SPECIAL INSERT: WEB SERVICES CONFERENCE & EXPO Pg. 41

U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

ColdFusionJournal.com

March 2003 Volume: 5 Issue: 3

RETAILERS PLEASE DISPLAY UNTIL MAY 31, 2003

$8.99US $9.99CAN

04>

0 71486 02689 1

SYS-CON MEDIA

ColdFusion Developer's Journal

editorial advisory board

Jeremy Allaire, *CTO*, **macromedia, inc.**
Charlie Arehart, *CTO*, **systemanage**
Michael Dinowitz, **house of fusion, fusion authority**
Steve Drucker, *CEO*, **fig leaf software**
Ben Forta, *products*, **macromedia**
Hal Helms, *training*, **team macromedia**
Kevin Lynch, *chief software architect*, **macromedia**
Karl Moss, *principal software developer*, **macromedia**
Michael Smith, *president*, **teratech**
Bruce Van Horn, *president*, **netsite dynamics, LLC**

editorial

editor-in-chief
Robert Diamond robert@sys-con.com
technical editors
Charlie Arehart charlie@sys-con.com
Raymond Camden raymond@sys-con.com
editorial director
Jeremy Geelan jeremy@sys-con.com
executive editor
Jamie Matusow jamie@sys-con.com
editor
Nancy Valentine nancy@sys-con.com
associate editors
Gail Schultz gail@sys-con.com
Jean Cassidy jean@sys-con.com
assistant editor
Jennifer Stilley jennifer@sys-con.com

production

production consultant
Jim Morgan jim@sys-con.com
art director
Alex Botero alex@sys-con.com
associate art directors
Louis F. Cuffari louis@sys-con.com
Richard Silverberg richards@sys-con.com
Tami Beatty tami@sys-con.com

contributors to this issue

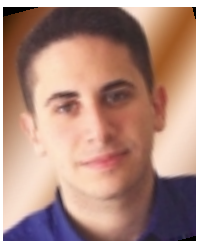
Charlie Arehart, Rob Brooks-Bilson, Phil Cruz, Steve Drucker, Ben Forta, Vlad Friedman, Hal Helms, Simon Horwith, Brendan O'Hara, Kevin Towes

editorial offices

SYS-CON MEDIA
135 CHESTNUT RIDGE RD., MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9600
COLDFUSION DEVELOPER S JOURNAL (ISSN #1523-9101) is published monthly (12 times a year) by **SYS-CON Publications, Inc.**
postmaster: send address changes to:
COLDFUSION DEVELOPER S JOURNAL
SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645

The End of an Era

A fairly big announcement related to the ColdFusion industry came out this week. No, not about a new version, or another service pack just yet, but a "personal news announcement" straight from the top. After eight years with Allaire, and then with Macromedia, Jeremy Allaire announced to the world via his blog, that he is leaving the company to pursue other projects – namely a Boston-area venture firm.



By Robert Diamond

Recounting the progress that ColdFusion has made in recent times, he wrote in his blog this week, "ColdFusion has been transformed and given new life, becoming a standards-based platform, adding powerful tools for XML, Web services, component, and rich Internet app development. ColdFusion MX continues to innovate in making building Web applications easier and enjoyable." Well said if you ask me…

He's not done with Macromedia, as he writes, "I'll have a role as a Founder Emeritus of Macromedia, keeping active with Macromedia's customer community, periodically writing and speaking on Macromedia's behalf, and also staying in touch with Macromedia management and helping great products like Macromedia Flash, Dreamweaver, and ColdFusion keep up with the times."

Those of us who have had the pleasure of hearing Jeremy speak many times, are happy to know that he won't be cutting out those engagements.

Reaction to his announcement has resulted in a lot of mixed feedback throughout the Internet. What's been written has been a mixture of best wishes, and accusations of selling out and then deserting the community that he has helped build up. Personally, I don't take his message as being any sort of desertion toward any of us; and I feel that he's left the world of ColdFusion in some very capable hands as long as Macromedia continues to give CF the support that it has. I'd like to see us all programming in CFML for years to come!

CFDJ wishes Jeremy the best of luck with his next project. We've got no doubt that he'll stay in touch and, if what he's doing is *i*-technology related, it'll fit into this, or one of SYS-CON's other magazines.

Back to the business at hand: *CFDJ* March is as packed as ever with some great articles. I'll mention just a few highlights instead of the complete rundown this month. Kevin Towes has contributed a fantastic piece about using the Flash Communication Server, Flash Remoting, and ColdFusion MX. I haven't had the occasion to try out Flash Remoting yet, but when I do, this article will be right by my side.

Simon Horwith's **Tales from the List** column (note to members of the CFDJList, you'll all be pleased that we've upgraded to better, faster, and more stable list management software; those not on the list – go sign up!) talks about some of the woes involved with caching XML.

Phil Cruz reviews CFMX Exam Buster from CentraSoft, a testing tool geared to help developers with the Developer Certification Exam. Vlad Friedman has written a helpful piece on ColdFusion Server recovery, which we've all had to deal with at one point or another (some more than others). Rob Brooks-Bilson has interviewed Jason Egan of CFDynamics about their hosting form… a good read for those looking at selecting a host. We'll be profiling other companies as well in the coming months.

Charlie Arehart, one of our illustrious co-technical editors, writes this month about Flash for CFers in, "Getting over the Hump." It's an article that manages to more than live up to its witty title!

Last but not least, if you haven't nominated your products yet for the 2003 Readers' Choice Awards – get to it at www.sys-con.com/coldfusion!

About the Author

Robert Diamond is vice president of information systems for SYS-CON Media, and editor-in-chief of **ColdFusion Developer's Journal.** *He is also the founding editor of* **Wireless Business & Technology**. *Named one of the "Top thirty magazine industry executives under the age of 30" in* Folio *magazine's November 2000 issue, Robert holds a BS degree in information management and technology from the School of Information Studies at Syracuse University.*

robert@sys-con.com

SYS-CON MEDIA

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

business development
vp, business development
Grisha Davida grisha@sys-con.com
COO/CFO
Mark Harabedian mark@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 account manager
Megan Ring-Mussa megan@sys-con.com
associate sales managers
Carrie Gebert carrieg@sys-con.com
Kristin Kuhnle kristin@sys-con.com
Alisa Catalano alisa@sys-con.com

sys-con events
president, events
Grisha Davida grisha@sys-con.com
conference manager
Michael Lynch mike@sys-con.com
sales executives, exhibits
Michael Pesick michael@sys-con.com
Richard Anderson richarda@sys-con.com

customer relations
customer service representative
Margie Downs margie@sys-con.com
manager, jdj store
Rachel McGouran rachel@sys-con.com

sys-con.com
vp, information systems
Robert Diamond robert@sys-con.com
web designers
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com
online editor
Lin Goetz lin@sys-con.com

accounting
financial analyst
Joan LaRose joan@sys-con.com
accounts receivable
Kerri Von Achen kerri@sys-con.com
accounts payable
Betty White betty@sys-con.com

subscriptions
Subscribe@sys-con.com
1 888 303-5282
For subscriptions and requests for bulk orders, please send your letters to Subscription Department
Cover Price: $8.99/issue
Domestic: $89.99/yr (12 issues)
Canada/Mexico: $99.99/yr
all other countries $129.99/yr
(U.S. Banks or Money Orders)
Back issues: $12 U.S. $15 all others

XML Caching Woes

Before examining this month's thread, I just want to take a moment to welcome Christian Cantrell to the community. Christian took the position of Macromedia Community Manager for ColdFusion and, shortly thereafter, introduced himself to the list. Since then, he has chimed in from time to time not only to voice the "official word" of Macromedia, but also to assist developers. So far, Christian has done a great job keeping the lines of communication open between Macromedia and the development community, and I'm sure I speak for the entire list when I say that we look forward to working with him in the future. Welcome, Christian!

By Simon Horwith

Now for this month's thread, which originated from a post by Candace Cottrell of The Children's Medical Center (Dayton, Ohio). Several months ago, Candace migrated her existing ColdFusion site to ColdFusion MX. One of the new features she is now trying to leverage is the XML parsing functionality in CFMX. Candace wrote to the list explaining that she is implementing an XML-driven Flash MX menu, and wants to know whether or not there is anything she can do server-side to prevent Internet Explorer from caching XML and/or Flash movies. She went on to explain the symptom: after adding data to the database tables being queried and used to generate her XML, the XML-driven menu does not reflect the change(s) made to her data.

Mike Townend replied, stating that Candace could try dynamically changing the URL on each request by appending a random number to the end. No sooner had Mike's response been received than a second response was posted by Stephen Moretti, longtime list contributor from overseas (United Kingdom). Stephen has been generating Flash menus with XML from ColdFusion 5 and has never encountered any caching problems! He suggested that this may be a CFC or (more likely) Flash Remoting issue. Stephen posted the following example of loading an XML file from ActionScript, which has always worked for him:

```
//begin - set string
XMLURL =
SiteURL+addedpath+"/xmlmenu/index.cfm?"+QueryString;
//initialize new xml object
menuxml = new XML();
menuxml.ignoreWhite = true;
//load xml from url
menuxml.load(XMLURL);
//end
```

Candace replied to both, stating that Mike's suggestion did the trick – Candace posted that the following URL call now works like a charm:

```
output.xml?PageType_ID = #PageType_
ID#&MenuID=#RandRange(1,100000000)#
```

I chose to discuss this thread because it is an excellent example of how new functionality in ColdFusion MX is leading developers to discover new solutions to development tasks. This thread is also indicative of the types of cross-product postings that are on the rise in the community as more and more developers broaden their skill sets (and development platforms) to include not one but several Macromedia products.

Be sure to read next month's column in which I will talk about my conversations with both Christian Cantrell (Macromedia ColdFusion Community Manager) and Sarge (Macromedia Senior Product Support Engineer). I'll fill you in on what Macromedia is doing to address both the community and technical support challenges that were created by the release of the MX suite of (integrated) products as well as post-MX Macromedia initiatives.

About the Author

Simon Horwith, senior consultant at Fig Leaf Software in Washington, DC, has been using ColdFusion since version 1.5. He is a Macromedia-certified Advanced ColdFusion and Flash developer and is a Macromedia-certified instructor. In addition to administering the CFDJList List Serve and presenting at DC-area CFUGs, Simon is a contributing author to Professional ColdFusion 5.0 *(WROX) and to* ColdFusion MX - The Complete Reference *(McGraw-Hill), as well as technical editor of* The ColdFusion 5.0 Certification Study Guide *(Syngress).*

shorwith@figleaf.com

ColdFusion & Java

More than just the sum of their parts

ColdFusion MX is fast approaching its first birthday, and what a first year this has been. A completely brand new ColdFusion, incredible new features and technologies, and even versions that run on J2EE servers on all sorts of platforms (including the oft requested Mac OSX). Which makes this a great time to review the ColdFusion-Java relationship, and what it means for you.

By Ben Forta

So, What Exactly is Java?

Over the past few years I have regularly started presentations by asking, "What exactly is Java?" We've come a long way in a short time – the blank expressions and assorted answers have been replaced by a spattering of raised hands and assorted answers. We're making progress.

The answers I get (ignoring the jabs, platform jokes, and painful groans) usually fall into one of two categories:

- Java is a language.
- Java is a platform.

So, which is correct? Actually, both.

Java is indeed a language, an object-oriented strongly typed bytecode portable language. And Java the language has nothing to do with the Internet. Java the language is just that – a programming language. Java the language was designed to be the ultimate programming language – write it once and it will run anywhere, and not just on computers either.

Java, or more accurately J2EE, is also a platform. Java the platform provides the foundations and building blocks for everything from database integration to transactions and guaranteed delivery to mobile computing to legacy system integration to security to directory service integration to… you get the idea.

Java the language and Java the platform are not one and the same. Java the language can be used to write applications that are not J2EE applications. And J2EE can be (or should be) leveraged independent of Java the language.

A Better ColdFusion

ColdFusion MX runs on top of underlying Java.

ColdFusion developers, for the most part, do not care for Java the language. This is quite understandable. ColdFusion is easy, intuitive, rapid, and even fun – all adjectives rarely heard when describing Java.

So what's in it for ColdFusion developers? Lots…

- The Java internals provide for a more reliable, more robust, and more scalable ColdFusion – even if those internals are mostly ignored.
- The new Java-based engine is faster than the old interpreter.
- Java APIs, specifications, and functionality can all be used by ColdFusion developers – without having to know Java.
- Java components, objects, and tags can be used by ColdFusion developers – again without having to know Java.
- ColdFusion can take advantage of Java internals, for example J2EE sessions – this way ColdFusion can share session state information.

No downside at all – you can use ColdFusion as you always have. And lots of upside too.

A Better Java

ColdFusion MX runs on top of underlying Java and in doing so makes for a better Java.

I asked earlier what Java was, now let's turn that around. What exactly is ColdFusion? Well, ColdFusion is (at least) two things:

- ColdFusion, or rather CFML, is a language.
- ColdFusion is an application server.

Prior to ColdFusion MX there was no way to separate ColdFusion the language from ColdFusion the application server. But that has changed. You can use ColdFusion as you always did – an application server with a built-in language – or you can use ColdFusion as a scripting language on an application server of your choice.

Why do we use ColdFusion? That's an easy one – remember the adjectives I used earlier? Easy, intuitive, rapid, and

—continued on page 14

Design Patterns in ColdFusion: Template Method Pattern

Implementing proven solutions to common development issues

ColdFusion has always been an interesting Rapid Application Development tool but never a language taken completely seriously by object-oriented programmers because of its purely structural nature. C++ and Java CFX tags, and the somewhat unreliable nature of <cfobject>, kept many advanced programming concepts unreachable by the average CF5 developer. With the release of ColdFusion MX, Macromedia simplified the use of Java from within ColdFusion. Additionally, the language itself has been extended to include a Java class-like construct known as ColdFusion Components (CFCs). With this new construct and a more object-oriented syntax comes an inevitable learning curve as we throw aside (some) of our old structured programming techniques and adapt others to a newer and ultimately better way of doing things. Hopefully you've already been experimenting with CFCs and are looking for ways to organize and reuse your code. If so, design patterns may be just what you are looking for.

By Brendan O'Hara

Enter Design Patterns

When ColdFusion developers think of design patterns we tend to think of graphical user interface (GUI) patterns, which may relate to navigation or a user's experience or path through a specific site or application. Two common examples are Bread Crumbs, in which a path of links is displayed to traverse back up through a site's hierarchy; and Double Tab, in which a top level of links reveals a different sub-level of links underneath. However, these are not what most developers think of when they think of design patterns.

When Java or C++ developers think of design patterns they tend to think of language-neutral patterns they apply to solve common programming problems. These common problems have already been encountered and solved countless times in the past. As developers we need only to be able to recognize these common problems and implement the appropriate solution. Think of them as tried-and-true answers to age-old questions.

Many of these answers and questions were compiled into a book by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides – commonly referred to as the "Gang of Four." Their book, *Design Patterns: Elements of reusable object-oriented software*, is the premier book on design patterns in software development. Interfaces for many of the patterns discussed in their book were specifically written into the Java foundation classes, which form the core of the Java programming language. Although the examples in the book are written in C++ and Smalltalk, the concepts presented remain the same. Some of these patterns may not be feasible or desirable due to the stateless and performance-driven nature of Web programming. Many other patterns are just as useful to ColdFusion developers as they are to our friends in C++. One of these is the Template Method pattern.

What Is the Template Method Pattern?

The Gang of Four book describes the Template Method pattern as "defining the skeleton of an algorithm in an operation, deferring some steps to subclasses, while letting subclasses redefine certain steps of an algorithm without changing the algorithm's structure."

OK, so what does that mean? It means that the Template Method pattern is firmly rooted in the principles of two major OOP concepts: inheritance and polymorphism.

Inheritance in ColdFusion involves a top-level CFC that contains common functionality and is known as a "base class" or base CFC. Any CFC that extends a base CFC is seen as deriving from or extending that CFC. A derived CFC "inherits" all of the base CFC's methods and data members. The relationship between a derived CFC and its base CFC is known as an "is a" relationship. The derived CFC "is a" base CFC. If we have a derived CFC called "Employee" that extends the User base CFC, then we would say that Employee "is a" User.

Polymorphism, or more specifically subtype polymorphism, is the ability to override and redefine methods in "derived" CFCs that they inherit from their "base class" CFC. Because of the relationship they have with a common base CFC, all related CFCs can be treated similarly and we can guarantee some level of compatibility with common methods. No matter what derived CFC we're working with, we know generally that an inherited or overridden method will work the right way for the actual CFC being processed. These methods will be processed differently depending on whether the actual method being executed at runtime is implemented in the base CFC or a derived CFC. Take the Employee and User CFCs shown. If User has an Authenticate() method, we know that through inheritance that Employee will have one as well. Although Employee may override the version of the method it inherits from User, we can assume that their arguments and what they return will act similarly, making them interchangeable in some circumstances. This interchangeability is what defines them as polymorphic (see Figure 1).

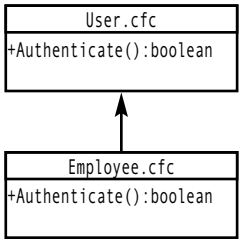


Figure 1: A UML diagram showing simple inheritance

The Template Method pattern is one of the most important concepts for those unfamiliar with OOP and design patterns because it relies so heavily on, and encourages the use of, inheritance and polymorphism. These fundamental object-oriented principles form the foundation of many design patterns.

Template Method Pattern Explained

The Template Method pattern requires at least two derived CFCs and a base class CFC, which may be abstract. In OOP, an abstract base class is one that defines the behavior of any classes that extend it while deferring some of the actual implementation to those derived classes. In other words, the base class CFC is abstract because it defines the names and function calls for certain methods that it does not itself implement. Abstract CFCs or methods are not an inherently available feature in ColdFusion as there is no place to define them as being abstract. However, implementing them is still relatively easy. For abstract methods in CFCs I use a <cfabort> tag with a custom error message to ensure the method implementation from the base class CFC is not called.

