web services EDGE world tour 2002

NEW YORK, NY......................**MARCH 19**
SAN FRANCISCO, CA ....**APRIL 22**

Page 36 ◄ Register for Web Services Edge 2002 East Gold Passport, Attend the World Tour FREE!

COLDFUSION Developer's Journal
CFDJ
Readers' Choice AWARD

2001 READERS' CHOICE AWARDS
Page 40

SYS-CON MEDIA

CFNORTH DEVELOPER'S CONFERENCE
www.cfnorth.com

COLDFUSION Developer's Journal

It's That Time of Year Again…

BY **ROBERT DIAMOND**

No, not Christmas, but our annual *ColdFusion Developer's Journal* Reader's Choice Awards announcements. Categories for this year's awards are Best Book, Best Consulting Service, Best Custom Tag, Best Database Tool, Best Design Service, Best E-Business Software, Best Education and Training, Best Testing Tool, Best Web Development Tool, Best Web Hosting, Best Web Performance Tool, Best Content Management Tool, Best Web Site, Best Web Application, and last, but not least, Most Innovative CF Application. With awards in these 15 categories, *CFDJ,* with the help of thousands of our readers, has once again recognized the best and the brightest in the world of ColdFusion. The results were varied for this year's awards – from the extremely predictable, like Ben Forta winning the top two prizes in the best book category, to some unpredictable results as well. Check out our complete coverage on page 38.

I think the Reader's Choice Awards are important to the ColdFusion industry as a whole, not just for their ability to showcase the best of what's out there, but also because, like *CFDJ,* they indicate the state of the industry. The success of a development language like ColdFusion can be measured in several ways. Of course, there are the typical statistics that float around to investors as well as developers about overall usage and market penetration, and the prestige of how many Fortune 50, 100, and 500 firms are deploying it on their enterprise applications. In addition to all these, there's something else that's equally important – how many commercially available solutions there are.

Add-on products for the developer community often become a branched-off industry of the product. In last month's issue (Vol. 4, issue 2) we focused on user-defined functions and custom tags to make your development life easier. We covered the reuse of code modules, going over the many ways they can fit into your projects.

Those custom tags that have become a commercial success and those products built in ColdFusion, or designed to ease CF development, are as important to growing a successful industry as adding new features to the product itself. They help propel the industry forward and are crucial examples of ColdFusion success stories that will convince others who are not developing with CFML to do so. We'll be looking at some of those throughout the next several issues, and if you have a success story, I encourage you to write to us here at *CFDJ.*

On the topic of successful ColdFusion projects, we recently relaunched our complete line of developer forums at SYS-CON.com using FuseTalk from e-ZoneMedia. Macromedia is now using FuseTalk as well, for those familiar with their excellent developer forums. Stay tuned for information on the 2002 Reader's Choice Awards with nominations starting soon. Also, if you're developing in either Java or Web services, look at our *Java Developer's Journal* and *Web Services' Journal* Reader's Choice Awards, with voting taking place now on www.sys-con.com.

ABOUT THE AUTHOR
Robert Diamond is editor-in-chief of **ColdFusion Developer's Journal** *as well as* **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 recently graduated from the School of Information Studies at Syracuse University with a BS in information management and technology.*

@ ROBERT@SYS-CON.COM

A Case for Me
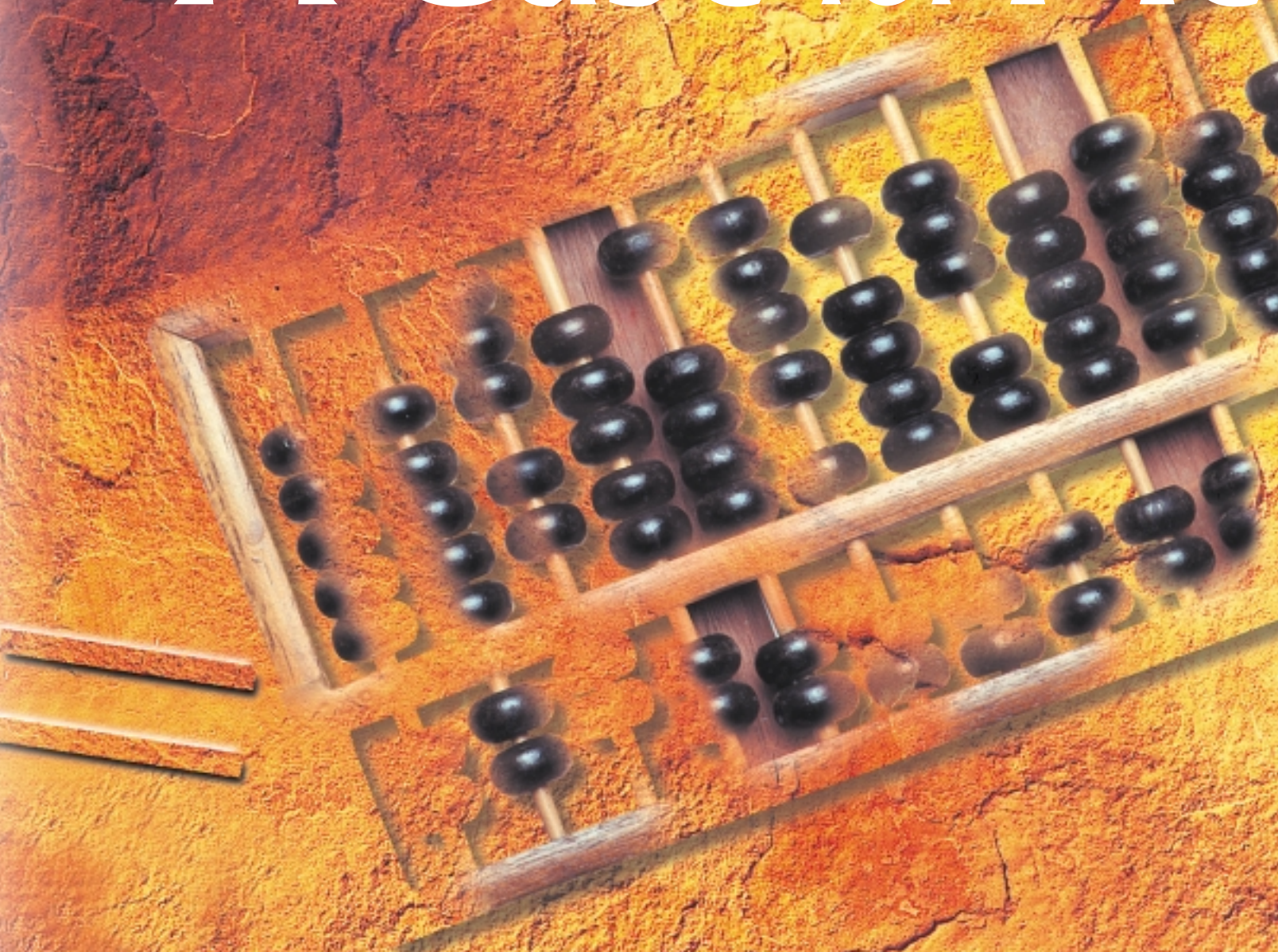
Who needs a methodology?

