunction>
<cffunction name="setAge" access="private" returntype="void">
    <cfargument name="age" type="numeric" required="true" />
    <cfargument name="x" type="boolean" required="true" />
    ... code ...
</cffunction>
```

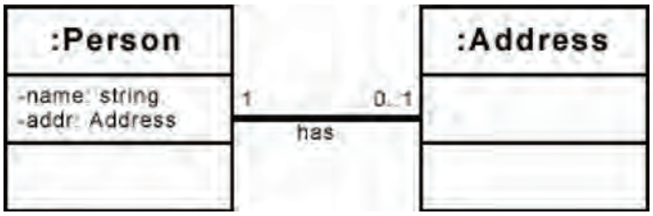
Relationships

Classes don't live in a bubble; they interact with one another. Any sort of interaction between classes is a type of relationship. A relationship is illustrated with a line or arrow between the two classes. There are many kinds of relationships in UML, but we'll focus on just three in this article, associations, generalizations, and messages.

An association means that one class uses another in some way. That could mean the first class utilizes a method of the second. Or it could mean that the first class contains the second as one of its properties. For example:



In this example, the Person class has an "addr" property that has a type of "Address." So there's an association between the two classes. The association line also has its own notation. At each end of the line is a "multiplicity" and in the middle is a "descriptor":

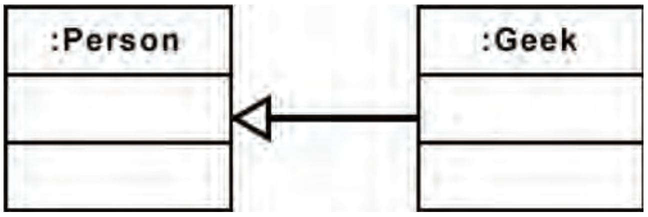


The descriptor in the middle simply describes the type of relationship. The multiplicities at each end of the line indicate how many of these classes are involved in the relationship. If you've ever worked with database schemas these may look somewhat familiar to you. Values can be a comma delimited or two dots (..) that indicate a range of values. An asterisk indicates an unknown number. For example:

0..1	Zero or 1
*	Any amount (zero or more)
1..*	Any amount greater then one
1,3,4	1, 3, or 4
1..6,10	1 through 6, or 10

These association lines should actually be readable top-down, left-to-right. So in the example above, one person has zero or one address.

The second type of relationship is a "generalization." This is when one class inherits another; in ColdFusion this is called extending. The rule of thumb is if you can say that "A is a type of B", then "A is a generalization of B". Or you could say B extends A. A generalization is noted in UML with a solid line with an open arrow at one end. For example:



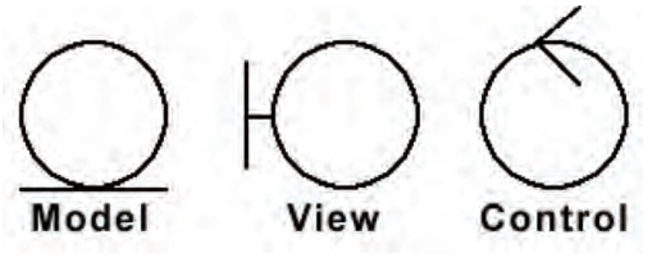
This says that a Geek is a kind of Person (they really are you know). In the Geek.cfc I'd expect to see this code:

```
<cfcomponent extends="Person">
    ...code...
</cfcomponent>
```

The last type of relationship you'll see in this article is messages, which are used in collaboration and sequence diagrams. Messages are when one class calls the method of another class. An arrow between the two classes is how we illustrate a message, usually with a description or method name just above or beside it. You'll see more on how messages are used in UML when we cover collaboration and sequence diagrams.

MVC symbols

As we said, UML can help enforce better MVC design. There are actually three special symbols used to indicate those kinds of elements:



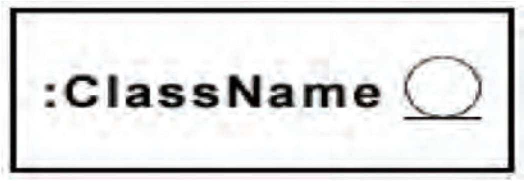
What does Model-View-Controller mean?

- Model: A person, place, or thing. This class is intended to represent (model) something in real life like a "car" class.
- View: These are the elements the user will interact with. They are forms, pages, pop-up windows, even e-mails.

- Controller: This is the middleman, the one that controls the flow of information. If the view needs to modify a model, it should go through a controller.

For a great example of using MVC in ColdFusion take a look at "ColdFusion Best Practices" at http://www.benorama.com/coldfusion.

These symbols can be put inside the class icon to indicate the type of class:



This isn't official UML – to my knowledge – but I've found it to be very useful.

Creating UML

There are a few basic UML diagrams I use to design a ColdFusion application. A UML diagram can't be created in a single pass. Like writing a story it takes a number of iterations to refine and complete. The process we'll follow here will help us gradually explore and define each diagram, using one diagram to help explore the next. The point of this process will be to generate a suite of diagrams that will tell you almost anything you need to know about how your application will work. The basic suite I use consists of three diagrams: a Class Diagram, a Collaboration Diagram, and a Sequence Diagram. This is only a subset of the UML diagrams available, and complex applications may require other more specialized diagrams to complete the suite. But in this article we'll only look at creating that core set of diagrams.

The Class Diagram

The first thing I do when designing a system is to create a class diagram. What's the point of a class diagram? It's a view of all the system's elements and how they'll interact with one another.

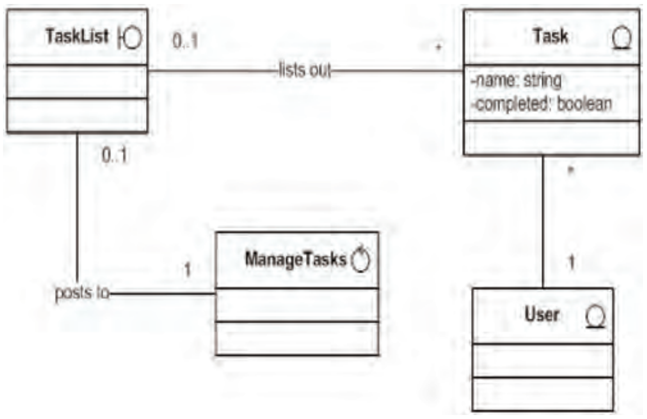
To get started we'll need to identify our classes. There's an old trick in the OO world that will work well here. Take the Use Case, or whatever document you used to gather the system's requirements, and highlight all the nouns. Take those nouns and put them into an Excel spreadsheet. Only put each noun in once, ignore duplicates, and ignore pronouns. This is the initial list of classes.

Next, in the column following the nouns, enter what type of class or element the noun is. Common possibilities are Model, View, Control, Property, or Other:

- Model: If the noun refers to something in the real world like "car" or "dog" then the noun is a model class (it "models" something).
- View: If it refers to something that the user interacts with, it's a view. Examples are forms, pages, pop-up windows, or e-mails.
- Control: If the noun is the name of a use case, process, or sequence of events then it's a controller.
- Property: This sometimes comes up when referring to the property of a class. For example, the use case might say, "Display the name of a person." There are two nouns there, "person" and "name." The "person" is a model class, and "name" is one of its properties.
- Other: Anything that's unclear or doesn't fit into one of the types above. Try to use this as little as possible, and if you do, write in the third column why it doesn't fit or what's unclear about it.

Don't stress the class identification too much; it's really just to help get the class diagram started so it does not have to be perfect. Just take your best guess; you can always change it later. We can now start the class diagram. Put all the classes from the spreadsheet into the diagram and note what kind of class they are (Model, View, Control) using the three symbols. Also add any properties you identified. Now add any relationships, methods, or properties that are obviously going to be needed. You're still drafting the class diagram so just put in what you know will be needed, and don't worry about how it looks, we can clean it up later.

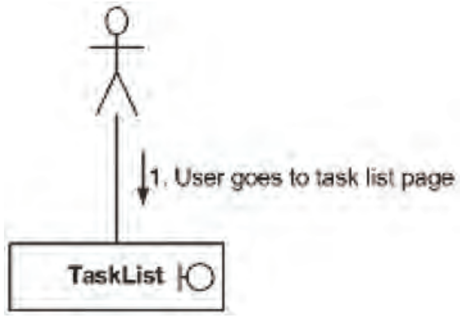
Let's say I'd like to create a very basic to-do list application. In this application the user goes to a task list page and all his uncompleted tasks are listed. He can mark a task as completed or add a new task by naming it and clicking an "add" button. To start my class diagram I'll take whatever requirements document I have, even if it's just a description like what I just gave, and identify the nouns. I use that to draft a class diagram, and add a few of the obvious relationships, methods, and properties. This is what I end up with:



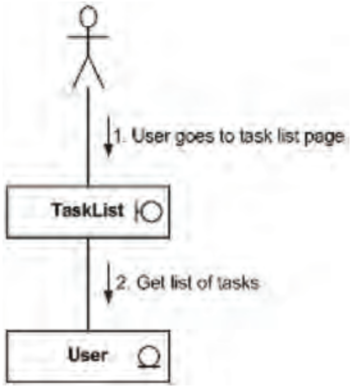
It's far from complete; most of my classes don't even have any details. I just fill in what I know so far.

The Collaboration Diagram

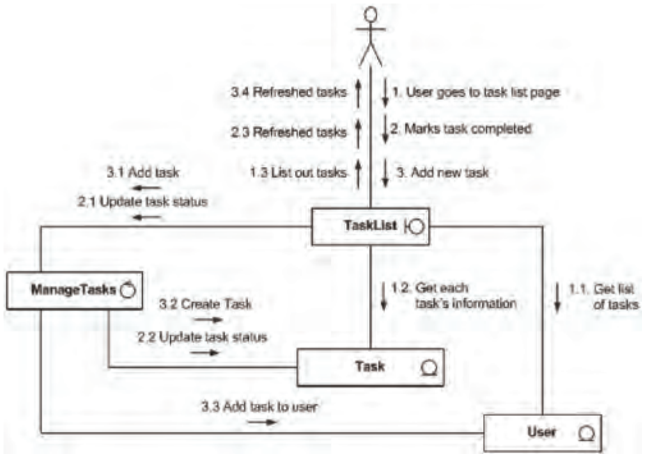
The first draft of the class diagram is now complete. Now we'll use the class diagram and make our collaboration diagrams. Start by copying the class icons from your class diagram to a blank diagram. Draw lines between the elements you think will interact with one another. Add a stick figure to represent the user and draw an arrow from the user to the first element in the diagram the user will interact with. Put a number "1" next to this arrow followed by a description of the interaction. Let's use the task list application from our previous example:



Then determine what the next interaction will be. In our example the TaskList view would have to get the list of tasks; let's assume it would get that from the User class:



Keep doing this, step-by-step, mapping the interactions of the application. This is another area that a Use Case, if available, can help. Again, don't worry about it being perfect just do your best. Here's what I ended up with after walking through my description of the task list application:

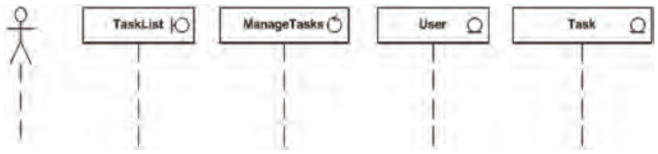


You might notice a few things. First, I used fractions for numbering some of the step (like 1.1, 1.2, 1.3...). This is a common technique to keep a series of step clear and it also makes it easier to add more steps later. The important thing is to keep the diagram clear and easy to read. I could even create three separate diagrams for each of these sequences if I felt that would be easier to read.

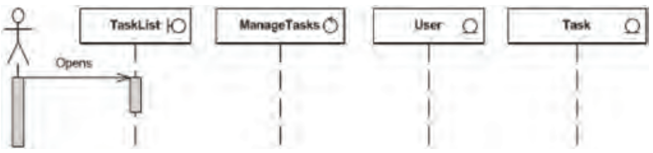
Second, you might notice that the TaskList page goes to the ManageTasks controller to update or add a task. Why do I do this instead of just updating the Task element? Well, as mentioned before when talking about MVC a view element can't modify a model element, it mostly goes through a controller. This is simply how I've chosen to architect my application. If you have a different standard, or use a framework, the collaboration diagram is still completely applicable; you simply diagram the interactions according to your own standards.

The Sequence Diagram

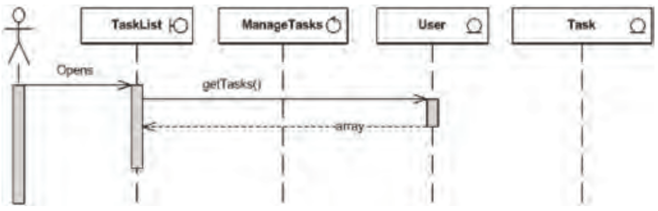
Now we have the first draft of the collaboration diagram. We've already revealed some relationships that were missed in the class diagram. But let's not revise the class diagram just yet; let's create a sequence diagram first to clarify these details even more. We'll use the collaboration diagram to create the sequence diagram. Once again start with a blank diagram and copy all our classes over to it, but line them along the top side-by-side. Put the views on the left, the controllers in the middle, and the models on the right. Then add another user stick figure on the far left. Below each of these draw dashed lines; these are the "lifelines" of the elements:



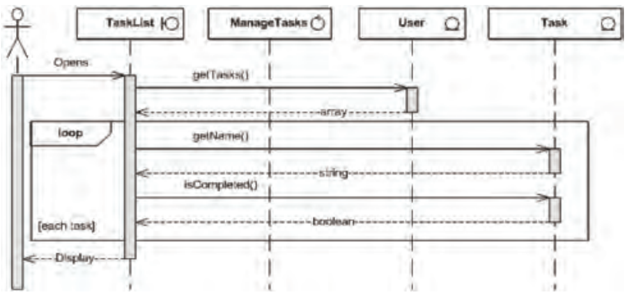
Now we'll map the same sequence of events from the collaboration diagram in more detail. Start with the first interaction on the collaboration diagram and draw a line from the user to the TaskList:



Notice I gave this arrow a short name that describes the interaction and put blocks on the two element's lifelines. These blocks represent activity. This first interaction is nothing existing, so let's move on to the next step in the sequence, step 1.1 from the collaboration:

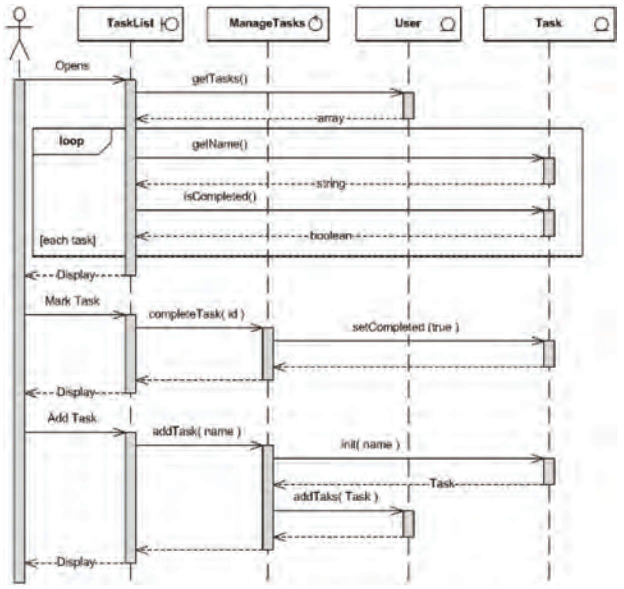


Ah, now things are more interesting. This time we have an interaction between the system elements, so instead of a description of the interaction, we actually gave it a method name. Now we not only know that these two elements interact with one another we're actually defining what methods will be used during the interaction. You might also notice the dotted line arrow going from the User back to the TaskList; this is a return message, and I've noted that an array is being returned. Now we've even defined what kind of information the method will return. Let's move on to the third step, 1.2:



Notice how I've taken "Get each task's information" and expanded that out into two methods, getName() and isCompleted(). Also notice the box around those methods with the funny looking upper-left corner titled "loop." This is how UML frame's special logic flows. This obviously indicated a loop. The note in the lower corner states "[each task]" meaning we'll loop the framed content once for each task. In other words, we're going to get the name and completion status of each task. Right after the loop we simply return the page to the user.

Keep walking through each interaction on the collaboration diagram adding them to the sequence diagram, giving system interactions actual method names. Here's what I came up with for our task list application:

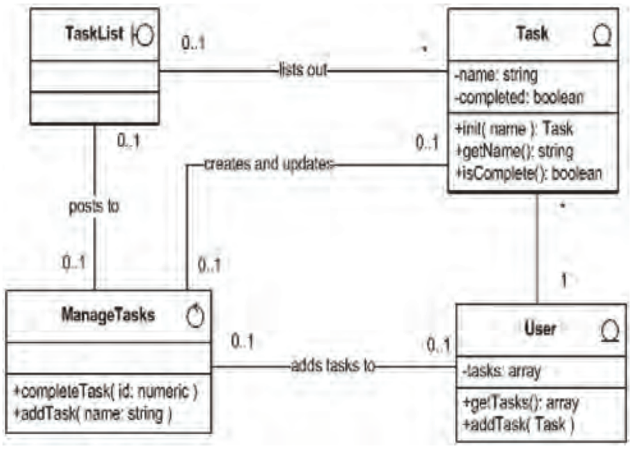


There are two things worth noting in this diagram. First, there are arguments for some of the new methods I added. I noted those by simply adding them in the parenthesis of the method.

The second thing I'd like you to notice is that the diagram doesn't specify how the TaskList calls the completeTaks() or addTask() method. This could be done with AJAX, a normal page refresh, or by posting to the CFC as described on http://www.benorama.com. The sequence diagram is very flexible this way; it helps clarify how elements will interact and communicate, but can apply to many different technologies or techniques. The previous sequence diagram could apply to Java or PHP just as easily as ColdFusion.

Rinse and Repeat

We now know a great deal more about how our application will work than we did when we first created the class diagram. Now it's time to revise the class diagram based on what we discovered in the collaboration and sequence diagrams:



This revision might have resulted in new classes, or classes being dramatically changed. Don't shy away from making drastic changes, that's what this process is all about.

Are we done? Not quite yet. We now repeat these steps, returning to the collaboration diagram to refine it further, and adding anything we might have missed the first time. Then revise the sequence diagram and go back to the class diagram. We'll repeat this process until we feel confident that the diagrams are complete and accurate.

Summary

This article covered some very basic UML. No single article can cover UML in its entirety; that would be like trying to explain everything about ColdFusion in a single article. But in the end the suite of diagrams I've showed you here will describe an application in great detail, and give you a starting point from which to explore UML further. One could hand these diagrams over to a developer – just like someone would hand a script to an actor – and improv development would be avoided.

About the Author

Robert Blackburn is a Web developer.

rwblackburn@gmail.com

Rich Internet Applications: AJAX,

SAN JOSE SILICON VALLEY

SYS-CON Events is proud to announce the first-ever AjaxWorld Conference & Expo 2006!

The world-beating Conference program will provide developers and IT managers alike with comprehensive information and insight into the biggest paradigm shift in website design, development, and deployment since the invention of the World Wide Web itself a decade ago.

The terms on everyone's lips this year include "AJAX," "Web 2.0" and "Rich Internet Applications." All of these themes play an integral role at AjaxWorld. So, anyone involved with business-critical web applications that recognize the importance of the user experience needs to attend this uniquely timely conference – especially the web designers and developers building those experiences, and those who manage them.

CALL FOR PAPERS NOW OPEN!

We are interested in receiving original speaking proposals for this event from i-Technology professionals. Speakers will be chosen from the co-existing worlds of both commercial software and open source. Delegates will be interested learn about a wide range of RIA topics that can help them achieve business value.

Flash, Web 2.0 and Beyond...

REGISTER TODAY AND $AVE!

October 3-4, 2006

Santa Clara Convention Center
Hyatt Regency Silicon Valley
Santa Clara, CA

To Register
Call 201-802-3020 or
Visit www.AjaxWorldExpo.com

May 7-8, 2007
First International AjaxWorld Europe
Amsterdam, Netherlands

"Over the two information-packed days, delegates will receive four days' worth of education!"

Early Bird*
(Register Before August 31, 2006)
.... $1,495**
See website or call for group discounts

Special Discounts*
(Register a Second Person)
.... $1,395**
See website or call for group discounts

(5 Delegates from same Company)
.... $1,295/ea.**
See website or call for group discounts

On-Demand Online Access
(Any Event)
.... $695

*Golden Pass access includes Breakfast, Lunch and Coffee Breaks, Conference T-Shirt, Collectible Lap-Top Bag and Complete On-Demand Archives of sessions in 7 DVDs!

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE, PLEASE SEE WEBSITE FOR UP-TO-DATE PRICING

"It Was The Best AJAX Education Opportunity Anywhere in the World!" —John Hamilton

Topics Include...

Themes:
- > Improving Web-based Customer Interaction
- > AJAX for the Enterprise
- > RIA Best Practices
- > Web 2.0 – Why Does It Matter?
- > Emerging Standards
- > Open Source RIA Libraries
- > Leveraging Streaming Video

Technologies:
- > AJAX
- > The Flash Platform
- > The Flex 2 Framework & Flex Builder 2
- > Microsoft's approaches: ASP.NET, Atlas, XAML with Avalon
- > JackBe, openLaszlo
- > JavaServer Faces and AJAX
- > Nexaweb
- > TIBCO General Interface

Verticals:
- > Education
- > Transport
- > Retail
- > Entertainment
- > Financial Sector
- > Homeland Security

GROUP DISCOUNTS AVAILABLE:
— 5 Delegates from Same Company — for only $995 (each)
— Register a Second Person — for only $1195

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

Receive FREE WebCast Archives of Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all conference sessions. We will mail you the complete content from all the conference sessions in seven convenient DVDs after the live event takes place.

► This on-demand archives set is sold separately for $995

HYATT
HYATT REGECNY SILICON VALLEY

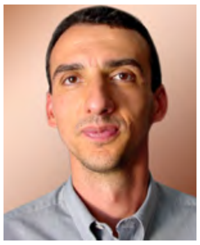
SYS-CON EVENTS

For more great events visit www.EVENTS.SYS-CON.com

VISIT WWW.AJAXWORLDEXPO.COM FOR THE MOST COMPLETE UP-TO-DATE INFORMATION

Version Control Using Subversion

Switching over from CVS



By Harry Klein

I've been meaning to switch from CVS to Subversion for quite a while. It seemed hard, but it actually took me only a couple of days to get it installed and configured for the whole development team.

This article describes the differences between CVS and Subversion and explains how to install Subversion and migrate an existing CVS repository. It also describes how to configure Subversion in a Windows environment, explains a basic Subversion project structure, and introduces the main Subversion clients. Finally, it shows ways to use Subversion with Ant and how to get connected to the repository via JavaSVN and ColdFusion.

Subversion Versus CVS

Subversion acts a lot like CVS, but fixes the flaws and addresses the shortcomings in the popular Concurrent Versions System (CVS).
The following features either fix CVS flaws or improve on existing CVS features and were the main reason for switching from CVS to Subversion:

- Subversion tracks both files and directories
- Subversion handles the copy, rename, and delete operations on files well. With Subversion, you can move files and directories.
- Branches and tags are normal directories. Tagging and branching are fast because tags and branches are just copied paths in the repository tree.
- Subversion performs its commit operations in an atomic fashion. So you will never have an inconsistent code tree.
- Subversion has a uniform method of dealing with all files, both binary and text, and uses a binary differencing algorithm.
- Subversion provides a well-documented API that lets other applications embed or extend it.

Installation and Repository Migration

There exist several Installation Guides for Subversion, like the "Mere-Moments Guide to installing a Subversion server on Windows".

If you're feeling lazy and prefer an automated install, you can use the "svn1clicksetup" project from Tigris. Svn1ClickSetup takes you through all the steps necessary to install the Subversion command-line utilities and TortoiseSVN as well as creating a repository and initial project.

There are several installation options for Subversion. I preferred the Apache installation. SVN Server together with Apache is the installation used most often and provides authentication, path-based authorization, and basic repository browsing.

I was advised never to use the Berkeley database so I selected a file system for our repository.
After the Subversion installation I was searching for a tool to import content from CVS to the new Subversion repository. It should preserve commits, authors, commit messages, and the dates of commits and also convert CVS branches and tags to SVN branches and tags.

The tool of choice was the Polarion "SVN Importer," a command-line utility that transfers data from other version control systems to SVN. It also supports the version control systems PVCS, VSS, ClearCase, and MKS.
All I had to do was modify the configuration file "config.properties," specifically the sections "SVN AUTOIMPORT OPTIONS" and "CVS PROVIDER CONFIGURATION." Then I had a long coffee break because the repository to migrate was very large. After five hours everything was imported without a single warning.

Configuration

I was happy to learn that Subversion supports directory and file properties. There are certain built-in properties, but you can also specify your own properties. Our existing files included the CVS keywords $Id$ and $Log$ in the comments at the top of the file.

Subversion supports $Id$, but not the $Log$ keyword. Subversion developers haven't included a $Log$ keyword (expands to list all of the log messages) because in their opinion all this extra stuff in the source files gets in the way of reading the code.
For Subversion to recognize things like $Id$ in source files, you have to tell it to do this explicitly by updating the "svn:keywords"

property.Changes such as this are versioned just like any other changes to the files so you'll have to commit after running this command for it to take effect.

To set the property automatically on files matching certain patterns, add lines such as this to your subversion/config file:

```
### Section for configuring miscellaneous Subversion options.
[miscellany]
enable-auto-props = yes

### Section for configuring automatic properties.
[auto-props]
•   .cfm = svn:eol-style=native;svn:keywords=Date Author Id
•   .cfc = svn:eol-style=native;svn:keywords=Date Author Id
•   .java = svn:eol-style=native;svn:keywords=Date Author Id
•   .htm = svn:eol-style=native;svn:keywords=Date Author Id
•   .js = svn:eol-style=native;svn:keywords=Date Author Id
•   .css = svn:eol-style=native;svn:keywords=Date Author Id
```

Both svn:keywords and svn:eol-style are file properties. svn:keywords is needed to add the keywords to the file; svn:eol-style determines the line ending character on a file.

I also thought about using the folder property "tsvn:logminsize," which would set the minimum length of a log message for a commit, but changed my mind.

The previous CVS installation used SSPI (integrated Windows authentication), so we needed something similar for Subversion. Subversion provides the SSPI module "mod_auth_sspi.so." Within the Apache httpd.conf file the following modules must be setup in order:

```
# Windows authentication module
LoadModule sspi_auth_module   modules/mod_auth_sspi.so

# subversion modules
LoadModule dav_svn_module "C:/Program Files/Subversion/bin/mod_dav_svn.
so"
LoadModule authz_svn_module "C:/Program Files/Subversion/bin/mod_au-
thz_svn.so"
```

I also added the authentication lines to the Location Tag:

```
<Location /svn>
   DAV svn
   # All repos subdirs of d:/svn_repos
   SVNParentPath d:/svn_repos

    # authentication
    AuthName "Subversion Authentication"
    AuthType SSPI
    SSPIAuth On
    SSPIAuthoritative On
    SSPIUsernameCase lower
    SSPIOmitDomain On
    SSPIDomain MUC
    SSPIOfferBasic On
    Require valid-user
    # authorization
    AuthzSVNAccessFile "d:/svn_repos/authorization.conf"
</Location>
```

A simple "authorization.conf" file would contain only these two lines:

```
[/]
•    = rw
```

This means that all users can read and write. Please consult the documentation for more advanced configuration options. If your Subversion repository is being served up through the Apache HTTP Server you can point any Web browser to your Subversion repository and navigate your way through the latest revision of your repository.

Subversion Project Structure, Clients and IDE Integration

The Subversion project officially recommends the idea of a "project root," which represents an anchoring point for a project. A "project root" contains exactly three subdirectories: /trunk, /branches, and /tags. These naming conventions are only a suggestion, but commonly used.

Trunk represents the main line of development, release Branches represent working code that differs from trunk. Tags (read-only) are significant events in a project's lifecycle.

A repository may contain only one project root or a number of them. Subversion versions the repository, not individual projects.

When you commit a change to the repository, make sure your change reflects a single purpose like fixing a specific bug, adding a new feature, or some particular task.

Our development team uses several IDEs like Homesite, Dreamweaver, or Eclipse (with the CFEclipse plug-in). In the past the preferred CVS Client was WinCVS.

After switching to Subversion, Homesite users installed the Tortoise SVN and were happy with the Windows shell extension. This extension let them execute SVN actions inside their IDE. CFEclipse users installed the Subclipse plug-in, which let them access the Subversion repositories inside Eclipse.

The Subversive project is a new Eclipse plug-in that provides Subversion support.

Ant Scripts

SvnAnt is an Ant task that provides an interface to Subversion revision control system. SvnAnt uses Javahl (a Java interface for the Subversion API) or the experimental svn command-line interface. SvnAnt supports most of the major Subversion commands. A sample Ant target with SvnAnt would read like this:

```
<!– define svn taskdef … ‡

  <target name="checkoutLatest">
    <svn>
      <checkout url="${svnant.latest.url}" revision="HEAD" destPath="src_
latest" />
    </svn>
  </target>
```

After several tries and connection problems I finally gave up and used the Ant exec task instead. Sample code from build.xml:

```
<!-- svn-settings -->
<property name="svn_root" value="SVN ROOT" />
<property name="svn_username" value="SVN USER" />
<property name="svn_password" value="SVN PASSWORD" />
<property name="svn_revision" value="HEAD" />

<!-- exports the ${module} from svn and puts it into the ${workdir}/
svn_checkout -->
<target name="checkOut">
    <exec executable="svn">
            <arg line="export ${svn_root} ${workdir}/svn_export -r
${svn_revision} --force --username ${svn_username} --password ${svn_pass-
word}"/>
    </exec>
</target>
```

Using JavaSVN

The JavaSVN Open Source library is used in several projects for interaction with Subversion. Projects that use JavaSVN include Subclipse, SmartSVN, and Atlassian JIRA. JavaSVN is a pure Java Subversion client library.

In the following example I use JavaSVN to access the Subversion repository from a ColdFusion application. It achieves a very simple task: It reads the latest revision from a file in the repository called “test.htm.”

The JavaSVN library must be accessible to ColdFusion so I copied javasvn.jar to the Directory “C:\CFusionMX7\wwwroot\WEB-INF\lib\.”

Example: getLatestRevision

First we have to initialize the library:

```
<!--- initialize the library to work with a repository for DAV (over http
and https) --->
<cfset CreateObject("java", "org.tmatesoft.svn.core.internal.io.dav.
DAVRepositoryFactory").setup()>
```

The configuration part is easy; we define the URL to the repository, the user credentials, and the file path:

```
<!--- configuration --->
<cfset sRepUrl = "http://rocket/svn/projects/trunk">
<cfset sUser = "">
<cfset sPasswd = "">
<cfset sFilePath = "test.htm">
<cfset oSVNURL = CreateObject("java", "org.tmatesoft.svn.core.SVNURL").
parseURIEncoded(sRepUrl)>
```

Next we create a repository driver object and authenticate the user:

```
<!--- create an SVNRepository driver object --->
<cfset oSVNRepository = CreateObject("java", "org.tmatesoft.svn.core.
io.SVNRepositoryFactory").create(oSVNURL.parseURIEncoded(sRepUrl))>
<cfset oAuthManager = CreateObject("java", "org.tmatesoft.svn.core.
wc.SVNWCUtil").createDefaultAuthenticationManager(sUser, sPasswd)>
<cfset oSVNRepository.setAuthenticationManager(oAuthManager)>
```

Finally we get the file using the method getFile() and dump the latest revision number for this file:

```
<cfset oSvnFileprops = CreateObject("java", "java.util.HashMap")>
<cfset oBaos = CreateObject("java", "java.io.ByteArrayOutputStream")>
<cfset oSVNRepository.getFile(sFilePath, JavaCast('long', -1), oSvn-
Fileprops, oBaos)>

<cfset latestRevision = oSVNRepository.getLatestRevision()>
            <cfdump var="#latestRevision#">
```

Please consult the JavaSVN documentation, it contains several other excellent Java samples.

Conclusion

The switch from CVS to Subversion was quite easy because of the excellent tools and documentation available. I hope that this article contains some useful information for developers wanting to switch from CVS or some other version control system to Subversion. Maybe it will help to convince your boss that Subversion is the right choice for your team.

Resources

- Subversion Project http://subversion.tigris.org/
- Subclipse Plug-in http://subclipse.tigris.org/
- Tortoise Client http://tortoisesvn.tigris.org/
- SVNImport, Subversive Plug-in www.polarion.org and http://polarion.org/projects.php
- Mere-Moments Guide to installing a Subversion server on Windows http://excastle.com/blog/archive/2005/05/31/1048.aspx1Click Installer http://svn1clicksetup.tigris.org/

Books

- *Version Control with Subversion* http://svnbook.red-bean.com/
- *Pragmatic Version Control Using Subversion* (Mike Mason)

About the Author

Harry Klein is cofounder and CTO at CONTENS Software GmbH, a leading supplier of enterprise content management software. He is a Certified Advanced ColdFusion developer and Microsoft MSCE.

klein@contens.de

Parse RSS del.icio.us Using ColdFusion

Why not do it yourself?

By Jennifer Curtiss

Using RSS as a means to create automatic dynamic content with minimal work fascinates me. Most bloggers probably create feeds on a regular basis – most likely at least a Flickr feed, and possibly del.icio.us.

These provide JavaScript services to parse the feed into your site; however, with the prevalence of server-side scripting, why not have fun and parse the feed into your blog yourself?

It's super-simple, and after looking at a couple of references, I parsed my delicious feed into my about page in mere moments (although I could have just as easily put it in a CF blog template). Before I explain the steps, these following references helped me:

- "Parsing RSS Feeds Using ColdFusion" by Domenic Plouffe (http://coldfusion.sys-con.com/read/issue/219.htm) for the CF tags to work with XML.
- If you are unfamiliar with traversing XML, check "A Really, Really, Really Good Introduction to XML" by Tom Myer (http://www.sitepoint.com/article/really-good-introduction-xml).
- My brilliant brother, who writes at Nazin (http://www.nazin.com/) and taught me nearly everything I know.

I'll explain the process the way I understand it, so if you're new to CF, programming, or what-have-you, hopefully you'll be able to read through this and understand a bit better. I'll explain the quick and dirty way first, then I'll show you how to cache the RSS doc to hasten the process and to ease the load on the servers.

The Quick and Dirty Way

1. Retrieve the document. Be sure to do this first.

```
<cfhttp url="http://del.icio.us/rss/myUserName" method="get">
```

2. Parse the RSS:

```
<cfset objRSS = xmlParse(cfhttp.filecontent)>
```

Obviously, this creates a var named objRSS and fills it with the parsed XML.

3. Output the list. I simply output the titles and links, so my code appears as follows:

```
<ul>
<cfloop from="1" step="+1" to="10" index="counter">
<cfset xTitle = objRSS.xmlRoot.xmlChildren[counter + 1]["title"].
xmltext>
<cfset xLink = objRSS.xmlRoot.xmlChildren[counter + 1]["link"].xmltext>
   <cfoutput>
     <li><a href="#xLink#">#xTitle#</a></li>
   </cfoutput>
</cfloop>
</ul>
```

A brief delineation is in order for those new to RSS and ColdFusion.

```
<cfloop from="1" step="+1" to="10" index="counter">
```

This will loop from one to 10, incrementing by one, thus, 10 iterations.

```
<cfset xTitle = objRSS.xmlRoot.xmlChildren[counter + 1]["title"].xmltext>
```

This sets into a variable named xTitle the text in the "title" tag of our RSS.

- objRSS is the variable holding the parsed RSS.
- xmlRoot, for lack of better explanation, "opens" it up. Really, it's the <RDF> that contains all of the subsequent

data inside.

- xmlChildren[counter + 1] is where the fun begins. I say counter + 1 because xmlChildren[1], with delicious, would open <channel>, which merely contains the boring synopsis of the feed. xmlChildren[2] and so forth are the items that I want – my remembered links. Then I specify which element of the child “item” I want by [“title”], remembering quotations, as XML is case sensitive and ColdFusion is not.
- xmltext is a tostring method, of sorts. It grabs the content of said tag.

```
<cfoutput> <li><a href="#xLink#">#xTitle#</a></li> </cfoutput>
```

We’re outputting what we’ve stored in the variables, with #s surrounding our variable names because we’re mixing variables with attributes and HTML output. Notice that earlier, when mixed with pure ColdFusion (as in the cfloop), these #s were unnecessary, and thus omitted. It still would compile, but we might as well learn it right the first time.

4. If you want to output the entire feed, change to=”10” to to=”#ArrayLen(objRSS.xmlRoot.xmlChildren)#”. Again, we have #s around ArrayLen. This is because it is an attribute. Remember the #s for attributes and HTML output. Otherwise, lose them.

5. Test it out and hopefully it works.

The Better Way

As things are, every time a person visits the page, it grabs the RSS document, parses it, and works with it. If this scripting is on every page on your blog, this is unnecessarily taxing to both your server and wherever your RSS document is hosted. del.icio.us, for one, prefers caching, where we take the doc, parse it, and store it for a defined period of time.

1. If you don’t already have a file on your server called “application.cfm”, create one.

2. Add to application.cfm:

```
<cfapplication name=”myApplicationName”>
```

3. If you worked through the Quick and Dirty Way, erase the lines beginning with cfhttp and the matching parsing cfset. Add to application.cfm:

```
 <cfif Not StructKeyExists(Application,"RefreshDT") or Application.
RefreshDT LT Now()>
    <cfhttp url="http://del.icio.us/rss/myUserName" method="get">
    <cfset Application.objRSS=xmlParse(cfhttp.filecontent)>
 <cfset Application.RefreshDT = DateAdd(“d”,1,Now())>
 </cfif>
 <cfset Variables.objRSS=Application.objRSS>
```

To be clear, this checks to see whether the application var exists or if the RefreshDT time has passed. If it has expired or does not exist, it pulls the RSS document and creates the application, creating a variable named objRSS and parsing the RSS into it. Then RefreshDT is set using the DateAdd() function. The arguments for DateAdd are (“datepart’, units of datepart to add to “date”, “date”). Our example adds one day (“d”) to Now().

Finally, whether the application had to do these steps or not, it makes a local var called objRSS containing the same data as the application var objRSS. This way, it only pulls and parses the feed once for the life of RefreshDT, which this example set to one day.

That was fun, and hopefully it proved to be helpful as well. Big thanks to Josh for teaching me CF in the first place!

About the Author

Jennifer Curtiss is a college student studying software programming and Web development. She volunteers extensively, including hurricane relief work. She supports herself through programming internships and is working toward a position as a Web developer. Visit her blog at www.neatlysliced.com.

jen@neatlysliced.com

CFDJ Advertiser Index

ADVERTISER	URL	PHONE	PAGE
AJAXWORLD	WWW.AJAXSEMINAR.COM	201-802-3022	28-29
CFDYNAMICS	WWW.CFDYNAMICS.COM	866-233-9626	2
CFUNITED	WWW.CFUNITED.COM		15
CFDJ	HTTP://COLDFUSION.SYS-CON.COM/	800-303-5282	43
COMMUNITYMX	WWW.COMMUNITYMX.COM/TRIAL		6
ECLIPSE	HTTP://ECLIPSE.SYS-CON.COM	888-303-5282	33
EDGEWEBHOSTING	EDGEWEBHOSTING.NET	1-866-334-3932	4
HAL HELMS	HALHELMS.COM		47
HOSTMYSITE	WWW.HOSTMYSITE.COM	877-215-4678	19,52
HOTBANANA	HOTBANA.COM/CFDJ	866-296-1803	47
INTERAKT	WWW.INTERAKTONLINE.COM/		51
ITSG		201-802-3021	20
ITVCON.COM	WWW.ITVCON.COM	201-802-3023	21
MACROMEDIA	WWW.MACROMEDIA.COM/GO/8_STUDIO8	415-252-2000	11
PAPERTHIN			17
SOA/EOS CON+ EXPO	EVENTS.SYS-CON.COM	201-802-3023	36-37
VITALSTREAM	WWW.VITALSTREAM.COM	800-254-7554	9

BONUS TRACK ADDED! (FREE $1,295 VALUE!)
"Real-World AJAX" 2-Day Seminar

SOA WebServices, Enterprise OpenSource, + Real-World AJAX!

The #1 i-Technology Educational and

Register at... www.SOAEOSConference.sys-con.com

The **10th International SOA Web Services Edge Conference**, colocated with the First Annual **Enterprise Open Source Conference**, will take place on **June 5-6, 2006** at the historical **Roosevelt Hotel in New York City**. SOA Web Services Edge and Enterprise Open Source Conference together deliver the #1 i-technology educational and networking opportunity of the year.

This year, more than 100 distinguished conference faculty members will present **96 cutting-edge SOA, Web Services and Enterprise Open Source topics** in educational classes, panels, and keynotes in six simultaneous tracks for two information-packed days, plus a bonus track, a **"Real-World AJAX"** two-day seminar, a $1,295 value!

The conference program does not merely present a comprehensive view of all the development and management aspects of integrating a **SOA strategy** and an **Open Source philosophy** into your enterprise, its organizing principle is that delegates will go away from the intense two-day program replete with why-to and how-to knowledge delivered first-hand by industry experts.

According to analyst firm Gartner Group, **by 2008 more than 60 percent of enterprises will use SOA** as the guiding principle when creating mission-critical applications and processes. **"Businesses that ignore the potential of SOA will find themselves outpaced by rivals who improve their agility and transform themselves into new kinds of enterprises,"** says Gartner analyst Yafim Natis.

Meanwhile, the **Open Source revolution** has spread into traditional areas of enterprise IT, with all major vendors and a global group of **Open Source entrepreneurs** developing the technology and delivering the solutions that are changing the way the world does business and the way governments serve their citizens.

SPONSOR BY:

CalAmp COMPOSITE SOFTWARE ROGUE WAVE SOFTWARE SUGARCRM CENTRICCRM metallect WEB AGE

Receive FREE WebCast Archives of all Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all 96 conference sessions. We will mail you the complete content from all the conference sessions in seven convenient DVDs after the live event takes place.

SOA WebServices • OpenSource • AJAX
On-Demand Webcast Archives
DVD

► **This on-demand archives set is sold separately for $995**

Receive 8 Days Worth of Education

FACULTY

Jeffrey Barr
Amazon
As Web Services Evangelist for Amazon.com, Jeff Barr focuses on creating developer awareness for the Amazon software platform. He has a longstanding interest in Web services and programmatic information interchange. Jeff has held development and management positions at KnowNow, eByz, Akopia, and Microsoft, and was a co-founder of Visix Software. Jeff's interests include collecting and organizing news feeds using his site, www.syndic8.com . He holds a Bachelor's Degree in Computer Science from the American University and has done graduate work in Computer Science at the George Washington University.

Israel Hilerio
Microsoft
Israel Hilerio is a program manager at Microsoft in the Windows Workflow Foundation team. He has 15+ years of development experience doing business applications and has a PhD in Computer Science.

Adam Kolawa
Parasoft
Adam Kolawa, Parasoft co-founder and CEO, is considered to be a visionary in his field. In 1983, he came to the United States from Poland to pursue his Ph.D. In 1987, he and a group of fellow graduate students founded Parasoft to create value-added products that could significantly improve the software development process. Kolawa's years of experience with various software development processes has resulted in his unique insight into the high-tech industry and the uncanny ability to successfully identify technology trends. As a result, he has orchestrated the development of numerous successful commercial software products to meet growing industry needs to improve software quality.

Jason Levitt
Yahoo!
Jason Levitt, Technical Evangelist on creating Flash-based Yahoo! Maps applications.

Duane Nickull
Adobe
As senior standards strategist for Adobe Systems, Duane Nickull is responsible for managing Adobe's participation in OASIS and UN/CEFACT, as well as ensuring that Adobe's enterprise solutions support emerging XML standards. Previously Mr. Nickull co-founded Yellow Dragon Software Corporation, a privately held developer of XML messaging and metadata management software, recently acquired by Adobe. Mr. Nickull currently serves as a vice chair of the United Nations Centre for Facilitation of Commerce and Trade (UN/CEFACT) where he oversees the United Nations Electronic Business strategy and architecture.

Bob Pasker
Azul
Bob Pasker is deputy CTO with Azul Systems. He has been designing and developing networking, communications, transaction processing, and database products for 25 years. As one of the founders of WebLogic, the first independent Java company (acquired by BEA Systems in 1998), he was the chief architect of the WebLogic Application Server. Bob has provided technical leadership and management for numerous award-winning technologies, including the TribeLink series of routers and remote access devices, and the TMX transaction processing system. Bob graduated magna cum laude and Phi Beta Kappa from San Francisco State University and holds a Masters degree from Brown University.

Brian Behlendorf
CollabNet
Brian Behlendorf founded CollabNet, with O'Reilly & Associates, in July 1999. The company provides tools and services based on open source methods. Before launching CollabNet, Behlendorf was co-founder and CTO of Organic Online, a Web design and engineering consultancy located in San Francisco. During his five years at Organic, Behlendorf helped create Internet strategies for dozens of Fortune 500 companies. During that time, he co-founded and contributed heavily to the Apache Web Server Project, co-founded and supported the VRML (Virtual Reality Modeling Language) effort, and assisted several IETF working groups, particularly the HTTP standardization effort.

Marc Fleury
JBoss
Born in Paris in 1968, Marc Fleury got his Ph.D in physics from the Ecole Polytechnique in Paris. He started in Sales at Sun Microsystems France and then moved to the US where he worked on early java enablement of SAP at SAPLabs. Marc started the JBoss project in 1999. An ex-Lieutenant in the paratroopers, Marc holds a degree in Mathematics from the Ecole Polytechnique, a master in Theoretical Physics from the Ecole Normale ULM and was a visiting scientist at MIT during his thesis. Marc's research interest focuses on aspect oriented middleware.

Andy Astor
EnterpriseDB
Andy is President and CEO EnterpriseDB, the world's leading enterprise-class, open source database company. Previously, Andy was vice president webMethods, leading the company's open source, standards, and Web services agendas. Andy was elected twice to the Board of Directors of the Web Services Interoperability Organization (WS-I), and led WS-I's marketing efforts. Prior to joining webMethods, Andy was vice president at D&B, where he led worldwide development of all on-line products. His work at D&B included the development and launch of one of the earliest commercial Web services.

Mike Milinkovich
Eclipse.org
Mike Milinkovich has held key management positions at Oracle, WebGain, The Object People, and Object Technology International Inc. (which subsequently became a wholly-owned subsidiary of IBM), assuming responsibility for development, product management, marketing, strategic planning, finance and business development. Mike earned his MS degree in information and systems sciences and a bachelor of commerce degree from Carleton University in Ottawa, Canada.

Peter Yared
ActiveGrid
Peter Yared is the founder and CEO of ActiveGrid. Most recently, he was CTO of Sun Microsystems's Liberty Network Identity initiative. Mr. Yared was also CTO of Sun Microsystems Application Server Division. Before its acquisition by Sun, Mr. Yared served as CTO of NetDynamics, which pioneered the then-leading J2EE application server. Earlier, Mr. Yared was founder and CEO of JRad Technologies, an enterprise Java company acquired by NetDynamics. Additionally, Mr. Yared was Chief Architect of client/server products at object-oriented tool maker Prograph International and the architect of several mission-critical systems deployed by U.S. government agencies and the GED Testing Service.

David Temkin
Laszlo
David Temkin is Chief Technology Officer of Laszlo Systems, Inc. In this role, he has positioned the company to become the next technology standard for rich Internet applications. Under his direction, Laszlo developed its patent-pending open-source product suite and extended operations to both coasts of the United States. Before founding Laszlo, Temkin was senior director of engineering at Excite@Home where he led a team of 55 engineers, designers and technical writers responsible for developing the company's consumer software. Prior to Excite@Home, Temkin was an engineering manager in the Newton division at Apple Computer and developed enterprise software at EDS.

Kevin Hakman
TIBCO
Kevin Hakman is Co-founder, TIBCO General Interface, TIBCO Software Inc. Prior to TIBCO General Interface, he was the co-founder of Versalent Inc. a leading provider of enterprise client technology. Prior to Versalent, he founded a series of successful emerging Internet technology and ecommerce ventures. He has also written for eBusiness Journal and HotWired.

Coach Wei
Nexaweb
Coach Wei currently serves as CTO for Nexaweb, which develops the leading XML-based rich client technology platform for building and deploying Enterprise Internet Applications. Previously, he played a key role at EMC Corporation in the development of a new generation of storage network management software. Coach is a graduate from MIT, holds several patents, and is an industry advocate for the proliferation of open standards.

Luis Derechin
JackBe
Luis Derechin is CEO and Co-Founder of JackBe. Mr. Derechin has over 12 years of entrepreneurial and management experience. He has been part of the founding team of successful startups, including a catalogue retail company that achieved $15M in sales.

Jouk Pleiter
Backbase
Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur: he founded three successful Software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market.

Networking Opportunity of the Year!

SOA WebServices Edge 10th International conference

ENTERPRISE> OPENSOURCE CONFERENCE

REAL-WORLD AJAX SEMINAR

The Roosevelt Hotel

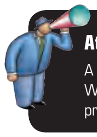
JUNE 5-6, 2006
Roosevelt Hotel / New York, NY

For Sponsorship Information ▸ Call 201-802-3020

Discounted Price
Incl. Golden Pass to Both Conferences* $1,495

Conference Price (On June 5-6, 2006)
Incl. Golden Pass to Both Conferences* $1,695

*BONUS: "Real-World AJAX" Two-Day Seminar!



Attention Sponsors:
A Forum will be available to display leading Web services, OpenSource, and AJAX products, services, and solutions.

TOPICS INCLUDE...

SOA Web Services Edge:
- *Transitioning Successfully to SOA*
- *ebXML*
- *Orchestration*
- *The Business Case for SOA*
- *Interop & Standards*
- *Web Services Management*
- *Messaging Buses and SOA*
- *SOBAs (Service-Oriented Business Apps)*
- *Delivering ROI with SOA*
- *Java & XML Web Services*
- *Security*
- *Systems Integration*
- *Sarbanes-Oxley*
- *Business Process Management*

Enterprise Open Source:
- *Open Source Licenses*
- *Open Source & E-Mail*
- *Databases*
- *ROI Case Studies*
- *Open Source ERP & CRM*
- *Open-Source SIP*
- *Testing*
- *Grid Computing*
- *Open Source Middleware*
- *LAMP Technologies*
- *Professional Open Source*
- *Open Source on the Desktop*
- *Open Source & Sarbanes-Oxley*
- *IP Management*

AJAX User Experience:
- *Rich Internet Applications*
- *Flash-AJAX Integration*
- *Developing Interfaces*
- *AJAX on Rails*
- *Best Practices*
- *Enterprise AJAX*
- *Technology Demos*
- *Lessons from the Frontline*
- *Bringing Desktop Apps to the Web*
- *AJAX Power Panel*
- *and more...*

SYS-CON EVENTS www.EVENTS.sys-con.com

Packed into 2 Information-Filled Days!

ColdFusion U

For more information go to...

U.S.

Alabama
Huntsville, AL CFUG
www.nacfug.com

Arizona
Phoenix, AZ CFUG
www.azcfug.com

California
Bay Area CFUG
www.bacfug.net

California
Sacramento, CA CFUG
http://www.saccfug.com/

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

Colorado
Denver CFUG
http://www.denvercfug.org/

Connecticut
SW CT CFUG
http://www.cfugitives.com/

Connecticut
Hartford CFUG
http://www.ctmug.com/

Delaware
Wilmington CFUG
http://www.bvcfug.org/

Florida
Jacksonville CFUG
http://www.jaxfusion.org/

Florida
South Florida CFUG
www.cfug-sfl.org

Georgia
Atlanta, GA CFUG
www.acfug.org