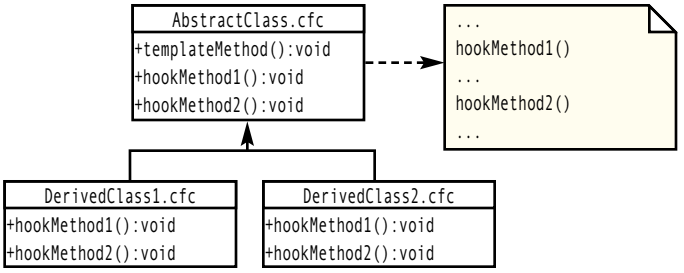


Figure 2: A UML diagram of a generic version of the Template Method Pattern

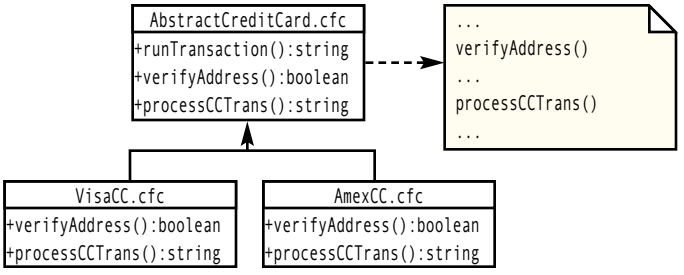


Figure 3: A UML diagram of our Credit Card Processing version of the Template Method Pattern

```
<cffunction name="DoSomething" access="public" output="false" return-
type="string">
    <!--- Abstract Method - Must be overridden or will abort --->
    <cfabort showerror="This Method is Abstract and needs to be overridden">
</cffunction>
```

A Template method is a method in the base CFC. A Template method is concrete, meaning it is implemented in the base CFC, and calls one or more Hook methods. A Hook method is a method in a CFC that defines a task that will be overridden in any CFC that derives from it. "Hook" refers to a place to hang your customizable code. Hook methods are usually abstract but can have a default implementation in the base CFC. It also must be ensured that all Hook methods behave uniformly to enforce polymorphism. Any CFC that derives from this base CFC is referred to as concrete if it overrides all inherited abstract methods. These concrete derived CFCs provide the implementation of the Hook methods (see Figure 2).

The Template Method pattern is most useful when you have two or more different components (custom tags, CFCs) that duplicate a significant portion of each other's code yet do not reuse or share any portion of this code through custom tags, CFIncludes, or CFCs. If a change that affects these components becomes necessary, changes must be made to all the relevant members individually.

To alleviate this problem a developer may decide to break out the steps of the algorithm so the duplicate code can be shared and the variable code can be implemented independently. The shared steps are implemented as concrete method(s) in the base CFC, while the unique steps are given either an abstract or a default implementation as method(s) in the base CFC. These unique steps represent "hooks" that will need to be implemented as concrete methods in any derived

CFC. This is a somewhat unusual concept as the method being called is in the base CFC and it in turn is calling Hook methods in whatever derived CFC has been instantiated. The Gang of Four refers to this somewhat inverted top-down control structure as "the Hollywood principle" – "Don't call us, we'll call you."

Why the Template Method Pattern?

One major advantage to the type of polymorphic behavior exhibited in the Template Method pattern is that it generally removes the "switch" constructs required in purely structured languages like previous versions of ColdFusion. As CF5 developers we often try to repurpose custom tags to fit more than one type of situation. In doing so we often need to pass in a type attribute that describes what structure or data type the custom tag is performing the action on or possibly what type of action to perform. Here is an example custom tag call for processing a credit card transaction:

```
<cf_processCreditCard           type="#form.CCtype#"
CCnumber="#form.ccnumber#"
Expiration="#form.Expiration #"
Amount="#form.Amount #"
SecurityCode="#form.SecurityCode #"
Name="#form.Name#"
Street1="#form.Street1#"
Street2="#form.Street2#"
City="#form.City #"
State="#form.State #"
Zip="#form.Zip #"  />
```

This is then followed by a <cfswitch> or <cfif>, which separates the actual processing.

```
<cfswitch expression="#attributes.type#">
   <cfcase value="Visa">
          <!--- Verify Credit Card Address - - - >
          <!--- Process CC Transaction - - - >
…etc …
</cfcase>
   <cfcase value="Amex">
          <!--- Verify Credit Card Address - - - >
          <!--- Process CC Transaction - - - >
…etc …
</cfcase>
   …etc …
</cfswitch>
```

Now the code that executes to "Verify Credit Card Address" may be the same for Visa and MasterCard, but we can't reuse it without another layer of <cfswitch> and <cfif> inside our existing <cfswitch>. However, by using inheritance and polymorphism, and the Template Method pattern of course, we can deal with this in a more architecturally elegant way.

The Template Method Pattern in Action

We know that to implement the ProcessCreditCard custom tag as CFCs using the Template Method pattern we need a base class for our Credit Card CFCs. We also know this base CFC needs two Hook methods; VerifyAddress() and ProcessCCTrans(). These will be abstract in our base CFC but will need to be implemented in any derived CFCs. Next we need to implement a Template method in our base CFC called RunTransaction(). RunTransaction() calls the two Hook methods VerifyAddress() and ProcessCCTrans() (see Listing 1).

Now we need our derived CFCs for Visa and Amex, which will have independent implementations of the VerifyAddress() and ProcessCCTrans(). These Hook methods from our base CFC must be concrete in our derived CFCs (see Listings 2 and 3).

Both derived CFCs contain implementations of the VerifyAddress() and ProcessCCTrans() methods. The AddressData argument is a structure that is passed through to the VerifyAddress() method implementations. The TransactionData structure is the equivalent for the ProcessCCTrans() method. Both implementations of these methods take one argument, a structure, which allows us to vary the number and name of arguments for each method depending upon their derived implementation (see Figure 3).

The Template Method RunTransaction() is fairly simple:

```
<cffunction name="RunTransaction" access="public" output="false"
returntype="string">
   <cfargument name="addressData" required="true" type="struct">
   <cfargument name="transactionData" required="true" type="struct">
   <cfif VerifyAddress(arguments.addressData)>
          <cfreturn processCCTrans(arguments.transactionData)>
   <cfelse>
          <cfreturn "Cannot Verify Address">
   </cfif>
</cffunction>
```

If the derived method implementation of VerifyAddress() being called returns false then we return the string "Cannot Verify Address" to the calling page. If it returns true (the address has been verified) we can proceed to processing the transaction. We simply return whatever is returned by the derived implementation of processCCTrans() to the caller.

On the calling page for our Template method example we are processing the submit form for a e-commerce application (see Listing 4).

First we need to instantiate a CFC for the type of credit card being processed. We create the data structures to pass in to the runTransaction() method that will be passed through to the verifyAddress() and ProcessCCTrans() methods. If a security code has been passed in via the form then add it to the transactionData structure to be passed through to the ProcessCCTrans() method. This would only apply if we had an instance of a VisaCC.cfc because American Express cards do not have security codes. We then invoke the RunTransaction() method passing in the two structures. Regardless of which credit card is being processed the algorithm remains constant while the actual implementation of pieces of said algorithm can vary.

design patterns

Factoring

The Template Method pattern also relies heavily on a software concept called factoring. Factoring is simply identifying duplicate behavior in related CFCs and moving it up into the base CFC. Because they are often written at different times, the code in related CFCs may not be duplicated exactly but the behavior may be. After you have separated the behavior that is duplicated from that which is not, you can move the duplicated behavior up into the base CFC while leaving the unique behavior in the derived CFCs. Refactoring is the process of going back to your derived CFCs every so often and factoring the common behavior up into the base CFC.

Conclusion

As we ColdFusion developers reuse, or more often repurpose, code in custom tags, we often end up solving a very specific problem with a very specific solution. While the <cf_ProcessCreditCard> custom tag does provide some code reuse, it is hardly the best or most future-proof answer to the problem. With the advent of CFCs and some knowledge of object-oriented design patterns we can learn to recognize common issues or patterns that come up often in software development and implement tried and tested solutions.

About the Author

Brendan O'Hara is one of the coauthors of Advanced Macromedia ColdFusion MX Application Development, *published by Macromedia Press. Brendan has a Macromedia ColdFusion MX Developer Certification along with Java and Linux certifications from Penn State University. He works as a ColdFusion architect and Java developer for E-tech Solutions and Amkor Technology in the Philadelphia suburbs.*

bohar@amkor.com

Listing 1 CreditCard.cfc

```
<!--- The Credit Card CFC is the Base Class for Credit Card CFCs. --->
<cfcomponent displayname="CreditCard" hint="Base Class for Credit Card CFCs">
<!--- The Template Method for CreditCard.cfc is RunTransaction().
RunTransaction() calls the two hook methods VerifyAddress() and
ProcessCCTrans() which must be implemented in any derived CFC's. --->
<cffunction name="RunTransaction" access="public" output="false" return
          type="string">
   <!--- The AddresData argument is a structure that is passed through
   to the VerifyAddress() method. Using a structure allows us to vary
   the number and name of arguments dependent upon their derived
   implementation. --->
   <cfargument name="addressData" required="true" type="struct">
   <cfargument name="transactionData"     required="true" type="struct">
   <cfif VerifyAddress(arguments.addressData)>
          <cfreturn processCCtrans(arguments.transactionData)>
   <cfelse>
          <cfreturn "Cannot Verify Address">
   </cfif>
</cffunction>
<cffunction name="ProcessCCTrans" access="public" output="false" return-
type="string">
   <!--- Abstract Method - Must be overridden or will abort --->
   <cfabort showerror="This Method is Abstract and needs to be overridden">
</cffunction>
<cffunction name="VerifyAddress" access="public" output="false" return
   type="boolean">
   <!--- Abstract Method - Must be overridden or will abort --->
   <cfabort showerror="This Method is Abstract and needs to be overridden">
</cffunction>
</cfcomponent>
```

Listing 2 VisaCC.cfc

```
<cfcomponent displayname="VisaCC" extends="CreditCard" hint="Base Class
for Credit Card CFCs">

<cffunction name="VerifyAddress" access="public" output="false" return
   type="boolean">
   <cfargument name="addressData" type="string" required="true" />
```

<bf> on <cf> *—continued from page 8*

even fun. ColdFusion, and in particular CFML, is the fastest and most efficient way to build applications for the Internet and Web.

So what's in it for Java developers? Lots…

- The simplicity and effectiveness of the CFML language.
- All of the ColdFusion integrated services and functionality.

Java developers can now take advantage of CFML. There are parts of any application that just should not be written in straight Java (the entire presentation layer for starters; you really don't want to manipulate URL and form values using low-level Java code) – CFML is ideal for that. Lots of upside, and no downside whatsoever.

For organizations that have invested fortunes in large J2EE installations, and in doing so have discovered that complexity makes the system unusable for a significant chunk of the organization, ColdFusion truly helps deliver the promise of Java and J2EE.

Conclusion

I've used this column many times to discuss building tiered structured applications, knowing where to use ColdFusion and where not to, leveraging back-end infrastructure, and more. In an ideal world, ColdFusion is the middle tier in a multitier application talking to all sorts of front ends and back ends. For ColdFusion developers, the Java platform provides the heavy lifting needed to power real-world applications.

For Java developers, ColdFusion on top of Java helps deliver the promise of Java, making this powerful platform actually usable. ColdFusion and Java are valuable independent of each other, but when combined, are more valuable than the sum of their parts.

About the Author

Ben Forta is Macromedia's senior product evangelist and the author of numerous books, including ColdFusion MX Web Application Construction Kit *and its sequel,* Advanced ColdFusion MX Application Development, *and is the series editor for the new "Reality ColdFusion" series. For more information visit www.forta.com.*

ben@forta.com

```
    <cfparam name="arguments.addressdata.name" type="string" />
    <cfparam name="arguments.addressdata.street1" type="string" />
    <cfparam name="arguments.addressdata.street2" default=""
          type="string" />
    <cfparam name="arguments.addressdata.city" type="string" />
    <cfparam name="arguments.addressdata.state" type="string" />
    <cfparam name="arguments.addressdata.zip" type="string" />
    <!--- In the real world you would actually do something here --->
    <cfreturn true>
</cffunction>
<cffunction name="processCCtrans" access="public" output="false" return-
type="string">
    <cfargument name="transactionData" type="string" required="true" />
    <cfparam name="arguments.transactionData.name" type="string" />
    <cfparam name="arguments.transactionData.ccnumber" type="string" />
    <cfparam name="arguments.transactionData.amount" type="numeric">
    <cfparam name="arguments.transactionData.securitycode" type="string" />
    <cfparam name="arguments.transactionData.expiration" type="date">
    <!--- In the real world you would actually do something here --->
    <!--- Instead a random determination of approval --->
    <cfif len(arguments.transactionData.amount) GT 100>
          <cfreturn "Decline">
    <cfelse>
          <cfreturn "98765432">
    </cfif>
</cffunction>
</cfcomponent>
```

Listing 3 AmexCC.cfc

```
<cfcomponent displayname="AmexCC" extends="CreditCard" hint="Base Class
for Credit Card CFCs">

<cffunction name="VerifyAddress" access="public" output="false" return-
type="boolean">
    <cfargument name="addressData" type="string" required="true" />
    <cfparam name="arguments.addressdata.name" type="string" />
    <cfparam name="arguments.addressdata.street1" type="string" />
    <cfparam name="arguments.addressdata.street2" default="" type="string" />
    <cfparam name="arguments.addressdata.city" type="string" />
    <cfparam name="arguments.addressdata.state" type="string" />
    <cfparam name="arguments.addressdata.zip" type="string" />
    <!--- In the real world you would actually do something here --->
    <cfreturn true>
</cffunction>
<cffunction name="processCCtrans" access="public" output="false" return-
type="string">
    <cfargument name="transactionData" type="string" required="true" />
    <cfparam name="arguments.transactionData.name" type="string" />
    <cfparam name="arguments.transactionData.ccnumber" type="string" />
    <cfparam name="arguments.transactionData.amount" type="numeric">
    <cfparam name="arguments.transactionData.expiration" type="date">
    <!--- In the real world you would actually do something here --->
    <!--- Instead a random determination of approval --->
    <cfif len(arguments.addressdata.amount) GT 100>
          <cfreturn "Decline">
    <cfelse>
          <cfreturn "98765432">
    </cfif>
</cffunction>
</cfcomponent>
```

Listing 4 ProcessCreditCard.cfm

```
<!--- Determine if this is a Submit event --->
<cfif IsDefined("form.CreditCard")>
<!--- Instantiate a CFC for the credit Card being processed --->
<cfif Not ListFindNoCase("AmexCC,VisaCC,DiscoverCC,MasterCardCC",
form.CreditCard)>
    <cfabort showerror="The Component ""#form.CreditCard#"" does not exist
          or is not authorized!">
</cfif>
<cfobject component="#form.CreditCard#" name="myCreditCard" />
<!--- Create a data structure to pass in to the processTransaction()
     method that will be passed through to the verifyAddress() method --- >
<cfset addressStructure = structNew()>
<cfset addressStructure.name = form.name>
<cfset addressStructure.street1 = form.street1>
<cfset addressStructure.street2 = form.street2>
<cfset addressStructure.city = form.city>
<cfset addressStructure.state = form.state>
<cfset addressStructure.zip = form.zip>
<!--- Create a  --->
<cfset creditCardStructure = structNew()>
<cfset creditCardStructure.name = form.name>
<cfset creditCardStructure.ccnumber = form.ccnumber>
<cfset creditCardStructure.expiration = form.expiration>
<cfset creditCardStructure.amount = form.amount>
<!--- If a Security Code has been passed in via the form then add it to
    the data structure to be passed through to the processCreditCard()
          method --->
<cfif form.securityCode neq "">
    <cfset creditCardStructure.securityCode = form.securityCode>
</cfif>
<!--- Invoke the processTransaction() method passing in the two structures
--->
<cfinvoke component="#myCreditCard#"
           method="processTransaction"
             returnvariable="approvalCode">
    <cfinvokeargument name="addressData" value="#addressStructure#">
    <cfinvokeargument name="creditCardData" value="#creditCardStructure#">
</cfinvoke>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <title>Credit Card Transaction</title>
</head>
<body>
<cfif approvalCode eq "Declined">
    We are unable to process your credit card transaction.
    <br><br>
    Your credit card has been declined.
<cfelseif approvalCode eq "Cannot verify address">
    We are unable to process your credit card transaction.
    <br><br>
    We cannot verify your address.
<cfelse>
    Your approval code is:
<strong><cfoutput>#approvalCode#</cfoutput></strong>
    <br><br>
    Please print this page for your records.
</cfif>
</body>
</html>
</cfif>
```

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

How to Find a Good ColdFusion Hosting Company

Finding a good ColdFusion hosting company is a challenge for many ColdFusion developers who don't have access to their own servers. This month, ***CFDJ*** interviews CFDynamics partner Jason Egan for the scoop on the company's ColdFusion hosting services.

Interview By
Rob Brooks-Bilson

CFDJ: **How did CFDynamics get started?**
Egan: CFDynamics is a division of Konnections, Inc. – we've been in business for over eight years. We have been hosting CF applications for that entire time on a limited basis, and started hosting CF publicly as CFDynamics.com about two-and-a-half years ago.

CFDJ: **Why ColdFusion hosting?**
Egan: We saw a gap in the ColdFusion community that needed to be filled – too few experienced hosts, all overpriced.

CFDJ: **What versions of ColdFusion do you support?**
Egan: ColdFusion 5.0 and MX for Windows, and we hope to have Linux versions of ColdFusion MX available in the future as well.

CFDJ: **And what Web server(s) do you run on your CF5/CFMX Windows machines?**
Egan: We're running IIS 5 servers.

CFDJ: **What do you feel separates you from other companies that offer ColdFusion hosting?**
Egan: We were a development firm before we were a host – so we have a greater understanding of what is required in a hosting environment from a customer and developer standpoint. We didn't get into the hosting business because we thought there was a "quick buck" to be made.

Rather, as developers, we saw there was a lack of quality hosting in the ColdFusion community and we think we have what it takes to fill that gap. We did a lot of research before we started offering hosting to the ColdFusion community. We also spent an extensive period testing services by hosting for well-known members of the ColdFusion community, receiving feedback, and fine tuning services until it was just right. This is why we have excellent services and offer so much with our hosting packages, like unlimited e-mail accounts, tons of disk space and bandwidth, and MS SQL Server access with every account!