thodologies

Several years ago I came across a statistic that I've since shared with many others and it never ceases to shock me: according to four separate studies, the failure rate for custom corporate software development hovers at 70%. In fact, some studies indicate the figure may be even higher.

What is software failure and what causes it? There's no simple answer, but one thing stands out: software failures are seldom technical. Instead, the causes are much more fundamental – the delivered software does not do what the client needs it to do. Without dealing with this root cause, much of the excellent work being done to improve the technical aspects of our jobs is analogous to strengthening the strongest link of a chain, but providing no overall improvement to the chain itself.

It's my experience that in the vast majority of cases, the success or failure of a software project is determined before the first line of code is written. Software projects certainly fail, quite often, but they seldom do so because the developers were unable to handle multidimensional arrays or matrix math, or deal with recursion.

In fact, the software may run just as the developers intended, and yet the client still deems it a failure. Pointy-haired bosses are often given to oversimplification ("you people did not give it 100%"), but an overwhelming majority of the developers I know are both competent and committed. Something else is at work here. What did Shakespeare write? "The fault, dear Brutus, is not in our stars, but in ourselves…" Or, perhaps in our methodologies.

What Is a Methodology?

If you tell me, "Take these blocks and stack them on the table over there," you've given me an order. Being a conscientious person, I'll try to stack them in a pleasing arrangement – pleasing to me, that is. If you happen to like what I like, you'll love the results. If not…What if you give me the following instructions instead?

1. Place the blocks from the canvas bag on the table.
2. Align the red blocks so they form a single, straight line.
3. Place the blue blocks atop the red blocks, but offset from the blocks underneath, so that the end of one blue block falls exactly in the middle of the red block beneath it.
4. Place the green blocks atop the blue blocks, but offset from the blocks underneath, so the end of one green block falls exactly in the middle of the blue block beneath it.
5. Place the remaining black block atop the green blocks, exactly centered.

Quill Design
www.quilldesign.com

Now you've given me a procedure for stacking blocks. If I follow it, I'll produce a pyramid. More important, anyone following the procedure will produce a pyramid. With an order, you'll get an infinite number of orderings. With a methodology, you'll consistently produce a pyramid.

I speak, write, and teach a good deal on methodologies in general and Fusebox in particular. There are often a few (brave) souls who will honestly admit that they don't want to know exactly what they're building beforehand – part of the excitement of their job is discovering new shapes and configurations for the blocks.

I recognize and greatly empathize with this attitude. For years, I was a woodworker, designing and building custom furniture pieces. I found the idea of buying a plan and then carefully following the instructions abhorrent. I saw myself as a craftsman/artist, not a machine turning out replicas.

There is certainly a call for independent craftspeople/artists in every discipline and industry, but unless programming is a hobby, you're a professional – you're getting paid – and unless the people paying you understand that they're paying for an independent craftsperson/artist, there's bound to be conflict. What you consider to be an ingenious, elegant creation too often they'll view as something that "doesn't work." That 70% figure quoted earlier represents a clash in perceptions.

Perhaps, then, the first question to ask is: Do we want a methodology? Are we hobbyists or professionals? Neither is right or wrong; a problem occurs only when we confuse the two. Methodologies are crafted so we can predictably and reliably produce something and do it again when needed. They're suited to the needs of professionals. Software methodologies are crafted so that we can predictably and reliably produce successful software deployments. As Steve Jobs once pointed out to a group of programmers, "Real artists ship."