Illinois
Chicago CFUG
http://www.cccfug.org

Indiana
Indianapolis, IN CFUG
www.hoosierfusion.com

Louisiana
Lafayette, LA MMUG
http://www.acadianammug.org/

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

Maryland
Maryland CFUG
www.cfug-md.org

Massachusetts
Boston CFUG
http://bostoncfug.org/

Massachusetts
Online CFUG
http://coldfusion.meetup.com/17/

Michigan
Detroit CFUG
http://www.detcfug.org/

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

Minnesota
Southeastern MN CFUG
http://www.bittercoldfusion.com

Minnesota
Twin Cities CFUG
www.colderfusion.com

Missouri
Kansas City, MO CFUG
www.kcfusion.org

Nebraska
Omaha, NE CFUG
www.necfug.com

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

New Hampshire
UNH CFUG
http://unhce.unh.edu/blogs/mmug/

New York
Rochester, NY CFUG
http://rcfug.org/

New York
Albany, NY CFUG
www.anycfug.org

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

New York
Syracuse, NY CFUG
www.cfugcny.org

North Carolina
Raleigh, NC CFUG
http://tacfug.org/

Ohio
Cleveland CFUG
http://www.clevelandcfug.org

Oregon
Portland, OR CFUG
www.pdxcfug.org

Pennsylvania
Central Penn CFUG
www.centralpenncfug.org

Pennsylvania
Philadelphia, PA CFUG
http://www.phillycfug.org/

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

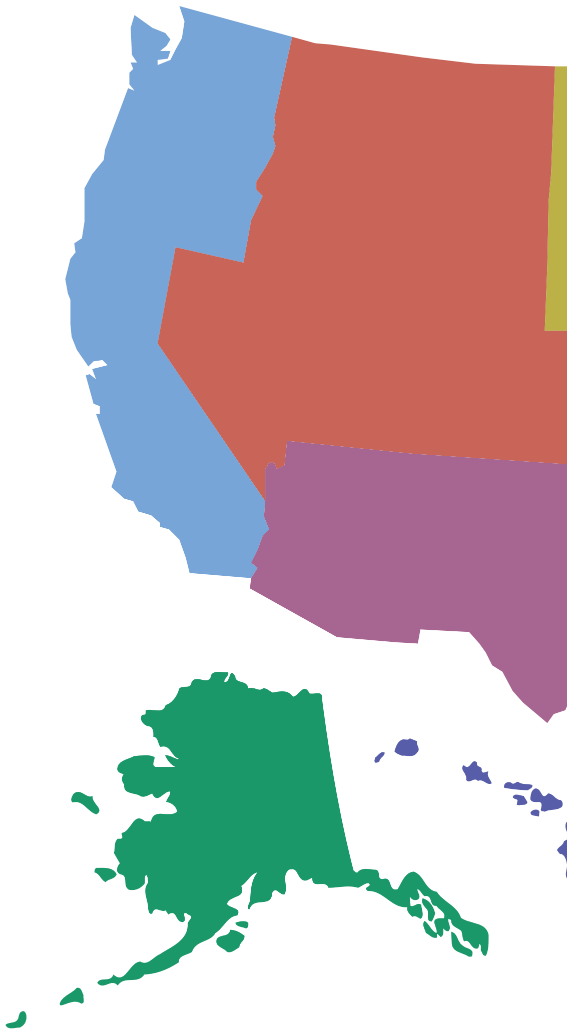
Tennessee
Nashville, TN CFUG
http://www.ncfug.com

Tennessee
Memphis, TN CFUG
http://mmug.mind-over-data.com

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

Texas
Dallas, TX CFUG
www.dfwcfug.org/

Texas
Houston Area CFUG
http://www.houcfug.org



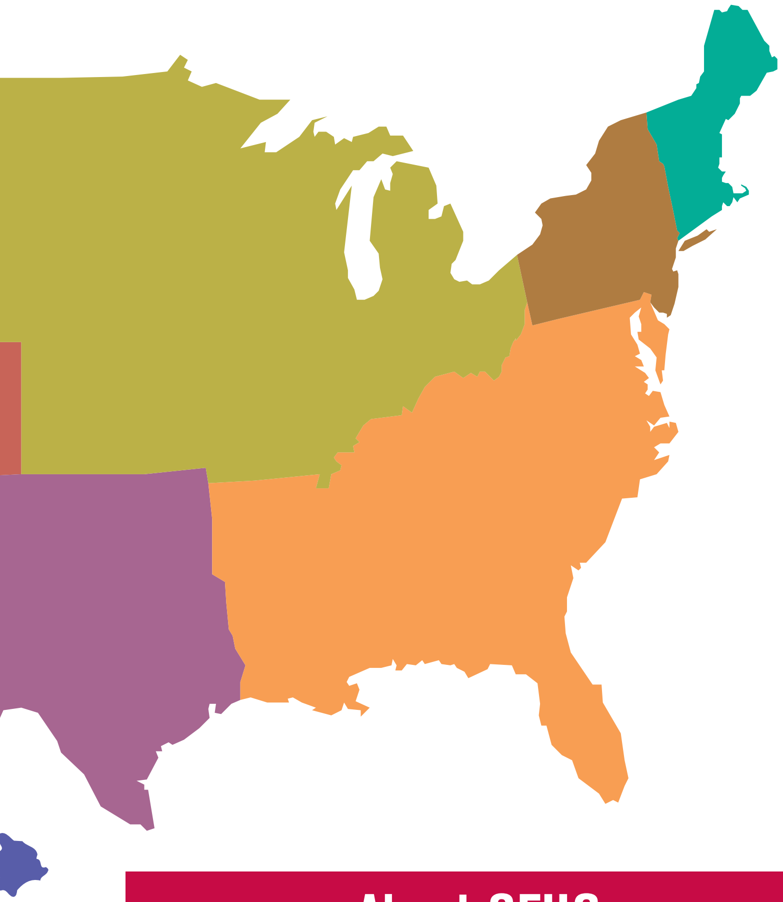
Utah
Salt Lake City, UT CFUG
www.slcfug.org

Virginia
Charlottesville CFUG
http://indorgs.virginia.edu/cfug/

Washington
Seattle CFUG
http://www.seattlecfug.com/

Jser Groups

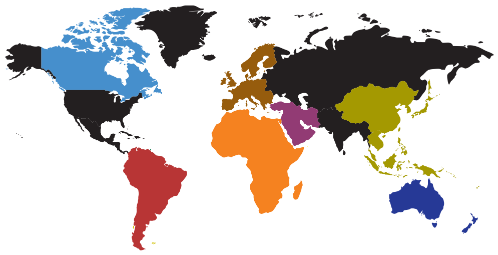
http://www.macromedia.com/cfusion/usergroups



About CFUGs

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

INTERNATIONAL



Australia
ACT CFUG
http://www.actcfug.com

Australia
Queensland CFUG
http://qld.cfug.org.au/

Australia
Victoria CFUG
http://www.cfcentral.com.au

Australia
Western Australia CFUG
http://www.cfugwa.com/

Canada
Kingston, ON CFUG
www.kcfug.org

Canada
Toronto, ON CFUG
www.cfugtoronto.org

Germany
Central Europe CFUG
www.cfug.de

Italy
Italy CFUG
http://www.cfmentor.com

New Zealand
Auckland CFUG
http://www.cfug.co.nz/

Poland
Polish CFUG
http://www.cfml.pl

Scotland
Scottish CFUG
www.scottishcfug.com

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

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

Spain
Spanish CFUG
http://www.cfugspain.org

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

Switzerland
Swiss CFUG
http://www.swisscfug.org

Turkey
Turkey CFUG
www.cftr.net

United Kingdom
UK CFUG
www.ukcfug.org

How ColdFusion MX 7 Made Me a "Hero" at the Office...

Making valuable and useful business information easily accessible

By Michael Markowski

Let me begin by saying that I don't consider myself a ColdFusion expert. In fact, I didn't even write my first ColdFusion application until 2004. Nevertheless, ColdFusion MX 7 still has made me a "hero" (as Simon would put it) at the office, and this is how it happened.

I am a Webmaster for the Air Protection Division, Environmental Protection Agency (EPA), Region 3, in Philadelphia. In June 2004 I wrote my first ColdFusion application in ColdFusion 5, which was the only version of the ColdFusion server available to me then. A Microsoft Access database served as the back end for this application. The application, which was deployed on my organization's intranet, was named the "Air Salient Issues ColdFusion Application (ASICA)." I know, I know, that's a very long, complicated name for a ColdFusion application, but this application's purpose was simple: store and facilitate access to my division's "salient issues." Salient issues are brief (one page) text descriptions of our most significant program activities, issues, events, and accomplishments. They are written by Region 3 employees on a weekly basis and are distributed, via e-mail, throughout the office to keep everyone (especially management) informed about the status of the work that's being done by every division, including the Air Protection Division (where I work).

The process of disseminating the salient issues to employees via e-mail was a very inefficient and cumbersome one. Worse, there was no way to search the salient issues to find a specific one when the need arose, and the need to search the salient issues arose quite often. The salient issues were written in Microsoft Word (or WordPerfect) and the documents were then placed into folders on a shared network drive. Therefore, if someone wanted to find a salient issue, they would have to manually search through the documents stored on the network. This was pretty much a hit or miss affair and, needless to say, very frustrating for employees. Consequently, most employees quickly forgot about their salient issues once they had been written and "published" via e-mail. My supervisor (David Arnold) and I decided that there had to be a better way of storing, organizing, and disseminating this information, so I began working on a ColdFusion application to do just that.

I built the ASICA as a standard "drill-down" search application in which users could search for salient issues by author, publication date, and/or keyword(s). To view or print the details of a salient issue, a user simply clicked on a link to load a "details" template that retrieved and displayed all of the pertinent information for that salient issue: Title, Publication Date, Summary, Author, Confidential Status, and Comments. A "Recent Salient Issues" page was deployed that listed the most recent salient issues published during the previous two weeks. This option was provided for managers who just wanted to get a quick listing of the most recent salient issues without having to perform a search.

I had no way of knowing this at the time, but the ASICA was destined to be the first in a succession of highly successful and well-regarded ColdFusion intranet applications that would place our most valuable and useful business information right at the fingertips of the people who needed it the most: our employees (all 1,000 of us).

As my programming skills and confidence grew, I became bolder with my code and began experimenting with new

ColdFusion features and tags. Well, at least they were "new" to me since I was a beginning developer at the time. I implemented a "Next N Records" interface and query results caching to limit the amount of information displayed on each page and to increase performance. Due to budget constraints, I was stuck with Microsoft Access as the data source and therefore needed to minimize database interaction, otherwise, frequent database calls would affect the performance of the entire application. In the "Keyword Search" portion of the application, I employed <CFLOOP> and list variables to break apart multiple keywords entered by users performing searches; this code then scanned the Title and Summary fields of the database for a match on one or more of the keywords. This produced a surprisingly powerful and flexible search engine for my application that could easily handle multiple keywords and phrases. I also utilized the <CFCONTENT> tag to add a "print version" template to the application so that users could generate a Microsoft Word version of any salient issue. It strikes me as somewhat ironic that the "print version" feature produces a Word document since the proliferation of Word documents on our network drive and our inability to organize and manage them was the reason for developing this application in the first place.

The ASICA rapidly evolved into an indispensable information storage, search, and retrieval tool that could be used by anyone who knew their way around a Web browser. Of course, that included pretty much everyone in our office. New features and search capabilities were added and, eventually, word of the application's usefulness spread throughout the office. Today, the ASICA is used extensively for everything from determining the status of an enforcement case to highlighting an upcoming public outreach program. Another indication of the application's extraordinary success was upper management's regular use of the application and their willingness to provide me with the time and resources that I needed to build it and keep it running smoothly.

Soon after the success of the ASICA, I began receiving requests from other division managers to build similar applications for them so that they too could place their salient issues on the intranet and make them instantly accessible to everyone in the office. In response to these requests and to preserve my own sanity, I created a "generic" salient issues application with the help of some custom tags and user-defined functions (UDFs). This "generic" application could be customized for use by any division or program office, and could be deployed in a matter of minutes simply by resetting a few variables in the Application.cfm template, e.g., <CFSET divname = "Hazardous Site Cleanup Division"> and <CFSET datasource = "hscdsalients">. In other words, the only things different in these new "clone" applications were the division name, logo images, data source, and other information specific to the division for which the application was being created; all other code remained exactly the same as it had been for the ASICA. This strategy really paid off because it enabled me to create and deploy salient issues applications for other divisions in a matter of days instead of months. So, I learned from firsthand experience that they don't call ColdFusion a "Rapid Web Application Development" technology for nothing.

Three more salient issues applications followed the ASICA onto our agency intranet. I created, customized, and deployed each of these applications that would subsequently be used by other divisions. When these new applications were deployed on our intranet, the number of salient issues that we had made available online (via ColdFusion) doubled. Today, there are over 4,500 salient issues from four separate divisions available on our intranet, courtesy of ColdFusion MX 7.

With respect to data entry and application security, I used session variables and provided user names/passwords to authorized staff persons in each division to limit access to the administrative templates in each application. Every week, one or two authorized employees from each division enter their salient issues for that week into their respective database using the administrative templates. Since the text of the salient issues can be copied and pasted directly from the Microsoft Word (or WordPerfect) documents into the online ColdFusion forms, entering the weekly salient issues typically takes less than 20 minutes per application. Today, over 2,000 salient issues (dating back eight years) have been entered into the ASICA alone. As we have discovered for ourselves, 20 minutes of data entry time per week is a small price to pay for having mission-critical information online and organized so that it can be accessed by anyone in a matter of seconds. Indeed, I lost count of all the “thank you” e-mails that I had received from appreciative managers and staff who were finally able to find the exact information they were looking for simply by using one of my salient issues applications.

After the resounding (and somewhat unexpected) success of the salient issues applications, my next project was to build a ColdFusion application that would serve as a comprehensive inventory of clean diesel fuel projects in the U.S. This new application was named the “National Clean Diesel Database” (NCDD) and it has become an agency-wide intranet tool that EPA employees can access and use from any of our 10 regional offices (located in major cities throughout the U.S.) and headquarters. I used ColdFusion’s <CFCHART> tag to generate attractive and informative bar and pie charts that neatly summarize the voluminous clean diesel project data (project cost, projected emission reductions, number of projects and vehicles per region, and so forth). I also used the <CFFUNCTION> tag to write UDFs that encapsulated redundant formatting and processing to further streamline my code and make it reusable. Some of the functions that I created while working on the NCDD were later used in my other ColdFusion applications. In the NCDD, I made frequent and effective use of the <CFOUTPUT> tag’s “group” attribute to create grouped output of query results by region. I also used the “Query of Queries” feature to keep database interaction to an absolute minimum. I ended up receiving an award in 2005 for my work on the NCDD.

In early 2006 I added the <CFDOCUMENT> tag to my code arsenal so that users could easily retrieve “printable” (and portable) versions of application pages in Adobe Acrobat (PDF) format. I think the ability to dynamically render Web pages in PDF (or Flashpaper) format is one of the most exciting and truly useful new features in ColdFusion MX 7. In addition to being a tremendous time-saver, it has allowed me to accomplish certain things that would have otherwise been impossible. For example, I’m currently using the <CFSCHEDULE> and <CFDOCUMENT> tags to automatically publish static pages in both HTML and PDF format. PDF and/or HTML versions of Web documents published via our intranet can be moved to a public Web site that has no ColdFusion support. In other words, with ColdFusion MX 7 I am able to use an intranet application to generate documents for use on a static (non-ColdFusion) Web site. This is a critically important feature for me because having an actual ColdFusion application on my agency’s public Web site is out of the question due to the high cost. The ability to produce static PDF (and HTML) documents from intranet database information for use on our public site is an effective, no-cost solution to this problem; this technique produces up to date, high-quality documents that can be deployed immediately on our public Web site. For examples of static Web documents that I have created using an intranet ColdFusion application, see http://www.epa.gov/reg3artd/permitting/permits1c.htm#lists and http://www.epa.gov/reg3artd/permitting/petitions2.htm.