CFDJ: **You say that one of the things that sets you apart from other hosting providers is that you have a true understanding of what ColdFusion developers are looking for in a host. An overriding theme in the conversations I've had with other ColdFusion developers regarding ColdFusion hosting has been a lack of timely support. How does CFDynamics address this issue?**
Egan: In general we have an excellent record of quick turnaround on support issues. Providing intelligent and friendly support is very important to us. That is why we have gone to such great lengths to provide a friendly support staff that is knowledgeable in a wide variety of Web technologies, server hardware, and Internet protocols. We take time to help the "newbie" and have the experience to help developers.

CFDJ: **In terms of hosting ColdFusion sites, what aspect do you find most challenging?**
Egan: I would have to say the most challenging part of hosting is finding out which customer may be causing a problem on a server, let alone debugging code as we help a customer determine why their code doesn't work or causes problems. Just about everyone has their own style of coding; with so many ways to get the same results, and then toss another language into the mix... the list goes on and on. We have developed some proprietary tools that help us pinpoint problems and easily take care of them, in most cases before customers are even aware there is a possible problem.

CFDJ: **Can you give our readers an overview of the types of hosting plans you offer, as well as the costs?**
Egan: We offer simple hosting accounts for basic dynamic sites with an Access or MS SQL Server database back end starting at as little as $16.95/month, up to hosting plans with a ton of disk space and bandwidth to meet the needs of the most demanding customers, starting at just $49.95/month. We also offer dedicated hosting for those who require greater control over their site.

"We are committed to continuing our extensive involvement with Macromedia partners"

—Jason Egan, CFDynamics partner

***CFDJ:* Going forward, how do you see CFDynamics growing, and what additional services/offerings are on the horizon?**

Egan: We see ourselves growing by continuing to stay involved with the CF community and by adding new and innovative services. One thing on the horizon is a completely custom control panel that is built from the ground up, and chock-full of customer suggestions.

We've recently added even more bandwidth from four major providers, and plan on implementing a redundant/clustered server network environment for our customers. We are committed to continuing our extensive involvement with Macromedia partners such as PaperThin and AbleCommerce, as well as with developers, to provide exciting and innovative products and services to our customers at reasonable prices.

***CFDJ:* If you could ask Macromedia for one enhancement to ColdFusion that would make your life as a hosting provider easier, what would it be?**

Egan: Better logging tools – making it easier to separate the logs by domain rather than having to sort through the logs for a single incident like a needle in a haystack. Thankfully we have automated tools that help with this.

***CFDJ:* Who can readers contact for more information, or to open an account?**

Egan: Sales@cfdynamics.com, or sign up online at www.cfdynamics.com.

About the Interviewer

Rob Brooks-Bilson is a senior manager at Amkor Technology, where he has worked since 1996. Rob's involvement with ColdFusion goes all the way back to version 1.5. He is the author of Programming ColdFusion *(O'Reilly). Rob is a frequent speaker at ColdFusion user groups and conferences, and is also a Macromedia Certified Advanced ColdFusion Developer.*

rbils@amkor.com

Blueprint 1.5 from Ivis Technologies, Inc.

1.0 application security and localization services for your CF applications

Over the last nine years, I must have coded at least 100 ColdFusion applications. Many of these contained similar features – calendaring, e-commerce, content management, and dynamic survey forms. Several were "one-offs" – based on work done for previous customers with some modified behavior and rebranding. Others were architected to work in either a dedicated or "Application Service Provider" mode. All of them required an administrative interface with an accompanying security model that required a significant amount of time to reimplement.

Reviewed By
Steve Drucker

Blueprint, from Ivis Technologies, Inc., provides the aforementioned services of application security and localization tools as a single coherent framework. In addition, they provide a slick, "out of the box" GUI (see Figure 1) for the administrative components of your custom software. In many respects, this software successfully provides the security services that CF 5.x "Advanced Security" and certain components of Allaire Spectra failed to deliver on.

Installing Blueprint

Installing Blueprint went reasonably well, relative to other CF-based products that I have tested. The installer not only copied the appropriate files into the proper locations, but it even went so far as to create a new, empty database on my SQL server and proceeded to write out its table schema.

It failed, however, while attempting to use the "sa" account to accomplish this task (using an account other than "sa" worked fine and Ivis later issued a patch). Upon further inspection, however, I found that the product does not make use of Transact-SQL stored procedures – a definite oversight considering that SQL Server is the only database platform currently supported. Also, for a product that concentrates on application-level security, you would think that passwords would be encrypted in the database. Sadly, this is not the case.

Using the Blueprint Administrator

After installation I was able to navigate to the Blueprint administration area and log in. I found their HTML/DHTML user interface to be quite intuitive, and was quickly able to register a new application, create new users and groups, and administer permissions. Blueprint has a very clean interface for modifying user account information, restricting access by IP range (see Figure 2), expiring accounts on a particular date, and logging user actions.

Deploying with a Preexisting CF Application

Integrating Blueprint with my preexisting application was a fairly painless process involving wrapping preexisting sources with a single tag – cfblueprint>…. </cfblueprint>. Their online tutorial was quite helpful in this regard, detailing their preferred naming schema for CF. While using the Ivis methodology to name your files is helpful, it is not required to leverage the security services. Also bundled with the product are examples detailing the ease of deploying Blueprint as a wrapper for preexisting applications using the Fusebox coding style.

Branding and Localization

The second core strength of Blueprint revolves around localization. The product provides comprehensive facilities for dynamically swapping out images and text based on country of origin and/or client login as depicted in Figure 3. In order to facilitate the dynamic manipulation of colors and images, the system comes bundled with additional third-party software – ImageMagick. ImageMagick is implemented as a set of DLL files that are invoked by Blueprint through the use of <CFEXECUTE> tags. This causes me some concern over future platform compatibility and scalability.

(Very) Platform Specific

Perhaps the biggest drawback for Blueprint is platform dependence. Currently, the system will run only on ColdFusion MX/Windows and Microsoft SQL Server. Ivis is readying a new release that will include broader support for operating systems and databases. The lack of support for authentication via

Figure 1: Registering application permissions

Figure 2: Rich functionality for restricting account access

Figure 3: The entire UI can be easily rebranded and localized through "themes"

Active Directory or LDAP, however, compromises its value as a security framework.

Now that Blueprint can free me from the drudgery of reintegrating or building my own custom application security model, I find that I have moved on to the new frontier of Flash development which, unfortunately, is largely unsupported by their product. For while Blueprint was written specifically for ColdFusion MX, their CFC API layer is currently undocumented, encrypted, and not available to be consumed as Web services through Flash Remoting.

Final Thoughts

Ivis's Blueprint is particularly compelling for CF developers who need to integrate security and/or localization into their CFMX applications, and have not yet established their own code base to supply these services. The product shows promise for essentially being a 1.0 release, and I look forward to reviewing future versions that may contain enhanced functionality, platform support, and broader appeal.

Vitals: Blueprint v.1.5

Ivis Technologies, Inc. • 2425 East Camelback • Suite 950 • Phoenix, AZ 85016
Phone: 602-346-5044 • Fax: 602-346-5049 • www.ivis.com

Dell Latitude C640, 512MB RAM 2.2GHz P-4
ColdFusion MX, SP2 • Windows 2000, SP2 •Microsoft SQL Server 2000, SP2
Standard Edition: $795.00 USD (one server, one domain)
Professional Edition: $1995.00 USD (one server, five domains)

Target Audience: Developers looking to add a consistent application-level security and "identity" services to their applications.

Pros: (Relatively) low cost • Easy to use/deploy application security services
Cons: Limited platform support (MX/Win/SQL Server) • No current support for Active Directory • Limited functionality • Initial release (despite v. 1.5 moniker)

Client Platform: Administrative GUI supports Microsoft Internet Explorer only.
Server OS: ColdFusion MX/Windows • Database Support: Microsoft SQL Server 7/2000

About the Reviewer

Steve Drucker is the CEO of Fig Leaf Software, a Macromedia premier solutions and training partner with offices in Washington, DC, and Atlanta, GA. He is also a certified Macromedia instructor and MM-certified Dreamweaver, Flash, and Advanced ColdFusion MX developer.
sdrucker@figleaf.com

Flash for CFers: Getting over the Hump

Resources for Web app developers

Where do you stand in the march of CF developers moving toward implementing Flash interfaces in your Web apps? Are you gung ho? Still on the fence? Dead set against it? Feeling left out? Left behind?

By Charlie Arehart

In this month's **Journeyman ColdFusion** column, I'd like to take up the cause of Flash for CFers. In particular, if you're a developer who's been feeling left out or left behind, or simply challenged to understand most articles and books on the subject, take heart. In this article, I will share some observations and insights to help get you on the right path.

As I say in my user group seminars on this particular topic, it's like there's a wall you need to get over to get started. It's not that high, but just high enough to make it hard to get started on your own without a little leg up. Once you get that leg up, however, there's a whole world of cool possibilities.

The Challenge of Getting Started: It's Not Your Fault

Have you been impressed with Flash integration, but felt that it was beyond your reach? Have you tried to follow along with Flash Remoting articles, but felt stumped by your lack of Flash experience? Have you tried to read beginner Flash books, only to be frustrated by their focus on drawing, graphics, animation, sound, and video? Or by their wading through *every* part of the Flash interface?

The challenge of getting started with Flash as a Web developer is *not* your fault. It's partly due to Flash's heritage as a design tool, and the recent addition of many features for developers. Similar challenges exist for Dreamweaver MX. Bringing together two worlds, designer and developer, has left many resources challenged as to how to present fundamentals to each audience.

Also, Macromedia's showing the Pet Market and Broadmoor examples might also have set high expectations, leaving the average newcomer to believe creating Flash interfaces must be difficult. Some have tried to propose that CFers should focus on the back end and leave Flash to designers. On the other hand, those complex examples may have even lowered the expectation of some, thinking such Flash-based replacements of complex processes are not a suitable opportunity to add to their site.

You *can* integrate Flash easily and effectively. It's just a matter of finding simple samples and walkthroughs, trying them out for yourself, and identifying suitable next steps and resources for learning more from this perspective.

Some Good Starting Points for Web App Developers

Note that I said the first step is "finding simple samples and walkthroughs." In this article, I'm not going to actually walk you through getting started with the Flash interface. I'm not going to show any screenshots nor a single line of ActionScript. Frankly, an excellent guide to getting started was written by our own Ben Forta in the November 2002 issue of ***CFDJ***.

In "Data Entry reFORMed," available at www.sys-con.com/coldfusion/article.cfm?id=528, Ben solves the challenge in a way that few have. He doesn't presume any prior Flash experience, but he walks through the basics of using the interface to build a typical "two-selects related" interface that's hard to do in HTML. Even with JavaScript, it could involve caching large amounts of data on the client and may not work on all browsers.

This sort of solution is easy in Flash, and he walks through building it without a lot of focus on needless details. Yet he doesn't leave anything out that you might need. Too many articles soon drop a "place this code in the first layer of the first frame," but if these are foreign terms to you, then the wall has been raised. If you visit that page and use the "print article" link, you'll have an excellent resource that gets you started in just 2300 words. From there, you should be able to proceed to other resources, appreciating all the articles on Remoting and using Flash for Web apps.

Unfortunately, other resources aren't as well-suited to us. For instance, many articles

recognize that readers may not have enough Flash knowledge and suggest that you "read the manuals." The problem is, they're really too designer oriented. As I said before, they tend to focus on aspects of using Flash that may not interest us, at least not from the start.

While it may seem counterintuitive, I don't recommend that you start with the Flash MX Manual, *Using Flash MX*. At least not at first. When you're ready, it's available at Help>Using Flash. There are also tutorials and samples available from the Flash help menu, but those, too, tend to be focused on using Flash in ways that may not immediately seem related to our needs as Web app developers.

Even Ben's book, *Reality ColdFusion: Flash MX Integration*, while seemingly the perfect starting point, presumes that you already understand Flash. The good news is that his article, and a couple of others I'll recommend, will help get you quickly up to speed to appreciate that book and several others, including the very popular and highly regarded *ActionScript: The Definitive Guide* (recently released in an MX version) and *Object-Oriented Programming with ActionScript*.

There is one other book that I'll recommend, and it may surprise you. Most will know that the *ColdFusion MX Web Application Construction Kit* is the latest edition of what's regarded by many as the seminal introduction to CF. Intermediate and advanced developers may spurn it, thinking if they already have one of the earlier editions that they are perhaps above getting it again. But the book has an excellent chapter, 23, on Flash integration. It was written by Nate Weiss and it's a great introduction to Flash and CF integration.

The chapter covers far more than Ben's article could (he has a whole chapter) and similarly, Nate never loses you nor bores you with Flash interface details you don't need. He builds a few simple but useful examples of Flash/CFMX integration, and even shows a simple, useful application of Flash's animation abilities. He also offers examples of server integration, using techniques other than Remoting. This chapter alone may justify the cost of the book – $35 at Buy.com at the time of this writing.

Other Good Resources

Once you've gotten over the humps explained by the resources above, you should be able to appreciate and take advantage of the many articles both in past issues of *CFDJ* and the Macromedia DesDev center as well as many other sites. They may have seemed daunting if you read them in the past, but would now be worth your consideration with just that little bit of Flash experience you didn't have before.

The *CFDJ* articles on Flash and CFMX integration have included, starting with the most recent:

- "Bridging the Gap Between Flash and ColdFusion MX," by Simon Horwith (Vol. 5, issue 2) at www.sys-con.com/coldfusion/article.cfm?id=570
- "Web Services in a Flash," by Dennis Baldwin (Vol. 5, issue: 1) at www.sys-con.com/coldfusion/article.cfm?id=556
- "The Flash MX Calendar Component," by Mike Britton (Vol. 5, issue 1) at www.sys-con.com/coldfusion/article.cfm?id=555
- "Flash Remoting with Macromedia's DesDev Feed," by Dennis Baldwin (Vol. 4, issue 12) at www.sys-con.com/coldfusion/article.cfm?id=540
- "Get Connected with Flash Debugging," by Dennis Baldwin (Vol. 4, issue 10) at www.sys-con.com/coldfusion/article.cfm?id=513
- "Flash Up Your Forms with Components," by Dennis Baldwin (Vol. 4, issue 6) at www.sys-con.com/coldfusion/article.cfm?id=451
- "Building the Development Team in Flash MX and ColdFusion MX," by Kevin Towes (Vol. 4, issue 6) at www.sys-con.com/coldfusion/article.cfm?id=448

There have also been some great introductory articles at the Macromedia DesDev center. Again, at first reading, without a little Flash experience, they may have seemed daunting and beyond you, but give them a look after getting just the little bit of experience offered by the two resources discussed in the previous section. They include:

- "Macromedia Flash MX Remoting: Open the Gate for Rich, Dynamic Content," by Kevin Towes at www.macromedia.com/desdev/mx/coldfusion/articles/remoting.html
- "Getting Started with Flash Remoting," by Mike Chambers at www.macrome dia.com/desdev/mx/coldfusion/articles/startremoting.html
- "Flash Remoting Basics for ColdFusion Developers," by Lucas Sherwood at www.macromedia.com/desdev/mx/flashremoting/articles/cfexample.html

You can find even more articles at the various DesDev centers, including the following:

- www.macromedia.com/desdev/mx/flashremoting
- www.macromedia.com/desdev/mx/coldfusion

- www.macromedia.com/desdev/mx/flash/
- www.macromedia.com/desdev/mx/flashcom/
- www.macromedia.com/desdev/rich_media_ads/

The last one is quite recent, having been opened just in January.

There are also lists of examples of such integration, including articles (some of which are a little more advanced) available at www.macromedia.com/desdev/topics/sample_apps.html. There's also a page devoted to resources specifically for Flash/CFMX integration at www.macromedia.com/support/flash/programs_cf.html.

There are even a number of online video presentations that walk you through the use of Flash, including several related to CF integration, available at www.macromedia.com/software/flash/productinfo/tutorials/gettingstarted.

One of those videos is on the accessibility features of FlashMX. This is a topic of significant importance to many developers and administrators, and Flash MX does add new features to address this important aspect of creating usable applications. Indeed, Macromedia has a site devoted to the topic, at www.macromedia.com/macromedia/accessibility/features/flash. Still one more non-Macromedia resource devoted to the goal of promoting usable Flash sites is a "Wiki" at www.whatisflash.com. The topic is also discussed in a couple of books, including *The Flash Usability Guide*, by Chris McGregor, et al. Indeed, a useful sample chapter is available online at www.friendsofed.com/books/flash_mx_titles/mx_usability/sample_chapter.pdf. Another is *Constructing Accessible Web Sites*, by Jim Thatcher, et al.

If you'd like to take a course on the topic of both Flash and integration with CFMX, Macromedia has recently released the class, "Developing Rich Internet Applications." It's an intense three days covering lots about ActionScript, fundamentals of the Flash interface, and integration with CFMX. Simon Horwith wrote about the class in last month's aforementioned "Bridging the Gap Between Flash and ColdFusion MX." Learn more about the class at www.macromedia.com/support/training/instructor_led_curriculum/dev_apps_fmx_cfmx.html.

What If You're Not Using CFMX or Flash MX?

While it would be best to leverage the power and improvements in CFMX and Flash MX, the reality is that some people simply can't or won't yet have migrated. Or maybe you have co-workers or clients interested in doing this integration with servers that do not support Remoting (it is currently supported only on CFMX, ASP.NET, and certain J2EE servers).