If we've decided that we're ready for a methodological approach to software development, there are many methodologies available, or you can produce your own. What makes a methodology a methodology is "an organized, documented set of procedures and guidelines," as defined in the dictionary. Too often, developers are asked to build a pyramid, without being given a methodology for building one. There are many methodologies available, but they tend to sort themselves into groups. As Mason Cooley put it, "Methodology is applied ideology." Let's look at a few.

The No-Methodology Methodology

Because there's a lot to master when learning programming, it's understandable that developers concentrate on mastering the alien syntax of a language first. Little reserves are left for reflection on methodologies – ways of approaching not just this problem, but problems in general.

Often, then, the first methodology unwittingly chosen is the most popular: the no-methodology methodology (NMM). Adherents of NMM may find that they each approach problems differently, but underneath this apparent diversity they're united in their approach: do whatever seems good at the time. The NMMers motto is adapted from the movie, *Blazing Saddles*: "Methodology? We don't need no stinkin' methodology!"

I once had a long discussion with the head of a well-respected consulting/development firm. I asked him if they used a specific methodology for their development. "Yes, indeed," he began. "We certainly do, and if I can ever figure it out, I'm going to write it down!" Here was someone who, for lack of forethought, stumbled into NMM rather than making a conscious choice. I suspect he exemplifies most NMMers.

There are a couple of problems with NMM. First, it doesn't scale well. If as a single coder I write a small application, I may well get by without rigor in my development methodology. But if the size of the project becomes large – as many often do – NMM will fail me. I'll find myself jumping from one file to another to another, not remembering which variables are being passed in, which ones are persistent, what scope variables are, what architecture…well, you get the point.

The no-methodology methodology also doesn't distribute well. A look at virtually any mature industry shows that a division of labor has evolved, allowing different people with different skills to all contribute to a project. "Many hands make light work," they say.

In a software project this causes the application to be broken into pieces. Now I can have other developers help me write code, database experts work on data design and queries, and layout specialists build the user interface. But this requires us to understand the whole well enough to break it into its parts – something that NMM can't offer. NMMers only know the whole (to whatever degree it's fully known) when it's finished. NMM almost guarantees high stress and long nights of lonely coding for its adherents.
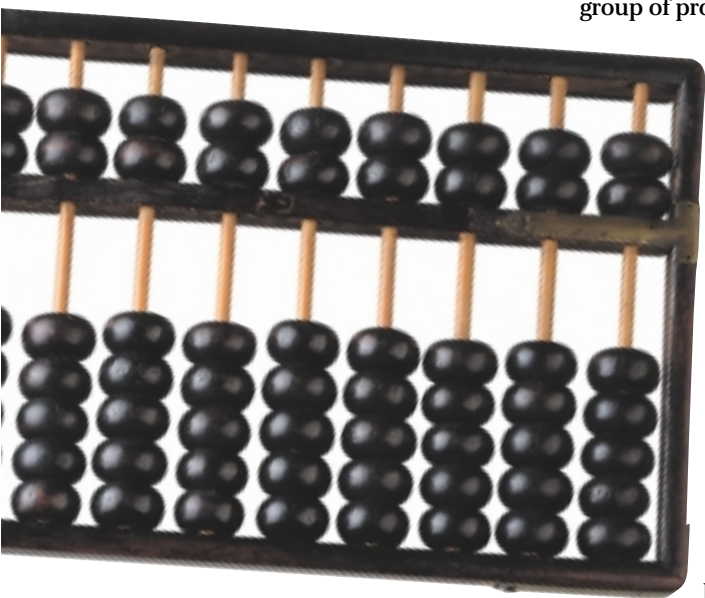
> **It's my experience that in the vast majority of cases, the success or failure of a software product is determined before the first line of code is written**

As tough as NMM makes life for the writers of an application, the lives of those maintaining that application are exponentially worse. At least the original coders have some sense, however ill-defined, of how the application works. But what of the person who comes along after the application is written and must make changes to that code. He or she has no such advantage. As you might guess, NMMers are not very good about documenting their applications, leaving the maintenance programmer to try to understand the program from raw code, and lots of it. Talk about a Sisyphean task.

The 'Best Developers' Methodology

Let's leave the NMMers to their unhappy lot and look at another popular methodology, which I dub the "Best Developers" methodology. Their motto is vigorous and has an undeniable macho appeal: "Our programmers can beat up your programmers." I first learned of this one while talking with the CTO of a large, well-known dot-com. The subject of methodologies came up. "Our methodology," he told me, "is to hire the best coders and get out of their way."

But if his plan for success involved hiring the best coders, didn't this imply that failure was the result of hiring less than the best coders? Wasn't the whole "hire the best" plan an attempt to transfer all responsibility for software project failure to the programmer? Yet, as I said, my experience was that failed software projects had already failed before the programmer wrote a single line of code. The failure just wasn't apparent yet.

What if other disciplines were to adopt this "methodology"? Would we hear the head of the Federal Aviation Agency discussing their new plan to "hire the best pilots – and get out of their way?" The FAA chief wouldn't be the only one "getting out of their way."