Recently, while experimenting with some code, I found that I could use ColdFusion as a “JavaScript code generator.” This technique takes the publication of static Web pages one step further by dynamically rendering JavaScript code in a browser; the code is then copied and pasted (by me) into static Web pages for use on the EPA’s public Web site. Just as ColdFusion MX 7 can be used to create static HTML Web pages and PDFs, it can also be used to generate JavaScript code for use in static Web pages. One example of this technique is my division’s “Title V Air Permits Search” Web page at http://www.epa.gov/reg3artd/permitting/t5permitsearch.htm. This is a standard HTML Web page with a twist: it utilizes an external JavaScript file (permitdata.js) that contains over 1,500 lines of JavaScript code. This JavaScript source code is effortlessly generated in seconds (by me) using a ColdFusion template on our intranet that queries our internal Air Permits Tracking System (Oracle) database and then renders, in the browser, all of the JavaScript code that’s needed by the Title V Air Permits Search Web page. It’s then a simple matter of copying and pasting the generated JavaScript from the browser to a text file and naming the file with a “.js” extension. I have also used UDFs to create dynamic JavaScript links (for lack of a better term) within ColdFusion pages that, when clicked, generate descriptive popup messages using the JavaScript “alert” method. These special popup message links are used throughout my ColdFusion applications to describe acronyms that appear on certain pages. These links make the applications much more user-friendly since users can click on them to quickly find out what the various acronyms mean.

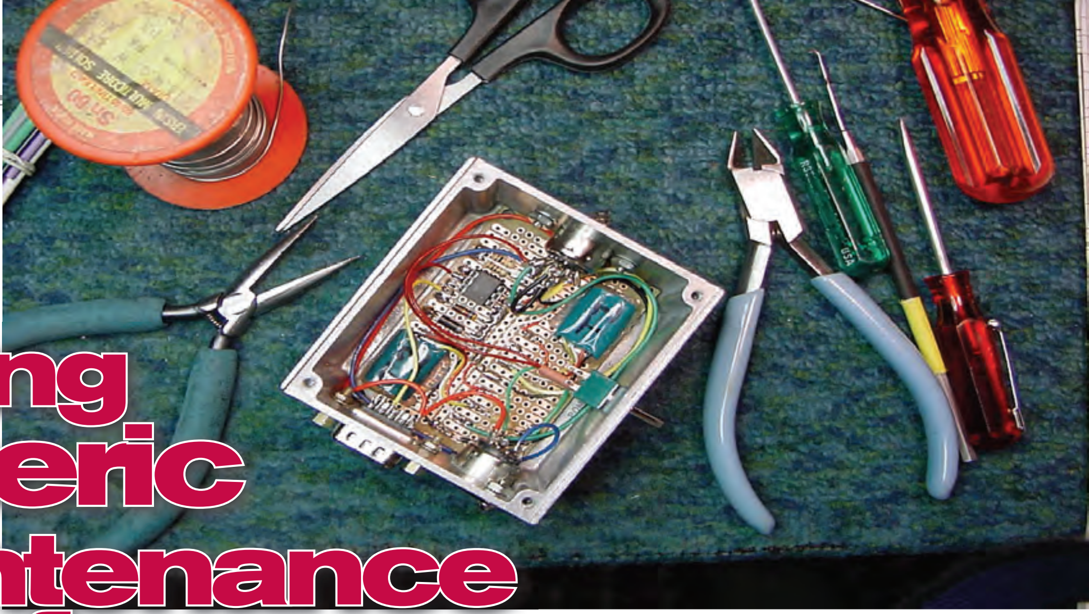
Some things I’m working on for the future include the use of Flash forms for data entry; and the <CFHTTP>, <CFFILE> and <CFFTP> tags for file manipulation, uploads to our public Web server, and interaction with other federal government Web sites. I have also begun experimenting with the <CFTRY> and <CFCATCH> tags to catch and programmatically handle database errors when they occur in my applications. I’ve been using <CFMAIL> to send myself an e-mail whenever our database server encounters a problem; I have found this to be a very useful technique since I am notified immediately whenever the server goes down. I can then investigate and attempt to correct the problem.

By using ColdFusion MX 7, we have dramatically improved access to our most critical business information. With the salient issues applications, we transformed an archaic, inefficient business practice into a powerful data management tool that can be used by all Region 3 employees. Using ColdFusion MX 7, we were able to overcome Web publication obstacles that had prevented us from providing important environmental information to the public on the EPA Web site. I can’t even imagine the exciting and productive things that we’ll be doing with ColdFusion MX 7 (or Scorpio) a year or two from now. The future looks very bright indeed for ColdFusion developers, and I am glad to be a part of that future. By the way, what are these Event Gateway things I keep reading about in CFDJ? Maybe it’s time to start experimenting with some code again…

About the Author

Michael Markowski works for the Air Protection Division at the Environmental Protection Agency and is a Macromedia/Adobe Certified Professional.

markowski.mike@epa.gov



Building Generic Maintenance Interfaces

Sparing yourself an awful lot of tedious work

By Craig Drabik

When the March 2006 issue of CFDJ arrived, I had just begun working on the maintenance interfaces for the support tables of a new system. There are many of these tables, and implementing the associated maintenance routines looked set to consume a lot of time.

I had just finished implementing DAOs for the first of the support forms when I came across Nic Tunney's article on ObjectBreeze. ObjectBreeze was going to save me a lot of time on implementation – and spare me an awful lot of tedious work building DAOs for each table.

ObjectBreeze saved me so much time it got me to thinking… Would it be possible to implement the user interfaces generically, in the same way that ObjectBreeze implements our data model? And if we have a consistent, generic interface to access a data model, can we leverage that interface in the view and controller tiers too? The controller tier seems simple enough – for a simple maintenance form it merely needs to actuate the CRUD methods of the DAO. As far as the view tier goes, if it knew what the data model looked like, it shouldn't be difficult. The potential benefits would be enormous – I could concentrate my time on the difficult tasks of aggregating data and implementing business logic without worrying about data access or user interfaces.

Building the Basics

For this article we're going to put together a fictitious system for a hypothetical racing club. This club needs to track the drivers, teams, and cars that enter club races in various classes. Figure 1 details our data model. We have tables for each object – drivers, cars, teams, and classes. A car belongs to a single team, races in a single class, and is driven by a single driver. Drivers can drive any number of cars in different classes for different teams. Teams can own any number of cars, and classes can contain any number of cars.

Let's start with the drivers. They're one of the simplest objects because they don't contain any references to other objects (tables). The club is interested in tracking a driver's first and last name and his nationality. First we'll set up our controller (drivers.cfm). This will be the only file that the browser requests when manipulating a driver object. Our controller is essentially a switch statement that interprets a form parameter called controlMethod. Our controller supports the standard CRUD operations – create (add), read (edit), update, and delete operations, plus a list operation. Each operation involves a database operation and a user interface to render the results. The list operation will retrieve all of the drivers and display them in a grid, and the edit operation retrieves one driver and outs his details in a form for editing. Create is similar to edit except that it creates a new driver record. Update applies the results of a create or edit operation and delete removes a driver's data from the database. ObjectBreeze is used to handle the data model. (The

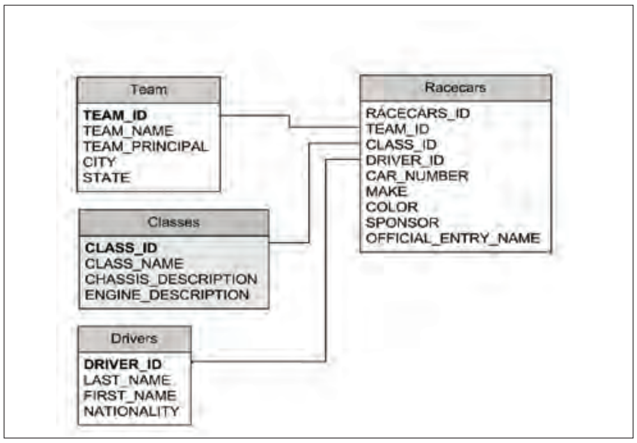


Figure 1

source code for this article can be downloaded from the online version of this article at http://coldfusion.sys-con.com.)

Each method that has to present data to the user also contains a call to the custom tag responsible for building the view. There's a tag for displaying lists (grid) and for editing a single record (detail). Since these templates are going to be generic, I've put them in a directory called genericUI. A CFMODULE call is used to invoke the generic code.

The Views

The maintenance for the drivers' table is a typical table maintenance application – a list of items and a detail form for editing an individual item. The generic grid template will display our list of items with controls to enable adding and deleting records, and a control to select an item for editing. To do this we have to tell the grid control which fields we want displayed, what the captions for those fields should be, and the data to be displayed. Using that information it's a pretty simple matter of looping over the list of captions to create the grid header. A two-dimensional loop displays the requested fields for each row of data.

The detail form is similar to the grid. For each field in the data model, we generate a table row containing a label and a textbox to edit the corresponding field. Arguments to our custom tag are similar to the grid except that we're passing a single data object.

See Listing 1 for the code we've built so far.

So how does it work? Not bad for a first stab. We've got a grid containing a list of drivers. We've got links for adding, editing, and deleting drivers. It definitely isn't perfect – there are a couple shortcomings we'll want to address. The first of these shortcomings is that the controller template is specific to the drivers' table. Before we can really reuse our code we'll want to build a generic controller template.

The Controller

Fortunately the task of converting the controller template is fairly simple. We want to call the controller as a custom tag, so instead of looking at form variables for our parameters we're going to be looking at attributes. We'll also have to tell the controller what fields we want displayed, and what their associated captions are. We'll also pass the form scope into the controller. Then we modify our custom tag calls to use these attributes instead of our hard-coded field lists. Our generic controller gets a new name – genericmasterdetail.cfm – and it moves into the genericUI folder.

The page that the browser invokes now becomes very simple – just a call to genericmasterdetail with the proper parameters. When we test our new drivers.cfm page in the browser, everything looks and functions exactly as before. Now we're ready to really leverage what we've built. The classes and team tables are both simple tables, similar in structure to the drivers' table. We make copies of the drivers.cfm file, change the parameters we pass to genericmasterdetail, and presto – fully functional maintenance forms for drivers, teams, and classes. I've done a little preprocessing to make things a bit neater. For example, I've included logic in classes.cfm to instruct our generic user interface to display only the class name during the list operation and all three fields during add or edit operations. See Listing 2 for the updated code.

Object Composition

The objects we've worked with so far are very simple in that all of the data they require is in one table. Usually, we're not that lucky; we often need data from multiple tables. ObjectBreeze supports models like these through composite objects. ObjectBreeze gives us a nice set of features for working with composite objects – we use the containsOne() or containsMany() methods to register the relationships between our tables with ObjectBreeze and it takes care of our database interactions for us.

Introducing composition into our generic user interfaces poses two problems. The first is how to tell the controller code to create a compound data object for us. The second is how to deal with this data in the user interface.

The second question is actually the easier of the two to answer. If we have a "contains one" relationship, such as the one between a team and a car, we can use a list box control to update that relationship. If we have a "contains many" relationship, we can use a cross-select to select the related items. Of course we'll have to know the relationship between the parent and children to know how to handle each item, but we have to describe that relationship to the controller to get the data objects configured properly in the first place.

XML to the Rescue

XML is the perfect choice for describing data to our generic user interfaces. It also makes it easy to enhance our code without making changes to the way it gets called. We're going to work with the racecars table for our first example. Cars are driven by a driver, entered in a class, and owned by a team. That is to say that there is a "contains one" relationship between our racecars object and the team, classes, and drivers objects.

The XML document that describes the racecars' object to genericmasterdetail is in Listing 3. (Listings 3 and 4 can be downloaded from http://coldfusion.sys-con.com.) The document describes the data model (datamodel element and children), the grid view (grid element), and the detail view (detail element). We'll feed this document into our genericmasterdetail custom tag and from there into the grid and detail custom tags. These tags have changed in several different ways. First, the arguments have changed. The form scope still gets passed into genericmasterdetail, grid, and detail but the configuration document has replaced the other arguments. All the references to the old arguments were updated to look at the configuration document instead. A new custom tag – uiformelements – has been created to handle the display of the individual form elements. Finally, two helper functions have been created to help us manage composite data access objects.

Listing 3 contains the racecars' XML document and the helper functions configureDAO() and getProperty().

All of these upgrades buy us a lot of functionality. We've moved quite a ways beyond simple forms. We can now add, update, and delete records in a wide variety of formats with just a few lines of XML.

The Generic Application

One glance at the code in racecars.cfm makes it obvious that there's more advantage to be gained through generic code. The only thing different in all our browser-invoked templates is the XML that gets passed into genericmasterdetail. We could read the XML documents into Cold Fusion using a form or URL parameter. A better idea would be to represent the entire application in one

XML document. This approach has several advantages. First, we only have to look in one place for the XML datagrams that define our forms, making maintenance easier. Secondly, with all of the application metadata in the same place we've introduce a degree of self-documentation into our applications. The entire structure of the application is laid out in the XML document. We can draw on that application metadata to enhance the generic application further by providing navigation controls based on the application metadata in our XML document.

To construct an XML document representing the entire application, I encapsulate each form's XML datagram inside of a dataform element. This element contains attributes for naming the form and attaching a caption to be displayed to the user. I've also added an element to point the application to a page to display as a home page. On the ColdFusion end of things, a new index.cfm file is constructed. The index.cfm file contains the XML describing the application and code to create a visual shell for the application. The shell displays the application's name, followed by navigation controls on the left and our generic forms on the right. The application's name is taken from the genericui element's title attribute. We build the navigation based on the application xml – each item in the array of dataform elements gets a link. We use the dataform element's caption attribute for the link text, and the name attribute gets used to decide which dataform element is passed into genericmasterdetail. Finally, a form and JavaScript are used to transmit the user's navigation selection back to index.cfm for processing.

Fire up a browser and surf to the new index page and the power of our generic application solution is immediately apparent. We've basically implemented maintenance for a system of four tables in 65 lines of XML. The interface and functionality might be a bit basic, but if you think about data-driven Web sites there are usually a bunch of support tables that either get updated via our database tools or consume valuable programming resources to implement maintenance interfaces. A generic structure can save you time, so you can devote your resources towards building business logic or a more impressive customer-facing site – or maybe you just might save enough time to take a vacation!

To learn more about objectBreeze see http://www.objectBreeze.com.

About the Author

Craig Drabik is a senior programmer with the State University of New York at Buffalo. He is a Macromedia certified ColdFusion developer who has been working in ColdFusion since version 4.51.

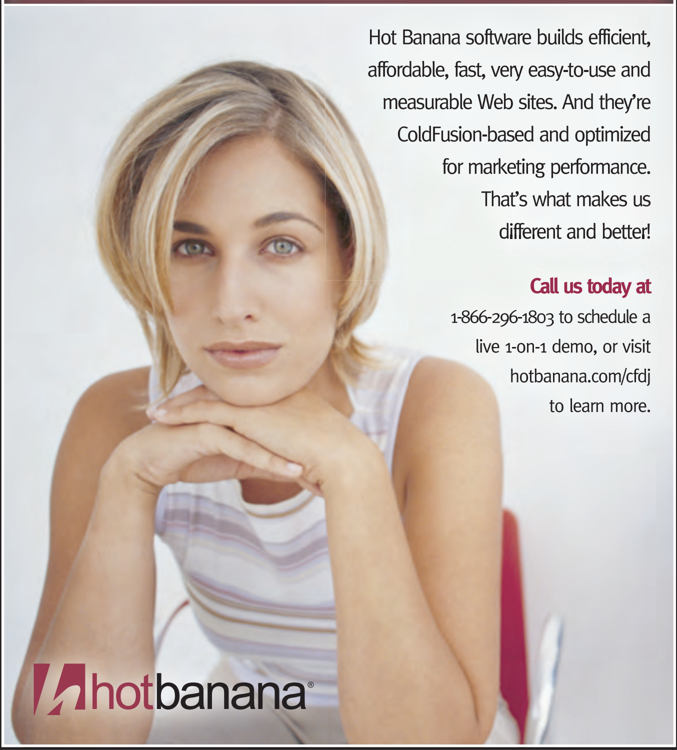
cjdrabik@buffalo.edu

Listing 1: drivers.cfm