What if you're on CF5? Or 4.5? Or PHP? Or Perl? Or some mainframe Web server, for that matter? Or still using Flash 5? You may be led to believe that you're out of luck and unable to play the game of Flash/CF integration. You can! It's just not as easy as it is with CFMX and Remoting. The good news is that we at *CFDJ* have been covering Flash and CF integration for nearly two years now. Check out any of the following:

- "Macromedia Flash 5 and ColdFusion 5," by Mike Chambers (Vol. 3, issue 11) at www.sys-con.com/coldfusion/article.cfm?id=366
- "ColdFusion-Driven Flash Content," by R. Drisgill and J. Montilla (Vol. 3, issue 9) at www.sys-con.com/coldfusion/article.cfm?id=347
- "Getting CF into Flash," by Randy H. Drisgill and Jason Montilla (Vol. 3, issue 8) at www.sys-con.com/coldfusion/article.cfm?id=346
- "ColdFusion Meets Flash," by Dennis Baldwin (Vol. 3, issue 8) at www.sys-con.com/coldfusion/article.cfm?id=309

Also note that the approaches described apply to pretty much any Web application server. It's worth noting, however, that if you have Flash MX it offers even more powerful new ways of integrating with Web app servers even without Remoting, in the form of the new LoadVars and XML objects. Do look into these if you're not using a server that supports Remoting. The bottom line is that it's pretty easy to pass data back and forth from a Flash movie on the client up to a server of any kind.

"It's just a matter of getting that gentle leg up over the wall that then opens to the many paths to creating Flash-enabled Web applications and interface widgets"

Other Next Steps

The focus to this point has been on resources that would be especially suitable for CF and other Web app developers who might otherwise find typical "beginner Flash" resources to be inappropriate. But as I said before, once you get over the initial hump of being shown the most fundamental aspects, your next step is to learn how to use the rest of the features of the Flash interface, such as layers, frames, animation, more about UI components, more about ActionScripting, debugging, troubleshooting, and more.

An excellent starting point once you're prepared to dive into more about Flash would be *Foundation Macromedia Flash MX*, a "Friends of ED" book. In fact, the first chapter of the book is available online at www.friendsofed.com/books/foundation/flashmx/Found_MX_Chapter_01.pdf and would be a great resource even if you don't get the entire book (though I'm confident that chapter will motivate you to get it).

A couple of other books that do a surprisingly good job of introducing fundamentals (including some server integration) are *Macromedia MX eLearning; Advanced Training from the Source*, and the recently released *Mobile Macromedia Flash MX*. Even if you're not interested in doing development for mobile devices, it's interesting to consider that the skills and efforts applied to building Flash apps for browsers can be applied just as readily to mobile devices (phones, PDAs, and more).

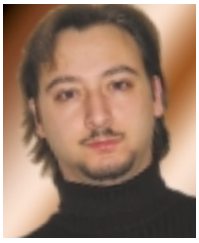
Still other things you may want to explore once you're "over the hump" are Macromedia's "Developer Resource Kits" (DRKs), which offer several useful new components, demonstrations, and more (mostly Flash-oriented). Learn more at www.macromedia.com/software/drk.

—continued on page 39

ColdFusion Server Recovery

Simple solutions to the four most common ColdFusion Server problems

Have you ever built that "killer" ColdFusion app only to have server problems bring your application to a grinding halt? Fortunately, armed with a little knowledge and a few batch files, most issues in ColdFusion 4.0–5.0 are easily resolvable in less than 60 seconds.



By Vlad A. Friedman

Many of these scripts are taken directly from the edgewebhosting.net internal support archives and have been created after many years of troubleshooting ColdFusion servers. The batch files in this article as well as several other scripts and utilities used for recovery can be downloaded at: http://edgewebhosting.net/quickfix.

Even if you don't have a dedicated server, you should still know how to perform basic ColdFusion server recovery procedures. You may need to recover an in-house server, or offer a little friendly advice to the support tech on the line who may not know how to resolve a problem with your shared hosting service.

This article assumes a Windows-based platform utilizing Microsoft IIS as the Web server, one of the most common environments for hosting ColdFusion Web sites.

Problem 1:

Applications are running very slowly, and opening Task Manager reveals cfserver.exe utilizing 90–100% CPU Utilization

- **Situation 1: The CFMAIL Bug**
  The most common problem we have seen to date has been around since version 4.0 and exists through version 5.0. It's called "The CFMail Bug."
  Background: Every time a CFMail tag is called, a mail file with an extension of CFMAIL is saved into a folder on the CF Server's local drive c:\cfusion\mail\spool. If ColdFusion is unable to complete writing this file, it will generate a file less than 1Kb in size that either has no data or just empty spaces.

  At this point the dll used by the server to process and send mail, dart.dll, will lock up and go into an infinite loop. This causes the cfserver.exe (ColdFusion Application Server) to lock at 90–100% CPU utilization. Restarting the ColdFusion Application Server service or rebooting the computer seems to have no effect since the affected file remains in the mail spooler's folder.

  Solution:
  1) Open the Services dialog
     a. NT40 Control Panel -> Services or
     b. Windows 2000 -> Control Panel -> Administrative Tools -> Services
  2) Stop the ColdFusion Application Server Service
  3) Stop the ColdFusion Executive Service
  4) Open a DOS window and cd to your \cfusion\mail\spool folder
  5) Remove all of the files from this location (del *.*) or move them to a different folder for review later
  6) Restart the ColdFusion Application Server Service
  7) Restart the ColdFusion Executive Service
  8) Review the files in the temp folder you just created, removing any invalid files (these files may be opened with NotePad)

  You can also save and run the following batch file. Make sure to replace the path with the correct path to your CFusion folder on your server.

```
REM Contents of fixmail.bat
NET STOP "Cold Fusion Application Server"
NET STOP "Cold Fusion Executive"
CD \CFUSION\MAIL\SPOOL
MKDIR TEMP
MOVE *.* TEMP
NET START "Cold Fusion Application Server"
NET START "Cold Fusion Executive"
```

- **Situation 2: Variable Deadlocks**
  Background: Variable deadlocks occur when several processes or users try to update the same memory in a Server, Application, or Session variable at the same time. On a site that receives a large amount of traffic, variable deadlocks on one site can cripple performance for the entire server. As I am sure you have read in several *CFDJ* articles, I cannot stress enough the importance of utilizing CFLOCK effectively within your applications. (For example, see "Super Wizards: Using ColdFusion for Data Manipulation and Processing," by Courtney E. Payne; ***CFDJ***, Vol.4, issue 12.)

  Solution: If you suspect that you have variable deadlock conditions causing your server to go haywire, log in to your CF Administrator and enable "Full Checking" on the Server, Application, and Session scopes in the Locking section of the configuration. After you have made this change, ColdFusion will throw an error if you try to use a shared access variable without a lock. It's not perfect, but it will catch most missing locks.

  Another option is to simply use Studio's multiple file search for the strings "server", "session", and "application" and ensure that you are using CFLOCK with all of your shared variables. This may take a while, but consider it a painful reminder that you *must* lock your code. A quick browse of your Application.log (\cfusion\log\application.log) will quickly reveal which CFM files and variables are causing the problem. You can then disable "Full Checking" and fix the code within the application causing the problem.

- **Situation 3: ColdFusion and Microsoft SQL Server installed on the same server**
  Background: ColdFusion and MS SQL server are both very resource-intensive applications. In a high-traffic environment, both applications start to contend for the available resources on a server.

Solution: If you can possibly avoid it, do not put ColdFusion and SQL server on the same machine if you intend to run in a mission-critical production environment. You will have a significantly more stable server if SQL is on a separate machine, even if that SQL machine maintains a higher CPU load than your CF Server. If this situation is unavoidable, modify the properties of your SQL server configuration to reduce the amount of available memory resource and CPU worker threads available to SQL Server in the Memory and CPU tabs within the SQL Server Properties in Enterprise Manager. Set the values to a necessary level for your application to avoid overloading your server. Usually 75% of the default values will produce effective results.

Problem 2:
The dreaded 500 IIS Internal Server Error

Background: A 500 Internal Server Error is a generic error thrown by the server when it decides it cannot access an internal resource it thinks necessary to display your page. Although this is a much more common problem when developing applications using Active Server Pages, it does occur sporadically on all versions of ColdFusion Server. Although it is not specifically a ColdFusion problem, but an IIS problem, it still occurs often enough that CF developers should know how to solve it.

Solution: The following steps should resolve the issue:

1. Open Internet Services Manager
   a. NT40 Start->Programs->Windows NT 4.0 Option Pack->Internet Information Server -> Internet Services Manager or
   b. Windows 2000 -> Control Panel -> Administrative Tools -> Internet Services Manager
2. Right click and open the Properties of the Web site with the problem
3. Click on the Home Directory Tab
4. Click the Unload Button

If you are hosting in a shared environment, call your Web host and ask them to "Unload your site in IIS." They should be familiar with the process.

Problem 3:
ODBC Errors

Background: There are several bugs in ODBC that can cause ColdFusion to stop processing one or all MS Access databases on your server. You may receive errors such as:

- The Microsoft Jet database engine cannot open the file "(unknown)". It is already opened exclusively by another user, or you need permission to view its data.
- Not enough space on temp drive
- Memory allocation error

Several other errors may occur while accessing your MDB file.

Solution: Although this can be caused by several different conditions, most often all that is required is a restart of your ColdFusion Application Server service. If that does not alleviate the problem, most likely your MDB file is corrupt, the database requires a password, or the ColdFusion Server does not have the correct NTFS permissions to access the MDB file. If restarting the service does not resolve your issue, see http://edgewebhosting.net/quickfix/mdb for a set of utilities to repair MDB files or identify the presence of an MDB password.

Please be aware that Macromedia and Microsoft do not recommend using Microsoft Access databases for production environments. We've found that once you start receiving more than 25 concurrent users to your Web application, this problem will reappear much more frequently.

For more information, see Macromedia's notes on the subject: www.macromedia.com/v1/Handlers/index.cfm?ID=1540.

Problem 4:
Cannot access the system registry

Background: Any time you attempt to access client variables, add a new data source, or configure just about any setting in the CF Administrator, you receive the error "Cannot access the system registry." As terrifying as this error may sound, it is easily resolvable in most cases.

This problem occurs if your server is configured to store client variables in the system registry. By default, the variables are not purged for up to 90 days. On a server that receives moderate to heavy traffic, the registry can become full quite quickly. Once you have reached Windows' maximum allowable registry size, no applications on the server will be able to write additional data into the registry.

Solution: The first step of this solution is to purge the client variables from the registry. This may be accomplished by downloading Macromedia's utility on their support site: www.macromedia.com/v1/handlers/index.cfm?ID=20340.

Unfortunately, the Macromedia-provided utility may run for several hours to clear the entire client variables registry key. We have written an application to perform this process much more quickly: http://edgewebhosting.net/quickfix/registry.

The second step is a little more difficult. It is highly recommended by Macromedia that you not store client variables in the registry in a production environment. Client variables should always be stored in a database when used in a production environment; preferably a SQL Server database. If this option is not available to you, use the Macromedia TechNote to update your registry to ensure that your variables expire in a timely fashion: www.macromedia.com/v1/Handlers/index.cfm?ID=22419

Conclusion

Although CF recovery techniques and problems can be significantly more complex than the most common problems listed in this article, you'll find that with the tricks included here and the utilities provided on our Web site, you should be able to quickly recover from the majority of server problems. ColdFusion has matured as a development environment. Since its inception, each new release has added a new level of stability. Part of building any stable application platform involves understanding the problems to which your environment is susceptible, as well as the recovery procedures, in case the worst happens. I hope this article will help bring a higher level of reliability to your applications.

About the Reviewer

*Vlad A. Friedman is the CEO of Edgewebhosting.net, a Macromedia Alliance Hosting Partner and a **CFDJ** Readers' Choice Award finalist. Edgewebhosting.net provides both Managed Dedicated and Shared Hosting solutions for CF developers and their clients.*

vlad@edgewebhosting.net

A Developer's Story Part 2

A better way of building software

In last month's article, I introduced you to a company with a project in jeopardy. When I was called in to analyze the situation, I discovered that the project was a major revision; the first version had been very successful, but now the company was running into problems. I promised the CIO that I would make a presentation to his key development team. In preparation for that meeting, I found myself at a local store, Frank's House of Models…

By Hal Helms

Frank led me to the back of his small store, filled to the ceiling with models, some of which were so old that they must have been made when I was a kid. "Let's see," said Frank. "You're looking to build a model world – is that it?"

"Yes," I told him. "I want to illustrate to a group of businesspeople and programmers the idea of a scale model as a successful development paradigm." Paradigm? Had I really said that? Who spoke like that? "You know – a better way of building software."

"Well, I don't know anything about software. It seems like every time I use software, it's either crashing or fighting me. I bought an accounting system for my business. Here..." He pointed vaguely at the store. "But I don't know, it just doesn't work the way I do. It's my fault, no doubt. But I do know a good deal about models."

I spent the next half hour with Frank as he took me on a tour of his store. It turned out that some of the models were decades old. But they were in their original shrink-wrapped boxes and were highly collectible, their prices reflecting this. However nostalgic the old models made me, I had to admit that the new models were far superior in fit and finish to the older ones. He showed me photos of some finished model cars. I really could not tell that these weren't real. The photos gave me an idea.

"Say, Frank," I said. "I want to ask you a favor. Do you suppose that I could borrow these photos for an hour or so? I want to make color copies of them at Kinko's."

"Well, there's one about a mile and a half down the street. But you just want copies?"

"Yes. I'm not a model maker, at least not models like these. I could never hope to get the accuracy of detail and the beauty of these. But these photos represent exactly the point I want to get across. I'd be glad to pay you for the use of the photos."

"You don't need to do that," Frank said. "Just be sure to return them."

"An hour, tops," I promised him. We picked out photos of models that Frank and his son had made: a Ferrari F60 racecar, a Ducati 996 motorcycle, and an F/A-18 Hornet attack fighter aircraft. I found the Kinko's and 40 minutes later returned with Frank's originals. "Say, while I'm here," I said, "do you suppose you could show me something that a novice could build? I'd love to try doing this with my son."

Frank was more than happy to assist me with finding just the right car. Of course, I needed some tools and supplies; after all, what's a craftsman without the proper tools? I may have taken this a bit too far though. Certainly when I paid the bill for my 1/8th scale Aston Martin DB5 James Bond car (with working weapons!) and enough tools and supplies to build the real thing, I decided that it would definitely be worthwhile trying to convince my accountant that since the purchase of the model had been inspired by a business need, it would be perfectly legitimate to write it off as a business expense. Until I got an answer, I thought there was no reason to trouble my wife with this particular matter.

Back at my hotel, I began to sketch out some ideas. I quickly found photos of the real counterparts of my models on the Web and sent these to Kinko's to have color-copied as well. I prepared more questions for my clients and decided perhaps it would be wise to call my accountant at home. After I explained the situation and my creative interpretation of this as an investment, Vicki laughed out loud. "All I can say is you better use this for a presentation; the IRS sort of frowns on the inspiration thing. Either that or you can explain to Susie why you spent – how much was it again?" These accountants! Have they no souls?

When the day for the presentation dawned, I had my photos, my PowerPoint – but no Aston Martin model. The CIO was present. The lead architect, the developers, and several of the project managers were all there. I began by passing out the photos in pairs – the model followed by the real thing. Then, ignoring the photos for the moment, I began my presentation.

"First," I said, "I want to say that it's obvious from the research I've done that there's a high degree of commitment from each person on the development team. This is not a case of not caring and certainly not one of incompetence. I know

that many of you are frustrated by the situation – working very hard, but with too few good results to show for it. If it was easier to pinpoint the problem, to find the "real culprit," I think we'd all be relieved.

"But it's not. When you created the first version of this software about three years ago, you did all the things you read about. You talked with the users. You interviewed your customers. You watched how they worked. And from this, you created a requirements document.

"That requirements document was clear and unambiguous. From talking with the end users – the ones who participated in the original process – they spoke highly of the project management and development team. When the software was delivered, it was exactly what they asked for. And you all were heroes."

"Oh, for the good old days!" said one developer, as everyone laughed.

"Then what is the problem?" asked a project manager.

"The problem is just that – you listened to the users and gave them exactly what they asked for."

"That's the problem?" she asked.

"Yes, I'm afraid so," I answered. "By concentrating on their requirements and then using a process of functional decomposition to ensure that each requirement was met, you were able to deliver a product that did exactly what the users wanted – at that time. Had this been a movie, we could have ridden into the sunset and had the screen fade to black.

"But it's not. Within three months after the software was delivered, changes to existing laws required that the software's internal logic be changed to account for these new requirements. Then, two months later, the company reorganized and new requirements meant new changes. But then you acquired another company and had to roll their data into your systems. That meant more reworking of the software."

"I knew it," another person said. "It was that damned merger – that killed us." Several people nodded in agreement.

"No, it didn't," I disagreed. "What's hurting us is that the software was designed for a single moment in time. The requirements were frozen and the software was built for one thing – to meet those requirements. We all know that life isn't static – but our software is. No one could have predicted the changes to the law or the merger or any of the changes that occurred..."

"But we could have predicted one thing – that changes would occur," said the CIO.

"Yes. Exactly that. We know that to a certainty. But you'd never know that from our software. And when I say, 'we,' I'm talking about the great majority of software that gets developed."

"How can we develop software then? When we know there will be changes but don't know what they'll be?" asked one of the programmers.

"Take a look at the photos I passed out. The first one is of a Ferrari F60. The next one is also of a Ferrari F60, but with a difference. The next one is a Ducati motorcycle, followed by another picture of the same model Ducati. The last one is an F/A-18 Hornet strike fighter. And then another Hornet."

"I don't get it," confessed the CIO.

"I don't either," said a project manager.

"Of the three sets of photos you have,

one of them is a photo of a real vehicle and one is a photo of a model."

"A model? Like a toy model?"

Thinking of the bill for my "toy," I responded, "Not a toy – a faithful scale model. They're remarkably accurate."

"Yeah, I can't see the difference," said one of the programmers. "But how does this affect us? We don't build models."

"That's just the problem. We're building software machines that do exactly what the user asks us to do – but that can't respond to the new things that users want to do.

"Software looks very different to all of us in this room than it does to end users. We know what goes into it. We know how hard it is to build. We know the technology that goes into it. We know so much, in fact, that we forget that to users, the interface is the application. When they see something on their screen – an invoice or a user record – they assume that they're looking at a real invoice, that the user they see is somehow alive in their computer."

"But that's crazy!" said one of the programmers. "The invoice is just a visual screen of information in a database. It's not alive."

"No, it's not. But users aren't that sophisticated. We show them an interface and they believe it. They believe it so much that they naturally want to do things with what they see – they want the software counterparts to act just like the real thing.