The more I thought of this the more I disliked the approach. While seeming to flatter the chosen developers, it really paints a bull's-eye on them. When failure inevitably follows, then those developers involved must not truly be "the best." Such a shell game may insulate the manager from real responsibility by providing a whipping boy, but I am convinced it would do nothing to deal with the real problem of software failure.

The Proprietary Methodology

Another popular one is the proprietary methodology. It works tremendously well, but it's secret! We can't tell you what it is or, rather, we could tell you what it is, but then we'd have to kill you.

There are several problems with this approach. Because it's secret, it can't be peer-reviewed. Maybe it really is the best thing since sliced bread, but how can we verify this if the method can't be revealed? One of the advantages of an open methodology is that it's open to feedback from others. Proprietary methodologies, by their nature, are closed and secretive.

From the point of view of the client, proprietary methodologies tend to lock them into a single company. What assurance does the client have that the company will be responsive to their future needs or even (as we saw in the meltdown of dot-coms) that it will be around at all?

From the point of view of the individual programmer, working in a proprietary shop is a career dead-end. The investment made in learning the secret methodology is almost certainly useless, and the programmer has lost time that could have been spent learning a standard methodology that would put him or her in a better place for a career move.

Open Standards

It seems odd that we must reexamine the question of whether open standards are better than proprietary techniques, but the Proprietarians argue that the benefits of their secret methods outweigh the disadvantages. I often wonder if the secrecy just masks a no-methodology methodology from public view.

Remember the head of the technology firm who would write down their methodology once they figured it out? Well, apparently, they did figure it out, as the company transformed their "no methodology" into a "secret methodology" that they now tout as being superior. Sadly, of course, it's secret and so you'll just have to trust them. Isn't it possible, though, that the new secret methodology is just the old no-methodology methodology renamed for marketing purposes?

Open methodologies offer several significant benefits over either of the discussed alternatives. They give coders an incentive to become expert in a methodology. Developers know that the time and money they invest in improving their skills in an open methodology will pay off. Even if their current employer doesn't recognize their worth, there are other employers interested in their proven abilities to work within an open standard. Conversely, if my only skill is in your particular methodology, then I can only bargain with you. Such an arrangement disempowers workers.

Quill Design
www.quilldesign.com

It's not only programmers who benefit from an open methodology. Development shops that adopt open methodologies are much more likely to find contractors who can help them with overflow or temporary work. Shops are more likely to take on more projects, knowing that there probably won't be a large learning curve to overcome – there will be a number of developers who already know the methodology and can step into the project and begin contributing immediately.

Open methodologies also let companies leverage the investment they make in tools. Without a standard, companies are reluctant to invest in tools that make programming more productive and less tedious. Why invest in one set of tools for one developer using one methodology, when there's no expectation that other developers can also use this tool? But with an open standard agreed upon in advance, companies are much more willing to invest in tools. Their investment can be leveraged (and amortized) over many developers (present and future) and is not merely catering to the personal likes of one developer.

Clients, too, benefit from open standards. The investment they make in code can be leveraged. No one likes "solesource" solutions and clients are more likely to invest in coding projects if they have confidence that there are many competent sources to turn to.

In short, a public methodology with open standards benefits all involved – developers, development shops, and clients. In this category are several that work with ColdFusion: Blackbox (www.black-box.org), Switchbox (www.switch-box.org), and CF-Objects (www.cfobjects.com). They result from the hard work and intelligence of their creators and are all worth looking at.

To get the full utility from a standard, however, it must be widely adopted. Without this, a standard is simply an idea. Only when people adopt the idea does it become a standard. In this regard, one methodology has an enormous advantage over all others. While advocates of the methodologies listed above number in the hundreds, Fusebox advocates number in the thousands. There's a vital, active community of Fuseboxers, Fusebox training is available, Fusebox publications exist, Fusebox has been ported to multiple languages, and the Fusebox methodology is international.

Why Fusebox?

Fusebox is built on three key principles:
1. **Modularity:** Breaks the application into smaller, more tightly defined pieces ("circuits" and "fuses")
2. **Severability:** Makes modules as independent of each other as possible by eliminating, where possible, fixed references and replacing them with variable references
3. **Clarity:** Documents the interface (API) to individual code files through structured comments

I said earlier that Fusebox was, by far, the most popular development methodology for ColdFusion. While popularity with regard to an effective methodology is very desirable (more developers skilled in the methodology are available), the methodology itself must provide real benefits; popularity alone is no sure guide, as your mother used to remind you. What are the benefits thousands of developers have discovered, and what relationship does the Fusebox principles have to these benefits?

Benefits of Fusebox

One benefit arises from the ability to build independent software modules: separate developers, or separate teams of developers, can work on individual components simultaneously. This is a goal of virtually all methodologies, but few do this as well as Fusebox. This feature of Fusebox means that software can be built faster, as the need for serial correlation ("I must wait to do my task until you have finished yours") is greatly reduced.

It turns out that this *severability* allows software components, on both a large and small scale, to be reused more easily. Apart from the obvious cost benefit of not having to write software more than once, *reusability* means that developers can use pretested code, which both lowers the cost and decreases the risk of failure on a project.