```
<cfparam name="FORM.controlMethod" default="list">

<cfswitch expression="#FORM.controlMethod#">
  <cfcase value="list">
     <cfset oDrivers = APPLICATION.objectBreeze.list("drivers",
"last_name, first_name") />
     <cfmodule  template="genericUI/grid.cfm"
                fields="first_name,last_name,nationality"
                fieldNames="First Name,Last Name,Nationality"
                dao="#oDrivers#">

  </cfcase>

  <cfcase value="edit">
     <cfscript>
        oDriver = APPLICATION.objectBreeze.objectCreate("drivers");
        oDriver.read(FORM.pk);
     </cfscript>
     <cfmodule  template="genericUI/detail.cfm"
                fields="first_name,last_name,nationality"
                fieldNames="First Name,Last Name,Nationality"
                dao="#oDriver#">
  </cfcase>
  <cfcase value="add">

     <cfscript>
        oDriver = APPLICATION.objectBreeze.objectCreate("drivers");
     </cfscript>
     <cfmodule  template="genericUI/detail.cfm"
                fields="first_name,last_name,nationality"
                fieldNames="First Name,Last Name,Nationality"
                dao="#oDriver#">
  </cfcase>

  <cfcase value="update">
     <cfscript>
        oDriver = APPLICATION.objectBreeze.objectCreate("drivers");
        if (isDefined("FORM.pk") AND FORM.pk NEQ "")
        {
           oDriver.read(FORM.pk);
        }

        for (i = 1; i LTE listLen(FORM.fieldnames); i = i + 1)
        {
           if (NOT listFindNoCase("controlMethod,pk", listGetAt(FORM.
fieldnames, i)))
           {
              oDriver.setProperty(listGetAt(FORM.fieldnames, i),
FORM[listGetAt(FORM.fieldnames, i)]);
           }
        }
        oDriver.commit();
     </cfscript>
     <cfoutput>
     <form name="controllerForm" action="#CGI.SCRIPT_NAME#"
method="post">
        <input type="hidden" name="controlMethod" value="list" />
        <input type="text" name="garbage" />
     </form>
     <script language="javascript">
        document.forms.controllerForm.submit();

     </script>
     </cfoutput>
  </cfcase>

  <cfcase value="delete">
     <cfscript>
        oDriver = APPLICATION.objectBreeze.objectCreate("drivers");
        oDriver.read(FORM.pk);
        oDriver.delete();
     </cfscript>
     <cfoutput>
     <form name="controllerForm" action="#CGI.SCRIPT_NAME#"
method="post">
        <input type="hidden" name="controlMethod" value="list" />
        <input type="text" name="garbage" />
     </form>
     <script language="javascript">
        document.forms.controllerForm.submit();
```

Get more with Hot Banana!

Web Content Management + Active Marketing



Hot Banana software builds efficient, affordable, fast, very easy-to-use and measurable Web sites. And they're ColdFusion-based and optimized for marketing performance. That's what makes us different and better!

Call us today at 1-866-296-1803 to schedule a live 1-on-1 demo, or visit hotbanana.com/cfdj to learn more.

hotbanana®

"I was totally intimidated by Java, but I knew I had to learn it. Your class taught me what I honestly thought I couldn't be taught." - Sharon T

Java for ColdFusion Programmers?

Java for ColdFusion Programmers, the five-day Hal Helms, Inc. training class is designed for ColdFusion programmers; no previous Java experience is needed. You'll learn how to think in objects and program in Java.

For class information and registration, come to halhelms.com.

REPRINT IT!

Once you're in it...

...reprint it!

- ColdFusion Developer's Journal
- Java Developer's Journal
- Web Services Developer's Journal
- Wireless Business & Technology
- XML Journal
- PowerBuilder Developer's Journal
- .NET Developer's Journal

Contact Dorothy Gil
201 802-3024
dorothy@sys-con.com

REprints SYS-CON MEDIA

```
      </script>
      </cfoutput>
  </cfcase>

</cfswitch>


grid.cfm

<cfswitch expression="#thisTag.executionMode#">
  <cfcase value="start">
      <cfparam name="ATTRIBUTES.fields" default="">
      <cfparam name="ATTRIBUTES.fieldNames" default="">
      <cfif not isDefined("ATTRIBUTES.dao") OR NOT isObject(ATTRIBUTES.
dao)>
          <cfoutput>DAO Attribute is required</cfoutput>
          <cfexit method="exittag">
      </cfif>

      <link rel="stylesheet" href="genericUI/uistyles.css">

      <script language="javascript">
          function editRecord(id)
          {
              document.forms.gridControllerForm.pk.value = id;
              document.forms.gridControllerForm.controlMethod.value =
"edit";
              document.forms.gridControllerForm.submit();
          }

          function deleteRecord(id)
          {
              document.forms.gridControllerForm.pk.value = id;
              document.forms.gridControllerForm.controlMethod.value =
"delete";
              document.forms.gridControllerForm.submit();
          }

          function addRecord()
          {
              document.forms.gridControllerForm.controlMethod.value =
"add";
              document.forms.gridControllerForm.submit();
          }
      </script>

      <cfoutput>
          <form name="gridControllerForm" method="post" action="#CGI.
script_name#">
              <input type="hidden" name="controlMethod" />
              <input type="hidden" name="pk" value="">
          </form>
          <table class="genericUIGrid">
              <tr class="genericUIGridRow">
              <cfloop list="#ATTRIBUTES.fieldNames#" index="currentFieldNa
me">
                  <td class="genericUIGridHeader">#currentFieldName#</
th>
              </cfloop>
                  <th colspan="2" class="genericUIGridHeader" />
              </tr>
              <cfloop condition="#ATTRIBUTES.dao.hasNext()#">
                  <cfset currentItem = ATTRIBUTES.dao.getNext()>
                  <tr class="genericUIGridRow">
                      <cfloop list="#ATTRIBUTES.fields#" index="field">
                          <td class="genericUIGridCell">#currentItem.
getProperty(field)#</td>
                      </cfloop>
                      <td class="genericUIGridCellEdit"><a
href="javascript:editRecord(#currentItem.getID()#);">Edit</a></td>

                      <td class="genericUIGridCellEdit"><a
href="javascript:deleteRecord(#currentItem.getID()#);">Delete</a></td>

                  </tr>
```

```
              </cfloop>
          </table>
          <a href="javascript:addRecord();">Add</a>
      </cfoutput>
  </cfcase>
  <cfcase value="end">

  </cfcase>
</cfswitch>


detail.cfm

<cfswitch expression="#thisTag.executionMode#">
  <cfcase value="start">
      <cfparam name="ATTRIBUTES.fields" default="">
      <cfparam name="ATTRIBUTES.fieldNames" default="">
      <cfif not isDefined("ATTRIBUTES.dao") OR NOT isObject(ATTRIBUTES.
dao)>
          <cfoutput>DAO Attribute is required</cfoutput>
          <cfexit method="exittag">
      </cfif>

      <link rel="stylesheet" href="genericUI/uistyles.css">

      <script language="javascript">
          function submitEdit()
          {

  document.forms.detailControllerForm.controlMethod.value="update";
              document.forms.detailControllerForm.submit();
          }

          function cancelEdit()
          {
              document.forms.detailControllerForm.controlMethod.
value="list";
              document.forms.detailControllerForm.submit();
          }
      </script>
      <cfoutput>
          <form name="detailControllerForm" method="post" action="#CGI.
script_name#">
              <input type="hidden" name="controlMethod" />
              <input type="hidden" name="pk" value="#ATTRIBUTES.dao.
getID()#">
              <table class="genericUIGrid">
                  <cfloop from="1" to="#listLen(ATTRIBUTES.fieldNames)#"
index="i">
                      <tr>
                          <td class="genericUIDetailHeader">#listGetAt(AT
TRIBUTES.fieldNames, i)#</th>
                          <td class="genericUIDetailInput">
                              <input type="text" name="#listGetAt(ATTRIBUT
ES.fields, i)#" value="#ATTRIBUTES.dao.getProperty(listGetAt(ATTRIBUTES.
fields, i))#" size="50" />
                          </td>
                      </tr>
                  </cfloop>
                  <trclass="genericUIDetailControls">
                      <td colspan="2" >
                          <input type="button" value="Submit"
onClick="submitEdit();" />
                          <input type="button" value="Cancel"
onClick="cancelEdit();" />
                      </td>
                  </tr>
              </table>
          </form>
      </cfoutput>
  </cfcase>
  <cfcase value="end">

  </cfcase>
```

```
</cfswitch>
```

Listing 2: genericmasterdetail.cfm

```
<cfparam name="ATTRIBUTES.formScope" default="#structNew()#">
<cfparam name="ATTRIBUTES.table" default="">
<cfparam name="ATTRIBUTES.fields" default="">
<cfparam name="ATTRIBUTES.fieldnames" default="">

<cfswitch expression="#ATTRIBUTES.formScope.controlMethod#">
  <cfcase value="list">
      <cfset itemList = APPLICATION.objectBreeze.list(ATTRIBUTES.table,
"") />
      <cfmodule  template="grid.cfm"
                 fields="#ATTRIBUTES.fields#"
                 fieldNames="#ATTRIBUTES.fieldNames#"
                 dao="#itemList#">

  </cfcase>

  <cfcase value="edit">

      <cfscript>
          detailItem = APPLICATION.objectBreeze.objectCreate(ATTRIBUTES.
table);
          detailItem.read(ATTRIBUTES.formscope.pk);
      </cfscript>
      <cfmodule  template="detail.cfm"
                 fields="#ATTRIBUTES.fields#"
                 fieldNames="#ATTRIBUTES.fieldnames#"
                 dao="#detailItem#">
  </cfcase>
  <cfcase value="add">