"And that's just what we do with good scale models. What is so engaging about the model vehicles in those pictures is how amazingly lifelike they are. You can turn the steering wheel and the wheels turn. You can open the hood and the engine is faithfully represented. You get the sense that if you could shrink yourself down to Lilliputian dimensions, you could get in it and drive away."

"My son does models," said the CIO. "Some of them are pretty astonishing. But how exactly does this relate to our software?"

"We don't think of ourselves as model builders. We think of ourselves as algorithm-makers or data-manipulators. But until we start designing software to better reflect the real world – until we build scale models of the real world – we're going to run into this disconnect, what engineers sometimes call an 'impedance mismatch.'

"Developers think they're creating screens and reports; users believe they're looking at the face of something much richer. So when the requirements of the real world change, they expect that the little invoices that live in their computer will change too."

"But building things in software is hard, harder than the real world sometimes," argued a programmer.

"I agree with you," I said. But some of the reason that it's hard is because our design is clouded by thinking in terms of databases and algorithms and not enough of the world as the end user sees it. It's a bit like viewing humans as a collection of DNA molecules. On a certain level, it's true, but it's at the wrong level. The wrong level for humans to interact. At the human level, we care about others' intentions, passions, thoughts, emotions.

"The problem with our software is that the user sees something – an invoice, say. But, as Gertrude Stein once wrote about Oakland, 'there's no there there.' There is no *invoice* that we can go to. The concept of an invoice has to be assembled from bits of data in various tables and different functions that operate on that data. The data you see on the screen is a trick, an illusion rather than arising from the *invoice* itself."

"You're speaking as if we really could build software that's alive – scale models that are alive. Is that what you're saying?"

"Not truly alive, no, but we can build faithful scale models. This is why object-oriented programming has swept the development world. Objects are those scale models. Before you start working on database table structures or work on algorithms, you determine what entities will inhabit the universe you're creating. You determine which ones will know about other ones. You decide how they will communicate; what each will be able to do.

"Object-oriented software development is about creating scale models of the real world. Data and algorithms are important, of course, but only to the degree to which they make the scale model more faithful in its representation of its real-world counterpart.

"What's important for us to see is that without this correspondence between real-world entities and software entities, we'll never be able to keep track of the complexity and the dynamic nature of the real world. What's important for us to see is that object-oriented development can help us build software that is more maintainable."

I turned to the CIO. "About what percentage of the total life-cycle cost of your code is used for maintenance?" I asked.

"It varies, but from 70–90% – something like that." The eyebrows of several people went up.

"That's in line with most studies I'm familiar with," I told him. "How many of you developers realize that?" Only a couple of hands went up. "The problems that you're having is because the software that's being built is great, but only if things don't change very much.

"But things do change, and we all spend an enormous amount of money and time and frustration trying to patch software together to keep up the illusion that the users want to see...that we've convinced them they should be able to see. They believe it's real, but we don't. And we don't build our software as if it was."

"You really think that an OO approach can help us?" asked the CIO. "I mean, we're not building Microsoft Word here."

"Look at virtually all modern computer languages and they're object-oriented. I mean, good grief, even languages like COBOL have OO extensions to them. It's not just a fad. Right now, OO development offers us the best chance for writing software that is successful in the long run.

"That's true of small-scale projects and of enormous-scale projects. Object-oriented design encourages a separation between the model that underlies everything and the view the users see, allowing either one to change independently of the other. It's not a magic bullet..."

"Nothing is," the CIO remarked.

"No, but it can make a very, very significant difference in how sturdy our software is. Or maybe sturdy isn't such a good word. Maybe how flexible, how malleable, how adaptable our software is."

"Does that mean we need to move to Java then? Or the .NET world?" a programmer asked.

"The choice of a language is a complicated one. But the language isn't a magic bullet either. I've seen a lot of Java code that is nothing but procedural code stuffed into classes. Larry Wall once joked that real coders could write assembly code in any language.

"It's not the language so much as the way we approach design. You're doing most of your work in ColdFusion. CFCs offer some of the benefits of object orientation. And even if you decide to incorporate other languages such as Java or one of the .NET languages, you'll need to go through the experience of learning to think in objects. That just takes time. And practice."

One of the project managers who had been silent then spoke: "When I was learning French," she said, "I had to work at it for a long time. At first, I was translating everything from English to French. It was slow and clumsy and frustrating. But then after a long time, something happened and I actually started to think in French. That's what I think of when you're talking."

"That's a perfect analogy," I told her. "That's exactly what it's like moving to designing OO projects. At first, we keep getting mired, worrying about implementation details. Because there's so much we have to know – technical stuff – it's completely understandable that we view the world from that level."

The project manager speaks again. "But you're saying that's like living at the DNA level. Or like me thinking in English, but trying to speak French."

"Yes. If we're to make real progress, and not experience the frustration of projects that can't be maintained, we have to make that switch to viewing the world in objects to thinking of ourselves as model makers, as creators of virtual universes."

We talked a good deal longer, and the longer we talked, the more noticeable a change in the tone of the room became. We had gone from being defeated to being hopeful. Finally, the CIO spoke up. "I say we try it. He's right: the world has gone to OO and there's a good reason for it."

"I'm up for learning something new," said the chief developer.

"If it will help my customers, I'm all for it," said a project manager.

"And you'll be able to help us with this?" asked the CIO.

"I sure will," I replied.

It was a great meeting, but as I got into my car, I remembered the scale model I bought – and all those tools and supplies! Oh well, I figured I'd just explain to my wife that this was sort of a very late or very, very early Christmas present! And…maybe a birthday present thrown in, too.

Wait a minute…I pulled out my cellphone and dialed my accountant. "Vicki, if I were to do research on a book or article, I could write that off, yes?"

"What kind of research?"

"Any kind. If I was writing about France, I could go to France, right? If I was writing about cooking, I could buy tools and stuff?"

"Oh, this is about that stupid model you bought. Well, yes, you could expense it – if you were writing on models. But you're a programmer, Hal, not a model maker. How could you ever end up writing about making models?"

How, indeed?

About the Author

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

hal@fusebox.org

Persistent Data Communications

Looking at the Flash Communication Server in a whole new light

With the recent addition of the Flash Communication Server to our already powerful arsenal of tools, ColdFusion MX developers now have the opportunity to create applications that maintain a sustained communication channel with other applications. So what does this mean, and how can it affect you?

By Kevin Towes

As an exercise, I decided to put CF and the Communication Server to the test and develop a simple application that allows multiple people to interact with a shared database in real time. As a bonus, I'll discuss continuing to use the database without being connected to it. That's right, pull that Ethernet cable from the back of your laptop, and take your database on the road – without requiring the server! Seem impossible? Read on.

In this article, I'll make extensive use of the Flash SharedObject programming model, Flash Remoting, and the Flash RecordSet Object. You will also learn useful techniques to help you understand what persistent server communications can offer an application.

To make it all happen I use a real mixed bag of Macromedia MX tools and servers – think of it as an MX soup:

- ***Servers:*** ColdFusion MX Server, Flash Remoting MX, Flash Communication Server MX
- ***Tools:*** Flash MX and Dreamweaver MX

Flash Remoting MX is used to transfer data between the Flash Communication Server and a ColdFusion database.

The Remote SharedObject is used to transfer database data between the Communication Server and each player connected to it. It is also how the real-time synchronization process occurs. Because the connection between a Flash player and the Communication Server is persistent, synchronization messages are easily delivered to each connected player and the server. When a change to a record is made, the record can be immediately locked on all players until the change has been completed and recorded.

The RecordSet Object is used to manage and maintain the integrity of the "rows" returned from the database. Using this object's native methods, you can easily manipulate and interact with the data collections, transferring them back and forth between ColdFusion, the Communication Server, and the Flash players.

Let's first set the stage with a fundamental statement that most of you should know: "Connecting with ColdFusion via a Web browser or using Flash Remoting is done over the HTTP or HTTPS protocols. HTTP is a nonpersistent protocol, meaning that when you request something using HTTP, a server will respond once. The connection is open only when the client requests something from the server."

This is great for a lot of IP-based applications, but is limited when you want the server to "message" a client that something has happened – without the client "asking" for an update. Consider today's instant messaging programs. Instant messengers like Yahoo!, MSN, and AOL maintain a persistent connection with the messenger servers and are updated immediately when someone comes online or goes offline. This is how you can see when a friend comes online.

You may have seen Flash Communication Server examples that include a live chat, or a shared whiteboard. These are simple invocations of the real power of persistent server connections. A few years back, a company called PointCast introduced "push" technology with its PointCast screensaver. This screensaver opened and maintained a channel to the PointCast server. This channel was used by the server to send real-time data to each screensaver that was on.

Flash Communication Server applications operate in a way that's very similar to the PointCast technology, except they make use of the ubiquitous Flash player. The user is not required to download and install any additional technology to start using it.

This is where the Flash Communication Server (Communication Server) can come in handy for applications that require instant notifications of events. The Communication Server doesn't use HTTP. It uses RTMP, a unique protocol that operates like a Java socket connection. It opens a persistent transportation channel between a Flash player and the Communication Server. The channel can be used to stream video and audio, or my favorite, to share data! Yes, the Communication Server is about much more than just audio/video streaming. This is where I want to help you see it in a new light.

Consider a shared address book database. When a database record is inserted, updated, or deleted, everyone using the database must reflect the new changes. With traditional ColdFusion applications, the user would be required to "reload" the browser page to show the updated data. This is because ColdFusion can't communicate the change to the user without the user first requesting an update. There are two problems here. First, the user must be connected to the application server and second, the user must be notified that there is a change in the database.

I'm going to attempt to solve these two problems using the Flash Communication Server and some new techniques for making

offline data available. There is a working example of the application at http://flashcom.pangaeaNewMedia.com. You can also download the source code and the sample database. I'm not going to get into every detail of the working application in this article, but I have heavily documented the source code so you can see what's going on.

For this exercise, I have a Communication application called "CFDJ" with all files (CFC, ASC, SWF, HTML) located in the folder c:\inetpub\wwwroot\flashcom\applications\CFDJ. I have a ColdFusion MX Server running on the same computer as my Flash Communication Server MX. This is a standard "Developer" installation of the Flash Communication Server. Your workspace may be different.

Flash Remoting, ColdFusion, and the Communication Server

This shared address book application will use Flash Remoting in a way that's different from what you're used to. You may have used Flash Remoting to transfer information between ColdFusion and a Flash application running in the Flash 6 player. This time, the same techniques will be used to transfer data from ColdFusion to the Flash Communication Server. There is no difference in the ColdFusion Component (CFC) code, nor is there a difference in the Flash ActionScript used to communicate with ColdFusion. The only difference is that the ActionScript will not exist within the Flash Movie, but on the Communication Server in Server-Side ActionScript (SSAS).

The application calls a CFC on the ColdFusion Server that returns a simple RecordSet Object to the Communication Server. Getting comfortable with the Flash RecordSet Object is important; by the end of this article, you will have a good idea of what you can do with it. Once the Communication Server receives the RecordSet from ColdFusion, the RecordSet items (records) are copied into a Remote SharedObject. I'll explain this shortly, but for now, simply understand that to share synchronized data across multiple Flash players, you must use Remote SharedObjects.

The ColdFusion Component (dataControl.cfc)

Before we get knee-deep into Flash, let's start with something familiar, the ColdFusion Component (CFC). There are two functions that ColdFusion will be responsible for in this project. The CFC's first function returns a full collection of all records from the table, "Contacts," in the data source, "articleDb". The function name will be called "getAllRecords". (FYI, this table contains the following fields: fName, lName, emailAddr, phoneNum, faxNum, notes, isLocked [type: yes/no], editTime [type:date/Time]). A field called recordID is the autonumber key/identity field. All fields are text except the ones identified differently. The second function in the CFC, doUpdate, is a simple data update method. There's nothing spectacular about these two functions. Just make sure the access attribute in the CFFUNCTION tag is set to "remote" (see Listing 1).

In the second function, "doUpdate", each field in the database is declared in a CFARGUMENT tag. Flash will be sending values for these fields using Remoting. Field naming in this exercise is important, as I will use looping to generate lists and text objects in Flash. The file will be saved into the folder %wwwroot%/flashcom/applications/cfdj. The Flash gateway path to this CFC will be flashcom.applications.cfdj.dataControl.

Flash Communication Server SSAS (Main.asc)

I'm not going to get into all the details of building Communication Server applications. You should know that each Communication Server application maintains a unique file called "main.asc". This file contains SSAS and is responsible for handling connection requests and managing server-side objects. The SSAS language is very similar to Flash ActionScript. Just remember that it's case-sensitive.

As an event-based language, the Flash Communication Server has some automatic events that require custom event handlers. You should be aware of at least two event handlers, specifically:

- application.onAppStart(); called only once, when the Communication Server application instance first runs
- application.onConnect(); called each time a Flash player tries to connect to the Communication Server

The onAppStart() handler will contain the initial Flash Remoting call to ColdFusion. It will transfer the returned RecordSet into a persistent SharedObject that will be accessed later by the Flash player. Inside the Main.asc file, you must first load the NetServices class library. This ActionScript class library contains the objects required to connect with ColdFusion and handle the server response.

```
// Load the NetServices Class Libraries
load("netservices.asc");
```

The next sequence of code executes only once because it is placed within the onAppStart() handler. This handler is called only when the Communication Server application starts. Please refer to the source code available at www.sys-con.com/coldfusion/sourcec.cfm for a detailed explanation of the project.

Step 1: Flash Remoting Initialization

Just as I promised, if you're familiar with Flash Remoting, there is nothing different about the following lines of code (again, other than that it's running on the Communication Server, not the client). I have encapsulated this code within a function so I can control when it is called.

```
flashRemoting_init = function () {

NetServices.setDefaultGatewayUrl("http://127.0.0
.1/flashservices/gateway");
  gatewayConnection =
NetServices.createGatewayConnection();
  cf_service =
gatewayConnection.getService("flashcom.applica-
tions.cfdj.dataControl", this);
```

Step 2: Server Response Handler (getAllRecords_Result)

When the ColdFusion function getAllRecords is called (see Step 3), a function within SSAS is created to handle any objects returned from the call. Just like in Flash, the name of this result handler is the combination of the service-function name with a suffix of "_Result". Remember to capitalize the "R", as SSAS is case-sensitive.

The result handler does two things:

- First, using the getColumnNames() function, the names of the database fields are copied as an array into the columnDb_columnNames slot of a remote SharedObject called "serverData_so". (If you're new to SharedObjects and the concept of slots, this will be discussed further.)
- Second, the handler copies each row of the RecordSet into a unique slot in another SharedObject, contact_recordSet_so. A for-in loop is used to create the slots using the index value of the loop as the name of the new slot. The index value ("item") counts from zero to the number of records. It will also act as the Flash

identifier for the rows in the SharedObject. Figure 1 clearly shows how the original rows have been copied into the data property of the SharedObject.

```
this.getAllRecords_Result =
function(result_rs) {

serverData_so.setProperty("contactDb_columnNam
es", result_rs.getColumnNames());

    for (item in result_rs.items) {
   isRecord = result_rs.items[item].__ID__ !=
undefined;
   if (isRecord) {
     contact_recordSet_so.setProperty(item,
result_rs.items[item]);
   }
    }
 }
```

The untouched result_rs RecordSet Object is returned from ColdFusion via Flash Remoting. The __ID__ field is a Flash index key. The RecordID field is the database identifier or unique key. The items (or database rows) are "slots" within the contact_recordSet_so data property. Each row is a unique slot, which is important for synchronizing messages between the server and client.

Compare the structure of the original RecordSet Object with the Remote SharedObject structure. Notice how each data "row" is a "slot" within the data property of the SharedObject.

For this article, consider the Remote SharedObject as a memory area that stores data, accessible by every Flash movie (client) that is connected to it. The Remote SharedObject can be used to access and change data as well as send and receive messages. A message is the invocation of a function on all remote connections. Messages can originate from a Flash player or from the Communication Server.

A very unique feature of the Remote SharedObject is its synchronization messaging. Imagine a scenario that has multiple Flash clients able to change data within a common RecordSet. A method to inform each participant that a change has been made becomes very important. The Communication Server will do this for you automatically. All you have to do

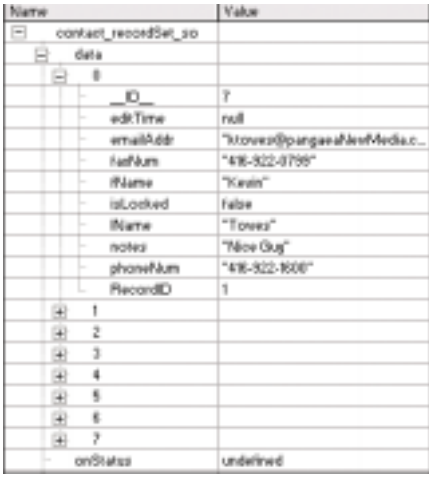


Figure 1: The untouched "result_rs" object (left) returned from ColdFusion compared to the same records mapped as "slots" within the SharedObject, "contact_recordSet_so" data property

is listen for it and code what happens when a change notification is made. This is the reason we have mapped the RecordSet to the SharedObject.

Applying this procedure allows an application to leverage this automatic messaging. Each time any data slot (record) is changed, the server will notify every Flash client which slot (record) has been changed. It does this by sending information objects containing what happened, and which slot (or in our example, which record) was affected.

Two key information object codes are "change" and "success". A value of "success" will be received by the connection that made the change, and a value of "change" will be received by every client connected to the SharedObject that needs to be updated. ActionScript handlers must be constructed to manage these messages. You will see some information object handlers shortly.

Step 3: Retrieve the Data

Request the database collection by calling the service-function, getAllRecords(), on the ColdFusion Server. That will finish off the Flash Remoting initialization function.

```
 callTheServer = cf_service.getAllRecords();
};
```

Step 4: Shared Object Initialization

With the Flash Remoting function ready to go, let's initialize the SharedObjects. Just as the Flash Remoting script was encapsulated into a function, the SharedObject initializations will also be encapsulated so we can control when it happens.