Another key feature of Fusebox is an integral documentation/structured comment system known as Fusedoc. Comments are written in XML format. A document type definition (DTD) provides structure and allows other software to work with them more easily. The Fusedoc tells the coder what the individual file is responsible for, what variables are available at runtime, as well as variables the coder is responsible for creating.

The Fusedoc forms a sort of "work order" that the programmer works from. In the same way that machinists work on parts without knowing what they will be used in, Fusedocs let developers work on code without knowing about the application, or even the underlying database. The implications of this are enormous. Some 25 years ago Fred Brooks wrote the classic software text, *The Mythical Man-Month*. Brooks pointed out that adding people to a project that's late actually increases the amount of time needed to complete the project. The reason? It takes time for those new developers to understand the application enough to contribute. Add to that the increased communication and coordination time needed to integrate those extra workers, and adding developers only makes a bad situation worse.

Part of Fusebox's architecture planning involves separating an application into separate modules (or "circuits"). These modules are then broken into separate code files (or "fuses") and Fusedocs are written for fuses. At the point of coding, a developer is working on a single fuse. He or she knows what the fuse is responsible for and the input/output parameters, but needs to know little else. Now, if the project begins to run late, additional coders can be added to it. The ability to have code completed by competent developers who don't need to know about the application itself means that we can compress the time needed to actually write the code, typically far less than what any experienced developer would think possible.

Such an abbreviated coding time requirement radically shifts the equation of software development. Instead of rushing through the requirements, design, and architecture in order to get to the "real work" of coding, the requirements, design, and architecture become the real work. Coding is simply a matter of "scribbling down the details" as Mozart once remarked of the process of writing down the notes inside his head.

The ability to have multiple teams work productively on the same project must come as pure joy to any project manager. It also greatly lessens stress on the coders, who – freed of the arduous, thankless task of trying to guess what the client wants – can concentrate on writing beautiful code.
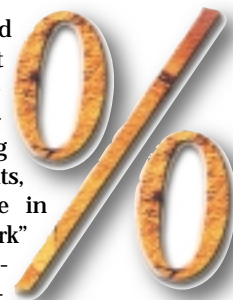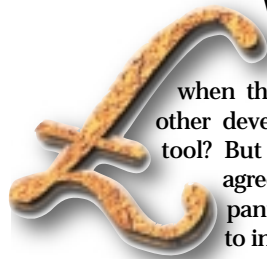
Fusebox also makes it possible for people with different skills and talents to contribute to the project. A Fusebox project often includes architects, database experts, an art director and graphic artists, an interface specialist, HTML/layout specialists, and ColdFusion coders. This is quite different from other development environments in which coders must handle all of this by themselves. Fusebox does this by separating an application's logic from its display. With the division of labor allowed by Fusebox, the overall cost of a project, the time to completion, and the risk of failure are all reduced.

Reusability is the holy grail of programmers. The appeal is obvious – write it once and use it over and over; if only it were so simple in practice. Every methodology tries to achieve it with limited success. Yet all that effort has taught us some things about reusability. The most important lesson is: all things being equal, it's easier to reuse a little bit of code than a lot. In short, simpler is better, so Fusebox encourages the use of small, well-defined code files (or "fuses") to complete larger tasks.

If complexity is the chief problem with code reuse, application-specific code must also rank high on a list of "Enemies of Reusability." Fusebox can help with this by providing for XFAs, a method where even page links can be set programmatically.

Once an application is built, the real costs begin. Maintenance of code normally accounts for as much as 80% of the total life-cycle cost. Much of this is due to the chronically poor documentation that forces maintenance programmers into detective work. Unless maintenance is planned for, it can quickly spiral into a black hole of costs for a company. More than one company has found itself saddled with obsolete software that's simply too complex and costly to adapt to new circumstances.

Fusedoc, an integral part of Fusebox, is enormously helpful to coders charged with making changes to others' code. Because each code file is well documented, the task of maintaining code, never a joy, is at least manageable.

No More Holy Wars

Anyone who has been involved in software knows the battles that rage over different languages, different operating systems, and different methodologies. We feel passionate about what we do. We ally ourselves with those who agree with us and avoid those who do not. But as we have seen all too clearly in the last few months, holy wars are capable only of creating suffering; they are powerless to change the hearts and minds of others.

I have personally found Fusebox to be very helpful to many developers I train and consult with. I use it exclusively on the work I do. Yet, I know some excellent developers who use other methodologies to great effect. It's important to remember that the purpose of a methodology – any methodology – is not to perpetuate itself, but to produce some desired result. What we all want, as professionals, is to reliably and predictably produce great software. Methodologies are important to the degree that they help us reach our goal, and that's something we can all agree on.

About the Author

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

@ HALHELMS@TEAMALLAIRE.COM

A Cold Cup O'Joe Part 7 of 8

Extending Java CFX Tags

BY
**GUY
RISH**

n Part 5 (**CFDJ**, Vol. 3, issue 8), "Java CFX Basics," I identified some shortcomings of the CFX model. In Part 7, I demonstrate a number of workarounds and solutions to these problems.

In many cases what I present here is viable across the board for all CFX types. However, this series deals specifically with Java interoperability, so I'll leave the modifications to you.

Complex Data Types

One of the critical problems with the CFX model is the inability to pass complex data types into and out of a CFX. Only simple types are possible – no structures, no arrays, and only a single query (using the QUERY attribute). This can be a considerable challenge when attempting to integrate with higher order applications that must work with large amounts of data using one of the "restricted" data types.