      <cfscript>
          detailItem = APPLICATION.objectBreeze.objectCreate(ATTRIBUTES.
table);
      </cfscript>
      <cfmodule  template="detail.cfm"
                 fields="#ATTRIBUTES.fields#"
                 fieldNames="#ATTRIBUTES.fieldnames#"
                 dao="#detailItem#">
  </cfcase>

  <cfcase value="update">
      <cfscript>
          detailItem = APPLICATION.objectBreeze.objectCreate(ATTRIBUTES.
table);
          if (structKeyExists(ATTRIBUTES.formscope, "pk") AND ATTRI-
BUTES.formscope.pk NEQ "")
          {
              detailItem.read(ATTRIBUTES.formscope.pk);
          }

          for (i = 1; i LTE listLen(ATTRIBUTES..formscope.fieldnames); i
= i + 1)
          {
              if (NOT listFindNoCase("controlMethod,pk",
listGetAt(ATTRIBUTES.formscope.fieldnames, i)))
              {
                  detailItem.setProperty(listGetAt(ATTRIBUTES.formscope.
fieldnames, i), FORM[listGetAt(ATTRIBUTES.formscope.fieldnames, i)]);
              }
          }
          detailItem.commit();
      </cfscript>
      <cfoutput>
      <form name="controllerForm" action="#CGI.SCRIPT_NAME#"
method="post">
          <input type="hidden" name="controlMethod" value="list" />
          <input type="text" name="garbage" />
      </form>
      <script language="javascript">
          document.forms.controllerForm.submit();
      </script>
      </cfoutput>
  </cfcase>

  <cfcase value="delete">
      <cfscript>
          detailItem = APPLICATION.objectBreeze.objectCreate(ATTRIBUTES.
table);
          detailItem.read(ATTRIBUTES.formscope.pk);
          detailItem.delete();
      </cfscript>
      <cfoutput>
      <form name="controllerForm" action="#CGI.SCRIPT_NAME#"
method="post">
          <input type="hidden" name="controlMethod" value="list" />
          <input type="text" name="garbage" />
      </form>
      <script language="javascript">
          document.forms.controllerForm.submit();

      </script>
      </cfoutput>
  </cfcase>

</cfswitch>


drivers.cfm

<cfparam name="FORM.controlMethod" default="list">
<cfparam name="FORM.pk" default="">

<cfmodule    template="genericUI/genericmasterdetail.cfm"
          table="drivers"
          fields = "first_name,last_name,nationality"
          fieldnames="First Name,Last Name,Nationality"
          formScope="#FORM#">


classes.cfm

<cfparam name="FORM.controlMethod" default="list">
<cfparam name="FORM.pk" default="">
<cfif FORM.controlMethod EQ "list">
  <cfset fieldList = "class_name">
  <cfset fieldNames = "Class Name">
<cfelse>
  <cfset fieldList = "class_name,chassis_description,engine_descrip-
tion">
  <cfset fieldNames = "Class Name,Chassis Desc,Engine Desc">
</cfif>
<cfmodule    template="genericUI/genericmasterdetail.cfm"
          table="classes"
          fields = "#fieldList#"
          fieldnames="#fieldNames#"
          formScope="#FORM#">


teams.cfm

<cfparam name="FORM.controlMethod" default="list">
<cfparam name="FORM.pk" default="">

<cfmodule    template="genericUI/genericmasterdetail.cfm"
          table="team"
          fields = "team_name,team_principal,city,state"
          fieldnames="Team Name,Team Principal,City,State"
          formScope="#FORM#">
```

Download the Code...
Go to http://coldfusion.sys-con.com

How ColdFusion MX 7 Made Me a Hero

Improving the donation process

By John Baldwin

Formed in 1921 to perpetuate the legacy of Hoosier Poet James Whitcomb Riley, Indianapolis's Riley Children's Foundation is at the forefront of serving Indiana's children.

The Riley Children's Foundation (a 501 (c)(3) non-profit corporation) is dedicated to the health and well being of the children of Indiana through its philanthropic leadership of:

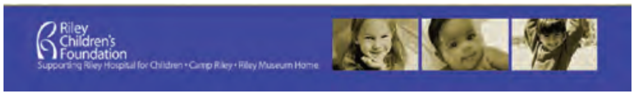
- ***Riley Hospital for Children:*** A renowned pediatric medical center
- ***Camp Riley:*** A nurturing recreational environment for children with special health needs
- ***The James Whitcomb Riley Museum Home:*** A Victorian preservation educating young and old on the life of James Whitcomb Riley, the "The Hoosier Poet"

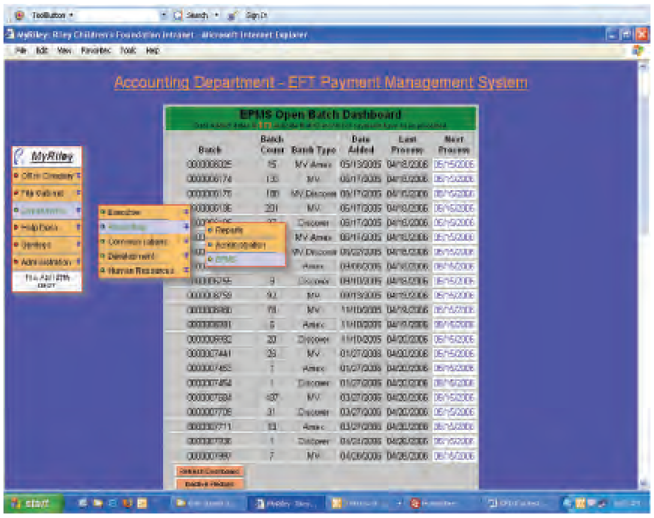
The Foundation generates financial support for its mission through philanthropic donations from many sources – individuals, corporations, trusts, and associations.

For pledge donations requiring periodic payments, our donor database system (BSR Sungard Advance, running under an Oracle database) maintains a corresponding payment schedule. In addition, for pledges being paid by credit card, Advance provides data tables to register credit card account information (e.g., account number and expiration date). Advance does not, however, provide a method to automatically process EFT (Electronic Funds Transfer) monthly credit card pledge payments with a third-party credit card processor. Credit card information must be entered manually and separately processed using third-party credit processing systems, and credit card processing results must be entered manually into the Advance gift batch system. Thus, the tedious processing of monthly credit card pledge payments can take up extensive amounts of dedicated staff labor.

Using Cold Fusion 7.01 and MS SQL Server technologies, the Riley Children's Foundation developed a Web services system (EFT Payment Management System – EPMS) that efficiently processes thousands of credit card pledge payments each month. The system pushes large volumes of credit card payment requests to a third-party credit vendor and, based on credit card payment results, automatically creates pledge payment batches in Advance. The system provides a comprehensive dashboard control panel for launching and monitoring payment processes, and it provides several utilities for managing individual payment change situations (e.g., pledge payoffs and pledge withdrawals). In addition, using embedded ColdFusion 7.01 reporting options, the system can generate on-demand highly customized end-of-year donor statement letters in Adobe Acrobat PDF format.

Before this system was deployed, the staff spent up to a week each month manually processing monthly credit card–based pledge payments. Now, monthly processing can be completed


Riley Children's Foundation
Supporting Riley Hospital for Children • Camp Riley • Riley Museum Home



Accounting Department - EFT Payment Management System

EPMS Open Batch Dashboard

Batch	Batch Count	Batch Type	Date Added	Last Process	Next Process
0000006025	15	MV Amex	05/13/2005	04/18/2006	05/15/2006
0000006174	130	MV	05/17/2005	04/18/2006	05/15/2006
0000006175	189	MV Discover	05/17/2005	04/18/2006	05/15/2006
0000006136	201	MV	05/17/2005	04/18/2006	05/15/2006
[illegible]	[illegible]	Discover	05/17/2005	04/18/2006	05/15/2006
[illegible]	[illegible]	MV Amex	05/11/2005	04/18/2006	05/15/2006
[illegible]	[illegible]	MV Discover	05/27/2005	04/18/2006	05/15/2006
[illegible]	[illegible]	Amex	04/08/2005	04/18/2006	05/15/2006
0000006755	9	Discover	09/10/2005	04/18/2006	05/15/2006
0000006759	92	MV	09/13/2005	04/18/2006	05/15/2006
0000006980	78	MV	11/10/2005	04/18/2006	05/15/2006
0000006981	5	Amex	11/10/2005	04/18/2006	05/15/2006
0000006982	20	Discover	11/10/2005	04/20/2006	05/15/2006
0000007441	25	MV	01/27/2006	04/20/2006	05/15/2006
0000007453	7	Amex	01/27/2006	04/20/2006	05/15/2006
0000007454	1	Discover	01/27/2006	04/20/2006	05/15/2006
0000007684	437	MV	03/27/2006	04/20/2006	05/15/2006
0000007706	31	Discover	03/27/2006	04/20/2006	05/15/2006
0000007711	13	Amex	03/27/2006	04/20/2006	05/15/2006
0000007936	1	Discover	04/24/2006	04/26/2006	05/15/2006
0000007997	7	MV	04/26/2006	04/26/2006	05/15/2006

in just a few hours. For staff, this system has been a godsend and has allowed us to expand our pledge donation program without adding processing staff. For me, I have been lauded a hero. This system is truly an example of ColdFusion's capability to quickly and easily apply an effective solution to seemingly intractable computing and data processing problems.

About the Author

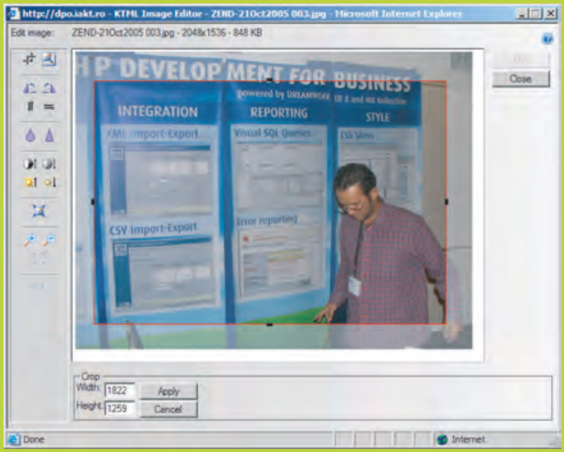
John Baldwin is the director of information technology at Riley Children's Foundation.

jbaldwin@rileykids.org

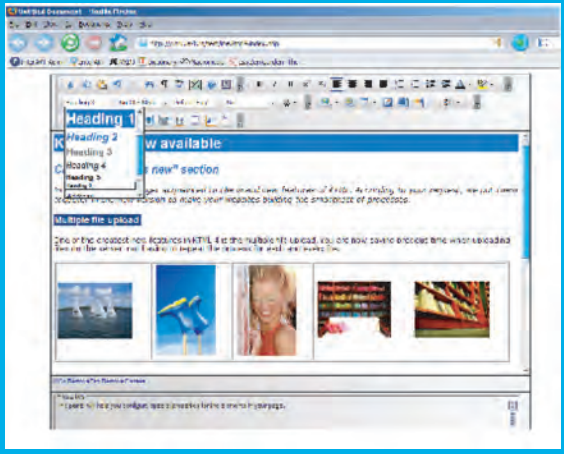
KTML4 Word editing in browser

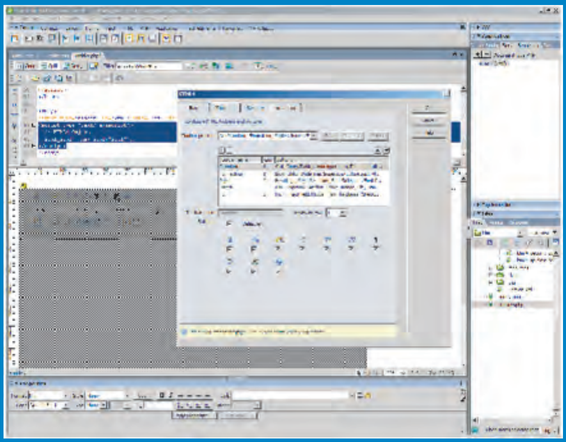
Along with the most affordable **UNLIMITED** licence

● Revolutionary Image Editor


http://dpo.iakt.ro - KTML Image Editor - ZEND-21Oct2005 003.jpg - Microsoft Internet Explorer
Edit image: ZEND-21Oct2005 003.jpg - 2048x1536 - 848 KB
Close
Crop
Width: 1822
Height: 1259
Apply
Cancel
Done
Internet

Word-like visual CSS styles ●





● Fast Dreamweaver integration

Instant paste from Word
Incredible speed
Easy to use Word-like toolbars
Improved CSS authoring
Remote File Explorer
XHTML 1.1 compliant

Wide browser compatibility ●

Multiple file upload at once
HTML Table Editor
Support for multimedia (Flash, Avi)
Documents management (.doc, .pdf)
Page templates
WAI compliant

Please visit **www.interaktonline.com/ktml4/** for details

work smart

Interakt

Other companies in this magazine spent a lot of time on pretty ads. As you can see, we did not. We spent our time hiring the best people and training them to deliver outstanding support for your website. We spent our time building a state of the art datacenter and staffing it with people who care about your website like it’s their own. Compassion, respect, credibility, ownership, reliability, “never say no,” and exceed expectations are words that describe our service philosophy. From the first time you interact with us, you'll see what a difference it really makes. And you'll also forgive us for not having a pretty ad.

HostMySite.com

WEB HOSTING • MANAGED DEDICATED SERVERS • COLOCATION • VPS • ECOMMERCE • BLOGGING • EMAIL