This function will do four main tasks when it's called:

- The first responsibility for this function is to connect the two local variables (serverData_so and contact_recordSet_so) to the Remote SharedObjects. These are the two SOs described earlier: one to track miscellaneous data, and the other to store the address book data. The way SharedObjects work (in Communication Server as well as in Flash) is that when they are first used, if they don't exist, they are created. Think of it as how CFPARAM works.
- The second responsibility is to clear any existing data that may exist in the SharedObjects.
- Third, a custom synchronization handler is applied only to the address book data (the recordSet SharedObject). This onSync script will be explained later.
- Last, after the SharedObjects have been set up, the Flash Remoting initialization script we defined above is called.

```
SO_init = function () {
    serverData_so =
SharedObject.get("serverData", true);
    contact_recordSet_so =
SharedObject.get("contact_recordSet", true);
    serverData_so.clear();
    contact_recordSet_so.clear();
    contact_recordSet_so.onSync = SO_onSync;
    // Initialize Flash Remoting (above)
    flashRemoting_init();
};
```

Step 5: SharedObject Synchronization Monitor (onSync)

This next function is one of the most important functions in the entire application. Its role is to monitor all changes to any data within the SharedObject. (We'll see how the data can be changed later.) When a change is detected, the object indicating the slot changed and a "change" or "success" code (as described at the end of Step 2) will be passed to the function describing what's changed. Assuming it's detecting a change, the changed record is copied into a local temporary variable, where its isLocked property, an actual field within the database table, can be accessed. It is used primarily to disallow users to edit a record if someone else is editing it, as described below.

Its value will also be used to trigger the server. Two values make the server respond: "save" will trigger the server to call the doUpdate() service function (CFC method) on the ColdFusion Server, and "true" will trigger the server to refresh every Flash Client by calling a function, SO_messages. A value of "false" is used as a null value that will force the server to ignore the change (see Listing 2). This function and these values will be reviewed when we look at the Flash ActionScript after Step 12.

Step 6: Communication Server Automatic Event Handlers

These next two steps are simple and use the application-level handlers that we discussed near the beginning: onConnect and onAppStart. Each Flash Communication application runs within an application scope (similar to the ColdFusion Application Scope) and uses its own unique SSAS. Because it is a true object-oriented language, applications actually run under application instances. Each instance of the application maintains its own variables, SharedObjects, and other communication objects. It's not important for this project that you have a complete understanding of application instances; however, it is good to know. Application handlers exist within the application object's instance within the Communication Server.

First, the onConnect method will be automatically called each time a Flash player makes a connection request to the server, such as a client of this shared address book application. It is called a request because, until the server application accepts the request, it will not open a connection. The function "acceptConnection", a built-in method of the application object, opens the door.

```
application.onConnect = function(clientObject)
{
    application.acceptConnection(clientObject);
};
```

The last handler, onAppStart, starts the ball rolling. It is only run once, the first time the application is accessed. Its only job is to call for the initialization of the SharedObject that we created in Step

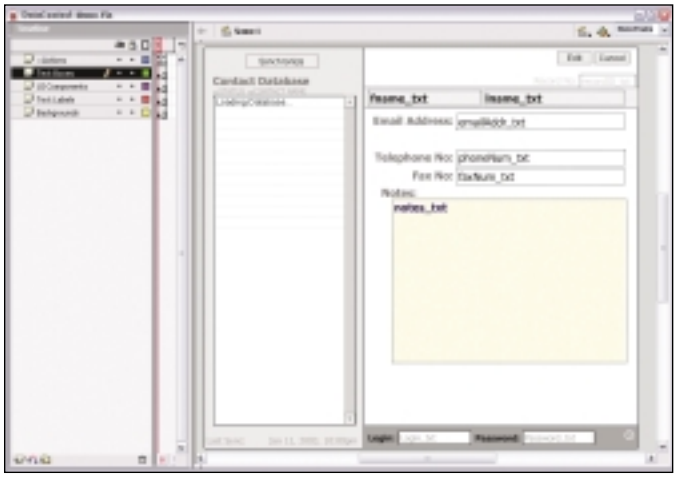


Figure 2: The Flash application interface

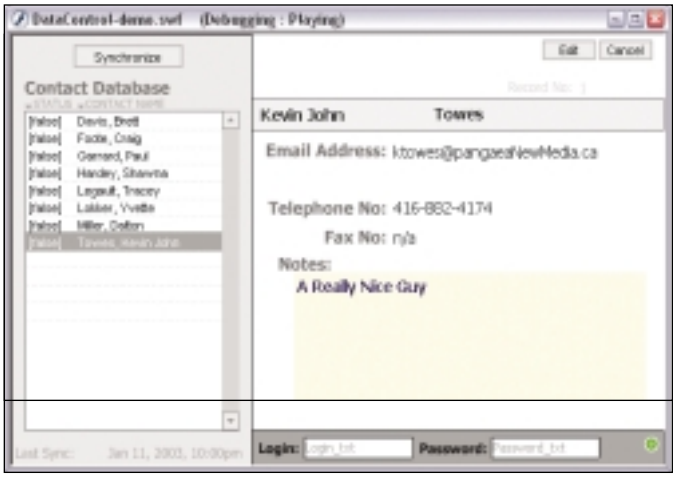


Figure 3: Editing a user record

4. This starts a chain reaction that includes initializing the Flash Remoting.

```
application.onAppStart = function() {
    SO_init();
};
```

A key point to remember is that each time you make a change to SSAS and your Main.asc file, you must reload your application using the Communication App Inspector. The App Inspector is a debugging tool that installs with the Communication Server. It is used to monitor the applications and the instances running on a single server. It is also used as a debugging Watch tool. Just like Flash has the trace() function, so does SSAS. The trace output is displayed in the App Inspector. It's a good idea to put some trace actions in your script so you can watch the startup sequence in the Live Log window of the App Inspector, a useful debugging tool you should learn about.

The Flash application interface (see Figure 2) is quite simple. It consists of a series of dynamic text boxes each named exactly the same as the database field where it will look for values. Each name has a suffix of "_txt" identifying it as a Flash text object. There is a combo box UI component stretched to allow for a long list of contact names. A synchronize button is simply used to keep our ActionScript simple. It's used here to advance the Timeline by one frame and execute additional ActionScript when the user is ready to continue. The edit/cancel buttons allow the user to lock a record, edit it, then save it.

That's about it for an explanation of the interface; refer to the online source code for a deeper view. We'll move on to the client-side ActionScript that makes it all work.

Step 7: Include the Flash Remoting Class Libraries

Setting up a Communication Server application is quite simple in the Flash MX authoring environment. You will need to include the NetServices and DataGlue class libraries, so make sure you have Flash Remoting installed in Flash MX. Although we aren't using Remoting in the Flash player, we will use the RecordSet and DataGlue objects to interact with the data arriving from the Communication Server. These objects are included in the Remoting classes.

```
#include "NetServices.as"
#include "DataGlue.as"
```

Step 8: Initialize the SharedObject: initSO()

Much like the SO_init() function used in SSAS, this function connects two local variables to the two persistent Remote SharedObjects created on the server. The server location of those sharedObjects is held in a my_nc object that will be defined in Step 12. The function also assigns a local function, SO_messages (created in the next step), to handle messages from the server. For the server to access any functions on the Flash player(s), they must be declared within the SharedObject variable.

```
initSO = function () {
  remote_so =
SharedObject.getRemote("serverData",
my_nc.uri, true);
  contact_recordSet_so =
SharedObject.getRemote("contact_recordSet",
my_nc.uri, true);
  remote_so.connect(my_nc);
  contact_recordSet_so.connect(my_nc);
  contact_recordSet_so.SO_messages = SO_mes-
sages;
};
```

Step 9: A SharedObject Handler Called by the Server to Update the Screen – SO_messages()

A SharedObject function can be called from any Flash player or server connected to the Remote SharedObject. This function has two parameters: the command and the source. The variable, doServerUpdate(), contains a true/false value determined if the source of the message is the server. This will help trap erroneous calls to this function.

The value, "updateLocal", was sent from the server's message in the onSync function defined in Step 5. You can easily build on this example code by expanding the switch cases. When called, it will update the localRecordSet with the values of the SharedObject using a function to be defined in the next step.

```
SO_messages = function (command, source) {
  var doServerUpdate = source == "Server";
  if (doServerUpdate) {
    switch (command) {
    case "updateLocal" :
   createLocalRecordSet();
```

```
    break;
     }
  }
};
```

Because of the nature of the Remote SharedObject architecture, you don't need to send the updated RecordSet to each Flash player. Once a change is made to the Remote SharedObject, the change is immediately available to everything that is connected to it.

Step 10: Create a RecordSet Object Pointer to the Remote Data: createLocalRecordSet()

Our next step will perform a local version of the process from Step 2, in which the Remote SharedObject was populated with data from the database. In this step, we'll take that data as passed from the Communication Server, and will create a local RecordSet to hold the data.

When you use a Remote SharedObject as a source to populate a new RecordSet object, something happens that is very powerful and may take you a while to wrap your head around. The new RecordSet is actually a virtual pointer to the Remote SharedObject data. This means that each time the local RecordSet is updated, the Remote SharedObject is updated, and so is every Flash player connected to it. If that isn't enough to excite you, the reason we transfer the SharedObject data back into a RecordSet is so we can use the massive collection of functions. Most importantly, DataGlue allows us to easily populate and format Flash UI components.

```
createLocalRecordSet = function () {
  fieldNames_array =
remote_so.data["contactDb_columnNames"];
  my_rs = new RecordSet(fieldNames_array);
  my_rs.removeAll();
  for (slot in contact_recordSet_so.data) {

my_rs.addItem(contact_recordSet_so.data[slot])
;
  }
  updateList();
};
```

It's important to note that while the Remote SharedObject will change when the local RecordSet changes, it does not work in reverse. The RecordSet is not changed automatically and neither is the screen. When the Flash player is informed of a change to one of the records, it will use this function to update the local RecordSet object, and the screen.

Step 11: Update the Screen – updateList()

This function populates the ComboBox UI component with the names of the common address book. Because the data is available as a RecordSet, it is one single line of code.

```
updateList = function () {
  DataGlue.bindFormatStrings(contactList_cb,
my_rs, "[#isLocked#] #lname#, #fname#",
"#__ID__#");
};
```

Step 12: Connect the Flash Player to the Flash Communication Server

The final major piece to the puzzle is really the first step run when the user begins running the movie. We want to connect the Flash player to the Flash Communication Server, which means requesting that the local client be synchronized with the application managing the Remote SharedObject. Note that when we connect, we're connecting not using HTTP, but instead RTMP, and we're con-

necting not to a CFC, but to our communication application on the Communication Server.

This application is just an example, so I have omitted any security authentication here. We're also using the connectionLight, which is one of the Communication UI components installed to support Communication Server applications in the Flash authoring environment, and that shows a traffic light icon indicating the success of our connection with the Communication Server. The onStatus function is another automatic function that's called when an event occurs on the NetConnection Object. In this example, we're monitoring for a code value of "NetConnection.Connect.success". When this occurs, we know that the connection request was accepted, so we can then start connecting to the SharedObjects and get the party started, calling the initSO function we created in Step 8.

```
my_nc = new NetConnection();
connectionLight_mc.connect(my_nc);

my_nc.onStatus = function(infoObj) {
  if (infoObj.code ==
"NetConnection.Connect.Success") {
  initSO();
}
};

my_nc.connect("rtmp://127.0.0.1/CFDJ/myInstanc
e");
stop();
```

Frame 2, User Interface Controls

When we created the user interface (see Figure 2), we created a synchronize button which, when pressed, passed control to the second frame of the movie. This is where we actually get the initial set of data from the remote server. Normally, you would just put this into another function, but to keep this example as simple as possible, I took this approach.

I am not going to go into great detail for the ActionScript in Listing 3, but there are a few things you should be aware of. When invoked, this script handles the management of the database data. It allows the user to edit a record and manage locking. The first responsibility is to call the createLocalRecordSet() function that we created in Step 10, in the previous Frame. From this point, the movie will run, responding to various actions such as the user selecting data records to view from the contact name list (see Figure 3), possibly editing and/or saving them. The user interface controls will call upon these functions to handle their events. Additional details on these functions can be found in the source code package online.

In several of these functions, you will notice the usage of a loop, for (pos in fieldNames_array){}. This ActionScript loops through the database column names (as obtained in Step 10) and returns the name of the field over each loop. It is used to assign values to dynamic text objects and to manage the data objects. This is why the user interface component instances were named the same as the server-side database columns. A benefit of this dynamic approach is that if the interface and database are changed to add or remove columns, this code doesn't need to change.

Note also that this is the code that sets and uses the isLocked property to decide whether to mark a record as being edited, or prevent the user from editing an already locked record (see Figure 3). Notice also that the label for the save button is changed dynamically based on the status of actions performed, just as the input fields are being dynamically enabled/disabled (see Listing 3). It's all coming together!

A Database to Go, Please!

As promised, here is the code to take your RecordSet on the road! How can you extend this application to allow you even to work with the address book when you're not connected to the server? With Local SharedObjects (as opposed to the Remote SharedObject we've been working with to this point).

Cookies are frequently used to store data on the client persistently, even if the browser is shut down. The Local SharedObject is becoming known as the Flash cookie. Except it is much more effective than an HTML cookie, in that not only is it persistent over browser shutdowns, you can store complex data within it – like a RecordSet Object. Hmm…where might we have a RecordSet we'd want to store persistently on the client?

Here's some homework for you. Use the example in this article and create a local persistent version of the RecordSet that can be accessed without an Internet connection or a connection to a Web server. Here is a leg up:

```
// copy the RecordSet into the Local
SharedObject
local_so = SharedObject.getLocal("contactDb");
local_so.data["contactDb_rs"] = my_rs;

// copy the RecordSet out of the Local
SharedObject
my_rs = new RecordSet();
my_rs = local_so.data["contactDb_rs"];
```

Allowing access to the application (and the data) without a connection to the server is definitely cool, but this wouldn't work without the record-locking and synchronization approaches described throughout this article. This is another frequent problem in Web applications that can be solved with Flash and Communication Server, and it's just one approach to solving such problems. Very powerful stuff!

Summary

Clearly, there are a lot of omissions with this example, including the most obvious features of adding and deleting records. Consider this as a foothold into leveraging the persistent connectivity power of the Flash Communication Server. At a basic level you could use a Flash widget to simply notify users, or force the browser to request an updated HTML page from the ColdFusion Server. An extreme use of these techniques would produce an extremely rich working environment for the user. Watch for my future articles on Flash Remoting and Flash Communication Server here and on Macromedia.com. For more information on the Flash Communication Server, check out my book, *Flash Communication Server MX*, published by Macromedia Press.

About the Author

Kevin Towes is the cofounder of Pangaea NewMedia in Toronto, and author of the book, Flash Communication Server MX *published by Macromedia Press. He is a regular speaker and writes articles for a variety of journals including the Macromedia Designer Developer Center.*

kevin@kevintowes.ca

journeyman cf —*continued from page 24*

Another possible next step would be to explore the Flash Communications Server product. While it's primarily shown as being a tool for streaming audio/video, it's much more than that and really about live shared data applications. This month Kevin Towes offers a great tutorial on getting started with that aspect of the Flash Communication Server in "Persistent Data Communications." Learn more at www.macromedia.com/desdev/mx/flashcom.

One more possible next step to consider is the "Flash Component Kit for ColdFusion." This toolkit, available at www.macromedia.com/v1/handlers/index.cfm?ID=20983, is really something from the Flash 5 time frame. It's a set of ColdFusion custom tags (that work in CF5 as well as CFMX) that provides some prebuilt components that integrate with CF but don't require you to have or use the Flash authoring tool.

Before concluding the article, I'd like to point out a few Web sites that are devoted to topics related to Flash/server integration. Some are portals with articles, links to other sources, mailing lists, blogs, and more. See:

- Flashcfm.com
- Devmx.com
- Flash-remoting.com
- FlashForProgrammers.com
- Flash-db.com
- actionscript.org
- flashmagazine.com
- flashguru.co.uk
- actionscript-toolbox.com
- flashsupport.com
- flashkit.com

Many of these (and an always updated list of them) are available at www.macromedia.com/support/flash/ts/documents/flash_websites.htm.

Conclusion

I hope this little compendium of resources for getting started in using Flash, as a Web application developer, has been useful for you. I'm simply sharing my observations and discoveries in the hope of helping others either get over their own frustration getting started, or avoid it in the first place.

Indeed, I'm thinking of creating my own book and class on the topic, focusing on certain things from this particular perspective and highlighting more lessons learned and key topics.

It's really not difficult to get into all this stuff. It's just a matter of getting that gentle leg up over the wall that then opens to the many paths to creating Flash-enabled Web applications and interface widgets. Now, be on your way!