As it turns out, Allaire (before they became part of Macromedia) provided a powerful tool for data marshaling – Web Distributed Data eXchange (WDDX). WDDX has been a part of the regular offering in CFML since version 4.0, so encoding complex data types to be sent to a CFX is simple. It's a little more complicated on the other end though. Fortunately, as part of the WDDX SDK (www.openwddx.org), there's a Java class API that resolves most simple marshaling issues.

Using WDDX allows the CFML data types to be marshaled into analogous Java data types. CFML arrays become Java Vectors, CFML structures become Java Hashtables, and CFML queries become a spe-cialized Java class from the WDDX API RecordSet.

To make the WDDX SDK work, an XML SAX parser is needed; I recommend Apache's Xerces parser (http://xml.apache.org). Once the WDDX SDK is downloaded and installed, it's important to include the WDDX SDK's JAR and the Xerces JAR in the ColdFusion Server's Java classpath setting. This can be done from the ColdFusion Administrator's Java Settings panel.

The methods of the WddxDe-serializer class (see Table 1) are intended to convert WDDX packets into various Java data types.

Conversely, the methods of the WddxSerializer class (see Table 2) are intended to convert various Java data types into WDDX packets.

Deserializing

Making slight modifications to the "Hello, World" CFX that we've been using since Part 5, I've created a new tag called CFX_HelloAll (see HelloAll1.java). (The source code for this article can be downloaded from the *CFDJ* Web site, www.sys-con.com/coldfusion/sourcec.cfm.) This tag receives a WDDX serialized array of strings as a Names attribute. Retrieve the attribute in the normal CFX fashion by calling the getAttribute method of the custom tag's Request object. Since the array is serialized as a WDDX packet, it passes as a regular character string, which is completely legal in the CFX model. This does, of course, require the attribute to be deserialized within the custom tag. I've conveniently created a method called deserialize-FromPacket, which does all the work (see Listing 1).

The method is fairly simple to use and in most cases works without a problem. The first few lines check the existence of the incoming argument. An interesting point in the method occurs on line 23, where the SAX parser to be used with the WDDX engine is set; this references a public and final property class called DEFAULT_SAXPARSER.

An attempt to create an instance of this class is made on lines 26–34 to verify that it's in the JVM's classpath. Then the packet is buffered in preparation for use on line 37. An instance of the WddxDeserializer is created, passing the classname of the SAX parser on line 40. On line 43 the deserializer is fed the buffered packet, then on line 45 the resultant object is returned to the calling logic.

The deserializeFromPacket is called in the tag's processRequest method in the case of the HelloAll tag. It expects the returned object to be a Vector (and casts it so), which it loops over, performing a write to the tag's Response object.

All this is called from the Cold-Fusion template, PacketExample1.cfm,

Method	Description
WddxDeserializer	The class constructor
deserialize	Converts a WDDX packet into a Java object

TABLE 1 WddxDeserializer class methods

Method	Description
WddxDeserializer	The class constructor
getDefaultSerializerFactory	Returns the default WDDX serializing factory
getSerializerFactory	Returns the active WDDX serializing factory
serialize	Converts a Java object into a WDDX packet

TABLE 2 WddxSerializer class methods

which merely creates an array and populates it with three names, uses the CFWDDX tag to convert the array into a WDDX packet, and passes it when it calls CFX_HelloAll.

Serialization

Another rehash of the tag Hello-All2.java does pretty much what the previous version did, but instead of directly streaming out the messages, it loads them into a Vector. The Vector of strings is then serialized into a WDDX packet and returned to the calling logic, PacketExample2.cfm. The packet is then deserialized into a CFML array, which is looped over and displayed. The most interesting part in this new pairing is the serialization method in the CFX, serializeToPacket (see Listing 2).

This method is even simpler than the deserialization method. The WddxSerializer is instantiated on line 5. Lines 8–13 test the incoming object to be serialized and line 16 does the serialization, receiving the incoming object and the buffer instantiated on line 6 to hold the resultant packet. The contents of the buffer are returned to the calling logic on line 18.

End Tags

The CFX model has no capabilities to create custom tags with end tags, and there's no simple way around this challenge. There are rumors that ColdFusion's next generation platform has a solution, though that's little comfort to people running on current releases. While the CFX model is not prepared to deal with this, CFML is. Leveraging a CFML custom tag as a wrapper around a CFX tag allows for more advanced designs.

I've created a CFX that performs XSL transformations on an inputted character stream. This tag, Transformation.java, doesn't require any special construction to achieve its end-tag capability. This is accomplished through a CFML custom tag, Transformation.cfm, which wraps it. The ColdFusion template, Transformation.cfm, which wraps it. The ColdFusion template, TransformationExample1.cfm (see Listing 3), shows the custom tag combination in action. Lines 1–4 are the setup for using the custom tag, building the name of the XSLT stylesheet to use and the data file that's used to dynamically construct

the XML. The tag opens on line 5, passing the dynamically constructed stylesheet name as an attribute; the body of the custom tag occupies lines 6–15; and the tag closes on line 16. To properly process all the variables, the body must be wrapped within a CFOUTPUT block. It's here that the XML is dynamically constructed, driven from the data drawn from the contents of the data file read on line 4.

The custom tag uses the This-Tag.ExecutionMode to determine whether it's processing. When the ExecutionMode is "start," the tag checks the incoming attributes. If the Source attribute is specified and there's no body, the CFX is called (this example is not shown), otherwise the tag body is processed. When the end tag is processed and the ExecutionMode is "end," the ThisTag.GeneratedContent is passed (as a regular String attribute) with the value of the Stylesheet attribute to the CFX.

Return Variables

Returning a value from a CFML-wrapped CFX requires an additional step. The regular way of doing this is through the CFX Response object's setVariable method. This remains true even with a CFML-wrapped CFX; the additional step results from the fact that the setVariable method is actually returning the value to the wrapping custom tag, not the top-level calling logic. This means that once the CFX returns the value, the CFML wrapper also needs to pass the value to its calling logic.

The ColdFusion template, TransformationExample2.cfm, sets up a condition for a returning variable as discussed. Much like the code in Listing 3, it dynamically constructs the stylesheet name, but uses a finished XML document to hand off to the custom tag and specifies a return variable name of "out".

```
<cfx_transformation
stylesheet="#Attributes.Style-
  sheet#"
source="#Attributes.Source#"
result="#Attributes.Result#">
 <cfset
"caller.#Attributes.Result#" =
Evaluate("Variables.#Attributes.
Result#")>
```

ABOUT THE AUTHOR

Guy Rish works for DevX, an online IT resource center. He holds instructor certifications from Rational Software and Allaire and teaches Web development for SFSU MSP. Guy is active in the ColdFusion and OO communities. He was the technical editor for Mastering ColdFusion 5 from Sybex.

This code snippet shows the two important lines from the CFML wrapper tag, and the incoming variable, Attributes.Result, being passed to the CFX as the Result attribute. Immediately after the CFX tag is called, the resulting variable is passed up the remaining layer to the calling logic. Since the CFX value will be passed into the Variables scope of the CFML custom tag, the CFML custom tag must create it and assign the value in the CFML tag's Caller scope.

Attributes Structure

One of the truly useful features of CFML custom tags constructed with CFML is the Attributes scope. This is another feature unavailable to tags written in the CFX model. Using the CFML-wrapped CFX strategy and WDDX serialization provide a kind of workaround to this shortcoming. To implement these example changes, I remade the XSL transformation tag set by creating new CFML tags, CF_TRANSFORMATION2 and CFX_TRANSFORMFORMATION2 (see Transformation2.cfm and Transformation2.java, respectively).

The alterations in the CFML custom tag center around two lines of code: the serialization of the entire Attributes scope into a packet and then passing it, and only it, to the CFX as the value of the Attributes Collection tag attribute.

```
<cfwddx action="cfml2wddx"
input="#Attributes#"
output="packet">
<cfx_transformation2 attributes
collection="#packet#">
```

The CFX alterations are a bit more involved. The inclusion of the deserializeFromPacket method used in the previous examples allows the serialized Attributes scope to be transformed into a Java Hashtable (see Listing 4).

Beyond these simple changes the rest of the tags remain the same. Using these tags from a user standpoint is identical to using the previous version, with the exception of its name; this is reflected in the ColdFusion template, TransformationExample3.cfm.

Wrapping It Up

There are a number of ways to overcome the immediate shortcomings of the CFX model. Combining CFML with the CFX model, Java's extensive libraries, as well as other tools and technologies, like WDDX, provide creative ways to do this. The techniques demonstrated in this article are for general use and may be incorporated easily enough into regular custom tag construction. There are, as I'm certain will be discovered through further study, specialized techniques that can be utilized from time to time. Further, when ColdFusion's next generation is released, the enhanced custom tag models and new technologies will extend this even more.

@ GUYRISH30@YAHOO.COM

Listing 1

```
01: private Object deserializeFromPacket(String packet)
02: throws Exception
03: {
04:   String parser = null;
05:   Object obj = null;
06:   InputSource source = null;
07:   WddxDeserializer deserializer = null;
08:
09:   // check incoming argument
10:   if(packet == null || packet.length() == 0)
11:   {
12:     // throw error
13:     throw new Exception("The incoming packet is NULL
       or empty.");
14:   }
15:
16:   // TODO:
17:   // setup configuration code for retrieving other
       possible SAX parsers
18:
19:   // set parser value to internal
20:   if(parser == null)
21:   {
22:     // use default parser
23:     parser = this.DEFAULT_SAXPARSER;
24:   }
25:
26:   // attempt to load the parser for verification it is
     in CLASSPATH
27:   try
28:   {
29:     Class.forName(parser).newInstance();
30:   }
31:   catch(Exception e)
32:   {
33:     throw new Exception("Unable to load SAX parser
       class.<br>The system was unable to load the
       required SAX parser classes: " + parser +
       ".<br>This might be because it are not in the
       ColdFusion Server's Java CLASSPATH setting.<br>");
34:   }
35:
36:   // create input source for the XML parser
37:   source = new InputSource(new StringReader(packet));
38:
39:   // create a WDDX deserializer
40:   deserializer = new WddxDeserializer(parser);
41:
42:   // deserialize the WDDX packet
43:   obj = deserializer.deserialize(source);
44:
45:   return obj;
46: }
```