About the Author

Charlie Arehart is co-technical editor of **ColdFusion Developer's Journal**. *He's also a certified Macromedia trainer/ developer, Team Macromedia member, and CTO of SysteManage. He contributes to several CF resources, is a frequent speaker at user groups throughout the country, and provides training, coaching, and consultation services. He is also now a partner in CommunityMX.com.*
carehart@systemanage.com

Listing 1

```
<CFCOMPONENT>
 <!-- Return a RecordSet Object to Flash -->
  <CFFUNCTION access="remote"
   name="getAllRecords"
 output="false"
 returntype="query">
    <CFQUERY NAME="getData" datasource="articleDb">
      Select * From Contacts order by lName DESC
      </CFQUERY>
      <CFRETURN getData>
     </CFFUNCTION>
    <!-- Update the database record with new values -->
   <CFFUNCTION
 access="remote"
 name="doUpdate"
 output="false"
 returntype="boolean">
     <CFARGUMENT name="fName" type="any" required="false" default="">
     <CFARGUMENT name="lName" type="any" required="false" default="">
     <CFARGUMENT name="emailAddr" type="any" required="false" default="">
     <CFARGUMENT name="phoneNum" type="any" required="false" default="">
     <CFARGUMENT name="faxNum" type="any" required="false" default="">
     <CFARGUMENT name="notes" type="any" required="false" default="">
     <CFARGUMENT name="recordID" type="numeric" required="false" default="0">

        <CFQUERY NAME="getData" datasource="articleDb">
           UPDATE Contacts SET
            fName = '#arguments.fName#', lName='#arguments.lName#',
                   emailAddr='#arguments.emailAddr#',
phoneNum = '#arguments.phoneNum#', faxNum = '#arguments.faxNum#',
notes = '#arguments.notes#', isLocked = false,
editTime = #CreateODBCDateTime(now())#
WHERE RecordID = #arguments.RecordID#
          </CFQUERY>
          <CFRETURN true>
   </CFFUNCTION>

</CFCOMPONENT>
```

Listing 2

```
SO_onSync = function (info) {
  slotToChange = info[0].name;
  if (info[0].code == "change") {
    var updatedRecord_obj = contact_recordSet_so.getProperty(slotToChange);

   // Check the Locking Value for Status messages
   switch (updatedRecord_obj.isLocked) {

   case "save" :
     // change the state of the isLocked value
     updatedRecord_obj.isLocked = false;

     // write the changed record back to the Shared Object
     contact_recordSet_so.setProperty(slotToChange, updatedRecord_obj);

     // the properties within the object contain the changed record
     cf_service.doUpdate(updatedRecord_obj);

     // now update all players
     contact_recordSet_so.send("SO_messages", "updateLocal", "Server");
     break;

   case true :
     contact_recordSet_so.send("SO_messages", "updateLocal", "Server");
     break;
   }
  }
};
```

Listing 3

```
// place the following on Frame 2 of the application
createLocalRecordSet();
stop();

showContact = function () {
  selectedName = contactList_cb.getValue();
    for (pos in fieldNames_array) {
      currentField = fieldNames_array[pos];
     this[currentField+"_txt"].text = my_rs.items[selectedName][currentField];
    }

  isLocked = my_rs.items[selectedName].isLocked;
  if (isLocked) {
    edit_pb.setEnabled(false);
  } else {
    edit_pb.setEnabled(true);
  }
};

enableInput = function () {
  // custom script to convert the dynamic text objects to input text objects,
  // and toggle the edit push button states

  for (pos in fieldNames_array) {
    currentField = fieldNames_array[pos];
    isKey = currentField == "RecordID";
    if (!isKey) {
   this[currentField+"_txt"].border = true;
   this[currentField+"_txt"].type = "input";
    }
  }
  edit_pb.setLabel("Save");
  edit_pb.setClickHandler("saveRecord");
};

disableInput = function () {
  // custom script to return the input text objects to dynamic text
objects,
  // and toggle the edit push button states
  for (pos in fieldNames_array) {
    this[currentField+"_txt"].border = false;
    this[currentField+"_txt"].type = "dynamic";
  }
  edit_pb.setLabel("Edit");
  edit_pb.setClickHandler("editRecord");
};

editRecord = function () {
  // Change the dynamic text objects to input text objects, and lock the record
  enableInput();
  my_rs.setField(selectedName, "isLocked", true);
};

saveRecord = function () {
  // Change the input objects to dynamic objects, and change the record state
  // to "save" also copy the values of the text objects to the RecordSet.
  disableInput();
  my_rs.setField(selectedName, "isLocked", "save");
  for (pos in fieldNames_array) {
    currentField = fieldNames_array[pos];
    my_rs.setField(selectedName, currentField,
this[currentField+"_txt"].text);
  }

};
```

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

THE LARGEST WEB SERVICES, JAVA, XML AND .NET CONFERENCE AND EXPO IN THE WORLD!

Boston 2003 | London 2003 | Berlin 2003 | Hong Kong 2003

JDJ EDGE conference&expo
web services EDGE conference&expo
.NET EDGE conference&expo
XML EDGE conference&expo

Register by February 14th SAVE UP TO $400
Register by March 14th SAVE UP TO $200

3rd Annual EAST

Web Services Edge Conference & Exposition

FINAL PROGRAM

Connecting the Enterprise with Web Services, Java, XML, & .NET

March 18–20, 2003
Hynes Convention Center
Boston

CALL TODAY TO REGISTER
201-802-3058
OR REGISTER ONLINE AT
www.sys-con.com

Event Sponsors

Microsoft · Sun microsystems · SYBASE · IONA · DIGITAL EVOLUTION · COMPUWARE · ORACLE · hp invent · Rational the e-development company · ALTOVA

Media Sponsors

JAVA DEVELOPER'S JOURNAL · JavaWorld · WebServices JOURNAL · LINUX BUSINESS WEEK · XML JOURNAL · SAMS · BASIS · Perl.com DEVMAG · .NET DEVELOPER'S JOURNAL · SD Times · asp.netPRO · WebSphere DEVELOPER'S JOURNAL · WebLogic · wireless · CF Advisor · ColdFusion Developer's Journal · PowerBuilder Developer's Journal

OWNED BY SYS-CON MEDIA

PRODUCED BY SYS-CON EVENTS

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

web services EDGE conference&expo

JDJ EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

THE LARGEST WEB SERVICES CONFERENCE & EXPO

FEATURES & ATTRACTIONS

- **3 Days Packed with Education and Training**
- **4 Keynotes & 3 Panel Discussions**
- **60 Hard-Hitting and Current Seminars**
- **FREE .NET Tutorial with Russ' Tool Shed**
- **Web Services & XML Tutorials**
- **Java University Certification Training**
- **HOT** Industry-Leading Certification Programs
- **1 Day Mobile .NET Tutorial**
- **INFORMATIVE** Round Table Discussions
- Opening Day Welcome Reception
- SAMS Meet the Authors Hot Topics Lounge
- **COMPELLING** Case Studies & Best Practices
- **FEATURED** Product Demonstrations on the Show Floor
- **RIVETING** Real-time SYS-CON Radio Interviews

WHO SHOULD ATTEND

- **Software Developer**
- **Software Engineer**
- **Development Manager**
- **Application Developer**
- **Technical Director**
- **Analyst/Programmer**
- **IT Manager**
- **Technical Architect**
- **Team Leader**
- **Software Consultant**

Keynotes & Highlighted Speakers

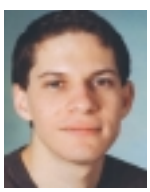
KEYNOTE SPEAKERS

John Magee
Vice President, Oracle9i

John Magee is vice president, Oracle9*i*, at Oracle. He has more than 14 years' experience in the enterprise software industry and has held positions in product development, product management, and product marketing. In his current role, Magee manages technical product marketing for Oracle's application server and development tools products, and is responsible for evangelizing Oracle technology initiatives around J2EE, XML, and Web services.

Mark Herring
Director Java, Web Services & Tools Business

Mark Herring is responsible for helping to define, set, and drive Sun Microsystems' product direction in the Java, Web Services & Tools Business. Prior to his current role, Herring was director of corporate strategy & planning, looking after Sun's interest in the Project Liberty Alliance and Network Identity. Herring joined Sun Microsystems in October 1999 as a result of Sun's acquisition of Forte Software. Forte Software was a leading provider of enterprise-class development and integration products. During his four years at the company, he ran several aspects of Forte's marketing organization, including product marketing and the Web channel.



Miguel de Icaza
Cofounder and CTO

As the founder and leader of the GNOME Foundation, Miguel de Icaza is one of the foremost luminaries in the Linux development community. With his seemingly boundless energy, de Icaza has galvanized the effort to make Linux accessible and available to the average computer user. He brings this same excitement to his role as CTO of Ximian. de Icaza was instrumental in porting Linux to the SPARC architecture and led development of the Midnight Commander file manager and the Gnumeric spreadsheet. He is also a primary author of the design of the Bonobo component model, which leads the way in the development of large-scale applications in GNOME.

Mark Hapner
Distinguished Engineer, Sun Microsystems

Mark Hapner is a Sun Distinguished Engineer and is currently lead architect for Java™ 2 Platform, Enterprise Edition (J2EE™). He has guided the overall architecture for J2EE 1.2, 1.3, and now the upcoming 1.4 release. In March of 1996, he joined Sun's JavaSoftware Division to participate in the development of the Java database connectivity API (JDBC). Following that, he was co-spec lead of the Enterprise JavaBeans specification and spec lead of the Java Message Service specification.

Simon Phipps
Chief Technology Evangelist, Sun Microsystems

Simon Phipps, currently chief technology evangelist at Sun Microsystems, speaks frequently at industry conferences on the subject of technology trends and futures. He was previously involved in OSI standards in the 1980s, in the earliest collaborative conferencing software in the early 1990s, and in introducing Java and XML to IBM.

Dave Chappell
VP, Chief Technology Evangelist, Sonic Software

Dave Chappell is the vice president and chief technology evangelist for Sonic Software. He has more than 18 years of industry experience building software tools and infrastructure for application developers, spanning all aspects of R&D, sales, marketing, and support services. Chappell has published in numerous technical journals, and is currently writing a series of contributed articles for *Java Developer's Journal*.

Eric Newcomer
Chief Technology Officer, IONA

In the role of chief technology officer at IONA, Eric Newcomer is responsible for IONA's technology roadmap and the direction of IONA's Orbix E2A e-Business Platforms as relates to standards adoption, architecture, and product design. Newcomer joined IONA in November 1999, and most recently served as IONA's vice president of engineering, Web Services Integration Products. He is a member of the XML Protocols and Web Services Architecture working groups at the W3C and IONA's Advisory Committee representative to UDDI.org.

Dean Guida
CEO and President, Infragistics

Dean Guida is CEO and president of Infragistics and was CEO and a cofounder of ProtoView Development Corporation. Mr. Guida has over 15 years of experience in the technical industry and oversees all aspects of the company's business operations and corporate direction. He is also responsible for cultivating strategic alliances and other external relationships, as well as managing corporate financial affairs.

Conference Advisory Board

Sean Rhody
Editor in Chief,
Web Services Journal
Partner, CSC

Alan Williamson
Editor in Chief,
Java Developer's Journal
Chief Technology Officer, n-ary

Derek Ferguson
Editor in Chief,
.NET Developer's Journal
Chief Technology Evangelist, Expand Beyond Corp.

Bob Familiar
.NET Architect,
Microsoft
New England

Thomas Robbins
Senior Technology Specialist,
Microsoft
New England

Hitesh Seth
Editor in Chief,
XML-Journal
Chief Technology Officer, ikigo

J.P. Morgenthal
Chief Services Architect, Software AG

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Conference at-a-Glance

THE LARGEST AND THE MOST

		JAVA	WEB SERVICES	.NET
TUESDAY MARCH 18	8:00AM -4:00 PM	Registration Open		
	9:00AM -9:50 AM	(JV1) Squeezing the Best Out of Java Alan Williamson, Java Developer s Journal	(WS1) Web Services Infrastructure Carl Sjogreen, BEA	(NT1) .NET Framework Overview Bob Familiar, Microsoft
	10:00AM -10:50 AM	Web Services Keynote: John Magee, Oracle		
	11:00AM -11:50 AM	(JV2) Testing Your Java Using JUnit Kyle Gabhart, LearningPatterns	(WS2) Web Services Management James Phillips, Actional	(NT2) Introduction to ASP.NET Russ Fustino, Microsoft
	1:00PM -1:50 PM	WS-I Panel: A Road Map for Web Services Standards - Moderated by Rob Cheng, WS-I		
	2:00PM -2:50 PM	.NET Keynote: The MONO Project - Miguel de Icaza, Ximian		
	3:00PM -3:50 PM	(JV3) Building/Deploying the Ant Way Kyle Gabhart, LearningPatterns	(WS3) Strategies for Using Databases in a World of Web Services Mike Lehmann, Oracle	(NT3) Introduction to VB.NET Russ Fustino, Microsoft
	4:00PM -4:50 PM	(JV4) Unlocking the Secrets of JDK1.4 Raghavan Srinivas, Sun Microsystems	(WS4) Using Web Services to Integrate J2EE and .NET Enterprise Applications - Odysseas Pentakalos, SYSNET International	(NT4) How to Develop an End-to End .NET Connected Appilication Allan da Costa Pinto, Microsoft
WEDNESDAY MARCH 19	8:00AM -4:00 PM	Registration Open		
	9:00AM -9:50 AM	(JV5) Java APIs for Web Services Security Standards Sang Shin, Sun Microsystems	(WS5) Combining BPM and BRM Technologies: A Major Step Towards Corporate Agility Henry Bowers, ILOG	(NT5) .NET: The Virtualized Execution Engine Yahya Mirz, Aurora Borealis
	10:00AM -10:50 AM	Java Keynote: Mark Herring, Sun Microsystems		
	11:00AM -6:00 PM	EXPO OPEN 11:00 a.m. - 6:00 p.m.		
	11:00AM -11:50 AM	(JV6) To Not Swing Is to SWT! The Swing Alternative - IBM	(WS6) Web Services for Real-Time Data Access in an Industrial Setting Stephan Van Dijck, ABB/SKYVA	(NT6) Introduction to DotGNU Barry Fitzgerald, DotGNU
	12:00PM -2:00 PM	BREAK & EXPO		
	2:00PM -2:50 PM	.NET Panel Discussion - Moderated by Derek Ferguson, .NET Developer s Journal		
	3:00PM -3:50 PM	(JV7) Unlocking the Power of XML Hitesh Seth, ikigo	(WS7) Web Services Architecture: The Next Big Spec. from the Mouths of the W3C Eric Newcomer, IONA (moderator)	(NT7) Introduction to SSCLI Yahya Mirz, Aurora Borealis
	4:00PM -4:50 PM	(JV8) Java and .NET Derek Ferguson, Expand Beyond	(WS8) Web Services: Next Steps After the Hype Claire Dessaux, Oracle	(NT8) Mobile Development with the Compact Framework Brad Mccabe, Infragistics
THURSDAY MARCH 20	8:00AM -4:00 PM	Registration Open		
	9:00AM -9:50 AM	(JV9) Writing SOAP Services Nigel Thomas, SpiritSoft	(WS9) Web Services Best Practices Chris Peltz, HP	(NT9) Best Practices for .NET Develop ment Joe Stagner, Microsoft
	10:00AM -10:50 AM	.NET Keynote - Jesse Liberty, Liberty Associates		
	11:00AM -4:00 PM	EXPO OPEN 11:00 a.m. - 4:00 p.m.		
	11:00AM -11:50 AM	(JV10) Working with Data the JDO Way Patrick Linsky, SolarMetric	(WS10) Web Services Startups: Telltales of the Future Simeon Simeonov, Polaris Venture Partners	(NT10) Best Practices for ADO.NET Development Thom Robbins, Microsoft
	12:00PM -2:00 PM	BREAK & EXPO		
	2:00PM -2:50 PM	Java Panel - The Future of Java , Moderated by Alan Williamson, Java Developer s Journal		
	3:00PM -3:50 PM	(JV11) Enterprise: The Next Generation Mark Hapner, Sun Microsystems	(WS11) Open Standards for Web Services Messaging Dave Chappell, Sonic Software	(NT11) How to Debug with .NET Tony Denbow, STAR Information Tech- nology
	4:00PM -4:50 PM	(JV12) Overcoming the Challenges of J2ME Dr. Jeff Capone, Aligo	(WS12) Web Services Security Marc Chanliau, Netegrity	(NT12) XML and Web-Enabling Legacy Applications Using BizTalk Mike Cramer, Microsoft

XML	VENDOR
(XM1) XML - A Manager s Guide JP Morgenthal, Software AG	Visit www.sys-con.com for details
(XM2) OASIS Standards Update Karl Best, OASIS	(VN2) The XMLSPY 5 Enterprise Edition Development Environment Trace Galloway, Altova
(XM3) A Definitive Introduction to XML Schemas Aaron Skonnard, DevelopMentor	(VN3) SOAP and Java: Marrying Them Off Skip Marler, Parasoft
(XM4) XML in Print - XSL:FO Frank Neugebauer, IBM	Visit www.sys-con.com for details
(XM5) XML Security Integration Challenges Phil Steitz, American Express	(VN5) Process-Centric Enterprises Eric Pulier, Digital Evolution
(XM6) Case Study: XML in Life Sciences Tim Matthews, Ipedo	(VN6) Pattern Driven Application Development Tom Shore, Compuware
(XM7) Using XML for EAI - Best Practices Dan Enache, TIBCO	(VN7) Managing the Developer Relationship Mike Bellissimo, Sun Microsystems
(XM8) Take XML with You: XML and Mobile Computing - Hitesh Seth, ikigo	(VN8) Web Services Diagnostics Dave Seidel, Mindreef
(XM9) XML, Ontologies and the Semantic Web - Ayesha Malik, Object Machines	Visit www.sys-con.com for details
(XM10) X Query Mike Champion, Software AG	(VN10) Model Driven Development of Web Services in UML for the J2ME Bill Graham, Rational Software
(XM11) XPath & XSLT 2.0 BEA Kurt Cagle, Cagle Communications	(VN11) Why Web Services Management? Jon Atkins, HP
(XM12) Third Generation XML Tools Michael Leventhal, Commerce One	Visit www.sys-con.com for details

JAVA UNIVERSITY℠ PROGRAM

9:00AM -5:00 PM

Web Services Programming Using Java™ Technology and XML

This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.

9:00AM -5:00 PM

Java 2 Platform Programmer Certification Fast Path

This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.

9:00AM -5:00 PM

Java 2 Platform Architect Certification Fast Path

This intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam s testing objectives.

FAST TRACKS & TUTORIALS

9:00AM -5:00 PM

XML Certified Developer Fast Path

This tutorial is for programmers who have some knowledge of XML and related technologies and would like to pass the IBM Certified Developer Test 141 on XML and Related Technologies.

9:00AM -5:00 PM

Russ Tool Shed

Join Russ as he shows you how to use Visual Studio.NET.

9:00-12:15 - Introduction to Web Services Using VS.NET

1:00-2:30 - Advanced Web Services Using ASP.NET

2:45-4:15 - .NET Remoting Essentials

9:00AM -5:00 PM

Mobile .NET

In this session, Derek Ferguson, editor-in-chief of .NET Developer s Journal, will give you a thorough introduction to the use of .NET with all manner of mobile computing devices.

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

SUN MICROSYSTEMS Java University℠ Program

Java University℠ Program: Aggressive, code-level training courses for experienced developers using Java™ technology, brought to you by Sun Microsystems, Inc. Attend seminars designed by industry luminaries and recognized experts. Sessions cover Sun certification and Web services technology. Whether you're a beginner or a veteran developer, architect, or software engineer, you'll benefit from these value-packed full-day courses. Register now. Seating is limited.

The Java University℠ program complements this year's Web Services Edge conference by offering 3 full-day training lectures for experienced software developers, architects, and engineers.

Java University℠ Program Take-Aways:

- Training designed and presented by expert Java technology engineers
- Student guides full of source code, examples, references and copies of instructors' materials
- Free Web-based training courses from Sun Educational Services

Tuesday, March 18, 2003

Web Services Using Java™ Technology and XML

SANG SHIN, SUN MICROSYSTEMS, INC.

Who Should Attend

Web services designers and programmers, application developers, and programmers using the Java programming language who have experience using the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites

Experience using the Java programming language and basic knowledge of XML

Overview

This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.

The session will start with an introduction on fundamental concepts and characteristics of Web services. This will be followed by a detailed explanation of how to implement, describe, register, discover, and invoke Web services using core Web services standards - Simple Object Access Protocol (SOAP); Web Services Description Language (WSDL); and Universal Description, Discovery, and Integration (UDDI). In addition, the ebXML standard, which defines the framework for the global electronic marketplace will be talked about in detail. Also, the tools for building and deploying Web services will be discussed. Each topic will be presented with concrete examples and demonstrations when possible.

Attendees will also learn how to use standard Java APIs for Web services, mainly Java API for XML Messaging (JAXM), Java technology API for XML-based RPC (JAX-RPC), and Java technology API for XML Registries (JAXR) for developing and deploying Web services.

Benefits

- Learn the fundamental concepts and characteristics of Web services. Gain detailed understanding on core Web services standards: SOAP, WSDL, UDDI.
- Gain a detailed understanding of ebXML, the standard framework for electronic business.
- Learn Java programming language APIs for Web services – JAXM, JAX-RPC, JAXR

Wednesday, March 19, 2003

Java™ 2 Platform: Programmer Certification Fast Path

PHILIP HELLER, PRESIDENT, HELLER ASSOCIATES

Who Should Attend

This session is designed for programmers who have some exposure to the Java™ programming language, and are ready to prepare for the Sun Certified Programmer for Java 2 Platform exam.

Prerequisites

Object-oriented software development experience and familiarity with the syntax and structure of Java technology-based development.

Overview

The development community recognizes that competency developing solutions using Java technology is vital to productivity, reaffirms your value to your organization, and increases your career advancement opportunities. This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.

Benefits

- Receive an intensive review of the advanced topics covered on the Sun Certified Programmer for the Java 2 Platform Exam
- Increase your understanding and knowledge of Java programming language syntax and structure
- Prepare for the exam by reviewing practice tests and questions
- Gain a strong understanding of Java fundamentals

Thursday, March 20, 2003

Java™ 2 Platform: Architect Certification Fast Path

SIMON ROBERTS, TECHNOLOGY EXPERT AND COURSE DEVELOPER, SUN MICROSYSTEMS, INC.

Who Should Attend

This session is designed for enterprise application architects, system analysts, experienced technologists, and developers using Java™ technology seeking certification as an architect for the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites

Understand the benefits of Java technology solutions; experience with object-oriented analysis and design; familiarity with concepts of distributed computing.

Overview

Many of the solutions in today's "Net economy" are, or soon will be, developed using the Java 2 Platform, Enterprise Edition (J2EE) architecture. Gaining recognized competency architecting J2EE platform-based solutions is vital to your success as an architect, reaffirms your value, and increases your career opportunities.

Developed and presented by Mark Cade, this intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam's testing objectives. Multiple real-world case studies are used to demonstrate correctly architected J2EE technology–based solutions and pinpoint key topics presented within the architect exam.

Additionally, you will learn how to interpret exam objectives, what each of the three exam phases contains, and clear guidelines and resources to use after the course.

Benefits

- Receive an intensive review of the topics covered on the Sun Certified Enterprise Architect for the Java 2 Platform, Enterprise Edition Exam
- Increase your understanding and knowledge of successfully architecting solutions using J2EE technology
- Understand the system qualities: scalability, availability, extensibility, performance, and security
- Understand trade-offs of different architectural choices as they pertain to system qualities.
- Describe the benefits and weaknesses of potential J2EE technology–based architectures
- State benefits and costs of persistence management strategies
- Review real-world case studies of J2EE technology–based architecture
- Prepare for the exam by reviewing practice tests and questions

Major Sponsors at the World's Largest Web Services, Java, XML, and .NET Conference and Expo

2003 East Gold Sponsors

COMPUWARE

Compuware will be the best worldwide provider of quality software products and services designed to increase productivity. We will continue to create practical solutions that meet our customers needs and surpass their expectations. We will provide an environment for our employees where excellence is encouraged and rewarded and where diversity is promoted at all levels of the company. Compuware is built upon a diverse employee population with a set of key attributes that is common to every individual: Secure, Moral, Ethical, In Agreement with Economic Purpose, Team Player, Other Oriented, Positive, Optimistic, Energetic Achiever and Bright.

Microsoft

Introducing Visual Studio .NET; visionary yet practical, the single comprehensive development tool for creating the next generation of applications has arrived. Developers can use Visual Studio .NET to:

- Build the next-generation Internet.
- Create powerful applications fast and effectively.
- Span any platform or device.

Visual Studio .NET is the only development environment built from the ground up for XML Web services. By allowing applications to share data over the Internet, XML Web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model.

ORACLE

Oracle Corporation (Nasdaq: ORCL) is the world's largest enterprise software company, providing enterprise software to the world's largest and most successful businesses. With annual revenues of more than $9.4 billion, the company offers its database, tools, and application products, along with related consulting, education, and support services. Headquartered in Redwood Shores, California, Oracle is the first software company to develop and deploy 100 percent Internet-enabled enterprise software across its entire product line: database, server, enterprise business applications, and application development, and decision support tools.

2003 East Silver Sponsor

hp invent

HP is a leading provider of products, technologies, solutions and services to consumers and business. The company's offerings span IT infrastructure, personal computing and access devices, global services, and imaging and printing. Our $4 billion (U.S.) annual R&D investment fuels the invention of products, solutions and new technologies, so that we can better serve customers and enter new markets. We invent, engineer, and deliver technology solutions that drive business value, create social value, and improve the lives of our customers.

2003 East Featured Sponsor

Sun microsystems

Sun was founded with one driving vision. A vision of computers that talk to each other no matter who built them. A vision in which technology works for you, not the other way around. While others protected proprietary, stand-alone architectures, we focused on taking companies into the network age, providing systems and software with the scalability and reliability needed to drive the electronic marketplace.

2003 East Corporate Sponsors

IONA

IONA is the leading e-Business Platform provider for Web Services Integration. IONA Orbix E2A enables the flow of information across disparate systems and the liberation of application assets for new business opportunities. For additional information about IONA, visit our Web site at http://www.iona.com.

Rational the e-development company™

Rational Software provides a software development platform that improves the speed, quality, and predictability of software projects. This integrated, full life-cycle solution combines software engineering best practices, market-leading tools, and professional services. Ninety-six of the Fortune 100 rely on Rational tools and services to build better software, faster. Additional information is available on the Internet at http://www.rational.com

SAMS

Sams Publishing has more than 500 titles in print, and is one of the most successful computer book publishers in the world. From introductory tutorials to comprehensive reference books, Sams Publishing focuses on teaching tomorrow's programmers, developers and system administrators the skills they need to build and manage emerging technologies.

SYBASE

Sybase has always delivered solutions that help customers to share data, is platform independent, and integrates everything: platforms, application servers, components, databases, portals, processes, message brokers, and mobile/wireless. Our technologies have the insight that will promote ease of use, leverage best practices, ensure positive ROI, and help your organization build a successful, pragmatic strategy based on next-generation technologies. Check us out at http://www.sybase.com/solutions/e-business

DIGITAL EVOLUTION
The Business Value of Web Services

DE solutions and thought leadership deliver the value of Web services to business users today. DE unleashes the value of underlying IT assets, giving enterprises the freedom to innovate and evolve business processes within a standards-based, vendor neutral, loosely coupled framework. Some of the largest corporations in the world rely on DE's Web services solutions. www.digev.com

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

REGISTER ONLINE TODAY @ SYS-CON.COM

JDJ EDGE conference&expo | web services EDGE conference&expo | .NET EDGE conference&expo | XML EDGE conference&expo

Don't Miss the Developer Event of 2003

Meet with the industry experts, professionals, and visionaries connecting the enterprise with today s i-technology!

Conference
March 18–20, 2003

Exposition
March 19–20, 2003

Hynes Convention Center, Boston

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

Actional · AltoWeb · Antenna House · Covasoft · CTI · DPRIG · Excelsior · FatWire · webGain · altio · attachmate · CACHEON · Cassatt · Apress · CORBA · atg · webputty · TogetherSoft · SilverStream · KENAMEA · sitraka · Rational · TIBCO · systinet · SEAGULL · ca · ADDS · aligo · bea · Bestning · Cape Clear · COMPUWARE · DataMirror · HITSOFTWARE · InCert · INFERDATA · InfoShark · IONA · LEARNINGPATTERNS.COM · MERANT · NOVALOGS · Pyxis · SAVAJE · sonic · TOWER · ORACLE · XMLCities · Borland · Integra · Matisse · IPEDO · VORDEL · Altaworks · boostnet · CSC · iam · NEXTEL · PRAMATI · INFRAVIO · lazy · MERCURY INTERACTIVE · KeyLabs · macromedia · METASERVER · MICRO FOCUS · VULTUS · webMethods · oakgrove · PointBase · RSA · SYBASE · Trivera · WAKESOFT · ObjectBuilders · Optimyz · plumtree · Trellix

CFMX Exam Buster from CentraSoft

When you want to beat the competition

"Do you shave your entire body?" It's kind of an odd question (and might get you a funny look or a slap in the face depending on who you ask!), but if you were to ask a world-class swimmer, the answer would almost certainly be "yes." If you're serious about competition, you need to be willing to take the necessary steps to give you an edge. In these days of stiff job competition, getting certified in technical skill sets may make the difference between landing your dream job or continuing to dream.

Reviewed By
Phil Cruz

Overview

CFMX Exam Buster from CentraSoft is a testing tool geared to help developers earn their ColdFusion MX Developer Certification from Macromedia. It consists of over 750 questions and 11 practice tests that are similar to those you would take as part of the real exam. The questions are designed to test your knowledge in the following areas of Web/ColdFusion development: Application Development, Database Concepts, Client State Management, and Data Exchange.

Installation

Installing the software was painless. The software is provided as an executable file. After launching the installer, simply answer some password/registration dialog boxes, specify the install directory, and that's it.

Taking a Test

Upon starting the program, you have three ways to proceed: select from a list of predefined tests, take a topic test, or take a random test. CFMX Exam Buster comes with 11 tests that are similar in format to the real certification exam. The test questions can be of three formats: true/false, multiple choice, and multiple response (select more than one correct answer). See Figure 1 for an example. After you answer a question, you can move on to the next question, go back, or go to a specific question.

If you are unsure of the answer, you can mark a question for review and come back to it at the end of the exam. If you're really stuck, you can opt for the "life preserver" that opens the cheat mode window with the correct answer. Cheat mode can be enabled/disabled in the program options.

As with the real test, you have 70 minutes to complete 66 questions. After you complete all the test questions, you can have the program score your test. A score of 65% or better is required to pass. A score of 80% or above means you have earned "advanced" status. Along with the overall score, you will be presented with the summary of all your questions and answers as well as a list of only the questions you answered incorrectly.

The program also categorizes each question into one of 22 topic areas, including ColdFusion components, arrays, and SQL and CFQuery concepts. You can see which topics need improvement and then take topic tests to concentrate in those areas.

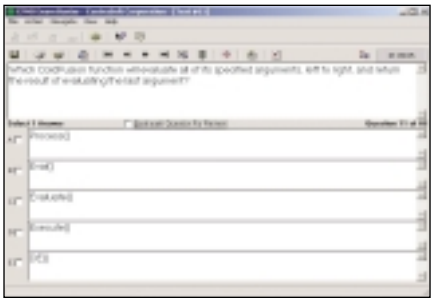


Figure 1: CFMX Exam Buster is similar in format to the real certification exam

In addition, the program has a "submit comments" feature. If, while reviewing your test, you think that your answer was correct, but marked incorrect, or if you think a question may not have been stated clearly, this feature will bring up a Web page with a form to submit comments to CentraSoft. I used this feature and received a quick reply.

Conclusion

The key to doing something well is lots of practice and a good practice environment. For the CFMX Certification Exam, CFMX Buster provides just that. After using the tool, I could go into the real exam with confidence and a feeling of "been there, done that." I don't know if you need to shave your body, but if you plan on getting certified I would recommend using CFMX Buster.

About the Reviewer

Phil Cruz is a certified ColdFusion developer. He has over 10 years of experience in the computing industry and specializes in the design and development of Web-based applications.

philcruz@yahoo.com

Vitals: CFMX Exam Buster

CentraSoft Corporation
http://centrasoft.com • info@centrasoft.com

Systems Alliance Launches Latest Site Executive, Version 3.0

(*Sparks, MD*) – Systems Alliance, Inc., a provider of Web-based solutions, has announced the release of Site Executive 3.0, the latest version of its Web Content Management System (CMS). Version 3.0 has been developed to provide a wider array of features, allowing companies to create and update their Internet, Intranet, and Extranet content in just minutes, with an interface simple enough for anyone to use.

Systems Alliance has designed Site Executive using Macromedia ColdFusion MX. The latest version includes several new features and additional provisions offering enhanced meta tagging strategies, which make Web sites more accessible to search engines. In addition, Version 3.0 records all updates that an enterprise makes to its site's content, and offers a reporting module that provides a comprehensive summary of activity in a format chosen by the enterprise. Site Executive's extensive template options will be subject to permissions, enabling content authors to only view templates appropriate to their areas and ultimately produce easier site-wide updates.

Additionally, Site Executive is cost-effective, paying for itself in 12–18 months, and permits most companies to repurpose a significant portion of the cost of their Webmasters.

"Site Executive is a perfect example of the powerful solutions customers are building with Macromedia ColdFusion MX," said Phil Costa, senior product manager, Macromedia. "With its rapid server scripting environment, Macromedia ColdFusion MX provides an ideal platform on which to build robust solutions like Site Executive."

Macromedia Delivers FreeHand MX

(*San Francisco*) – Macromedia, Inc., has announced the availability of Macromedia FreeHand MX, the multipublishing solution for conceptualizing and designing visually rich graphics for print, the Internet, and Macromedia Flash projects. FreeHand MX includes the streamlined Macromedia MX interface, new productivity tools and panels, improved Mac OS X performance, and increased integration with Macromedia Flash MX. Macromedia FreeHand MX is available for download from the Macromedia Online Store at www.macromedia.com/go/buyfhmx.

"Macromedia FreeHand MX enables graphic designers to streamline their workflow and easily move from concept to design to production in a single-file, multipage environment, empowering them to meet today's needs of doing more with less," said Norm Meyrowitz, president of products, Macromedia. "Macromedia Freehand MX contains a powerful set of illustration and design tools to help users realize their creative vision, whether the end result is print, the Web, or a Macromedia Flash MX project."

Macromedia FreeHand MX continues the product's rich heritage of solving design problems for graphic designers and illustrators by providing a streamlined workflow that enables users to create their artwork once and then easily repurpose it to work across multiple media. FreeHand MX provides everything necessary for storyboarding, creative design, multipage document production, and editing.

"Working on deadline for a newspaper, we need to produce complex graphics quickly. The powerful new features of FreeHand MX make that task easier," said R. Scott Horner, assistant graphics director, *South Florida Sun-Sentinel*. "The live raster effects such as shadows on text and the support of transparent PNG images means our print graphics look even better – with little to no increase in our production time. This is a major upgrade."

Macromedia Buys Presedia

(*San Francisco*) – Macromedia, Inc., has recently purchased Presedia Inc., and its main e-learning product, Express. Presedia offers online presentation and e-learning tools for sales and marketing, and communications purposes.

Macromedia Studio MX Plus Delivers Integrated Developer Tools

(*San Francisco*) – Macromedia, Inc., has announced Macromedia Studio MX Plus, a special edition of the award-winning suite of integrated tools that now includes Macromedia FreeHand MX, Macromedia Contribute, and Macromedia DevNet Resource Kit Special Edition at no extra cost. Macromedia Studio MX Plus helps customers leverage their existing investments. The $899 Macromedia Studio MX Plus, which replaces Macromedia Studio MX for Windows, is available for immediate download from the Macromedia Online Store at www.macromedia.com/go/buystudiomxplus.

"Our customers are being asked to do more with fewer resources – to wear multiple hats, to deliver complex projects within tighter timeframes, and to better leverage their existing investments," said Macromedia's Meyrowitz. "This special edition of Macromedia Studio MX provides added value to help developers streamline their workflow so they can deliver great experiences more efficiently."

Since its introduction last June, more than 350,000 customers have purchased Macromedia Studio MX, making it the fastest adopted product in the company's history. Macromedia Studio MX includes Macromedia Dreamweaver, Macromedia Flash, Macromedia Fireworks, Macromedia FreeHand, and Macromedia ColdFusion MX Developer Edition.

Macromedia Studio MX Plus is a special commercial edition for Windows customers, that also includes Macromedia Contribute, and Macromedia DevNet Resource Kit Special Edition. For more information on Macromedia Studio MX Plus, visit www.macromedia.com/go/studiomx plus.

Nominate Your ColdFusion Product/Service for the 2003 CFDJ Readers' Choice Awards!

Nominate your favorites for the 2003 ***ColdFusion Developer's Journal Readers' Choice Awards***, and help highlight excellence in the solutions and services provided by the top ColdFusion firms in the industry. Widely referred to as "The Oscars of the Software Industry," the Readers' Choice Awards program has become the most respected industry competition of its kind. Voting begins March 1, and continues until August 30, 2003. Then come to applaud the winners at Web Services Edge 2003 West, in Santa Clara, CA, October 19–22. http://www.sys-con.com/coldfusion/readerschoice2003