Listing 2

```
01: private String serializeToPacket(Object obj)
02: throws Exception
03: {
04:   String parser = null;
05:   WddxSerializer serializer = new WddxSerializer();
06:   StringWriter buffer = new StringWriter();
07:
08:   // check incoming argument
09:   if(obj == null)
10:   {
11:     // throw error
```

```
12:     throw new Exception("The incoming object is
        null.");
13:  }
14:
15:  // serialize object
16:  serializer.serialize(obj, buffer);
17:
18:  return buffer.toString();
19: }
```

```
01: <cfparam name="url.sheettype" default="html">
02: <cfset stylesheet =
    GetDirectoryFromPath(GetBaseTemplatePath()) &
    "booklist.#url.sheettype#.xsl">
03: <cfset booklist =
    GetDirectoryFromPath(GetBaseTemplatePath()) &
    "booklist.txt">
04: <cffile action="READ" file="#booklist#"
    variable="books">
05: <cf_transformation stylesheet="#stylesheet#">
06: <cfoutput>
07: <booklist>
08:  <cfloop index="idx" list="#books#">
09:     <book>
10:             <name>#ListGetAt(idx, 1, "|")#</name>
11:             <link>#ListGetAt(idx, 2, "|")#</link>
12:     </book>
13:  </cfloop>
14: </booklist>
15: </cfoutput>
16: </cf_transformation>
```

```
// grab attribute collection
 attribAttributesCollection =
 request.getAttribute("ATTRIBUTESCOLLECTION", null);
 if(attribAttributesCollection == null)
 {
  throw new Exception("Error! Missing required attribute:
ATTRIBUTESCOLLECTION");
 }
 mAttributesCollection =
(Hashtable)deserializeFromPacket(attribAttributesCollection
);

 // retrieve/default the incoming argument
 attribStylesheet =
(String)mAttributesCollection.get("STYLESHEET");
 attribSource =
(String)mAttributesCollection.get("SOURCE");
 attribContent = (String)mAttributesCollection.get
 ("CONTENT");
 attribResult =
(String)mAttributesCollection.get("RESULT");
```

**CODE
LISTING**
▸▸▸▸▸▸▸▸▸▸▸▸▸▸
The code listing for
this article is also located at
www.sys-con.com/coldfusion/sourcec.cfm

To Err Is Human, to Gracefully Handle Errors **Is Divine**

BY
BEN FORTA

Taking the high road with exception handling

Have you ever seen a ColdFusion error page – the bordered box on a white background that basically says someone (or something) messed up? I'm sure you have, as I'm sure your users have too. Errors, and error messages, are an unfortunate fact of development life. But while you may have to live with occasional errors, you definitely do not (and should not) have to live with that too-oft-seen CF error screen.

Prettier Error Pages

The standard ColdFusion error screen is very informative and useful. So what's wrong with it? Actually, several things:
- It's ugly, simple as that.
- It won't match the look and feel of your site (I hope).
- It's informative, but sometimes much too informative (you might not want end users knowing all the gory details of your filenames, paths, databases, queries, variables, or worse, your mistakes).

> **"**…the standard error screens don't create a professional (and confidence-inspiring) impression**"**

- The sum of the prior bullets is that the standard error screens don't create a professional (and confidence-inspiring) impression.

The simple solution is to use the <CFERROR> tag, which lets you replace the standard error screens with ones of your own. For example:

```
<!--- Specify error page --->
<CFERROR TYPE="request"
         TEMPLATE="/error/stan
            dard.cfm">
```

Once this code is executed, the standard.cfm file (in the error directory) will be displayed instead of the default error screen. Within the file itself any and all client technologies may be used (HTML, JavaScript, DHTML, images, etc.).

However, CFML may not be used within <CFERROR> files. After all, if an error condition has occurred, allowing the execution of CFML could in turn generate an additional error, causing the code to execute again, generating yet another error causing…you get the idea. So no CFML tags or functions. Instead, a special set of variables is made available to you within error pages, and you can place them in your code to refer to things like:
- Date and time
- Browser
- Client IP address
- Template name
- Referring page

Using <CFERROR> and these variables it's possible to create consistent and better-looking error screens, while also controlling (to some degree) the amount of information presented to users.

<CFERROR> can also be used to define other error screens, for example, the screen displayed when form validation fails.

> Note: *<CFERROR> needs to be executed on each request, so the ideal location for this tag is the Application.cfm page.*

Understanding Exception Handling

<CFERROR> and error handling are very useful in presenting a better and more professional end-user experience, but <CFERROR> is not enough. After all, displaying a prettier error screen is wonderful but:
- When an error occurs you still won't know about it.
- If an error occurs, it will likely keep recurring; with <CFERROR> you can't fix or handle the situation.
- When an error occurs, processing stops; you can't gracefully handle the error and keep going.

In other words, <CFERROR> lets you replace the error page, but that's it. It doesn't let you handle errors – processing them or taking actions when they occur.

That's where *exception handling* comes into play, with an emphasis on *handling*. Exception handling is a mechanism by which errors (or other conditions) may be trapped and then responded to. Using exception-handling error conditions doesn't halt processing, instead processing is diverted to special blocks of code that you can write – code that can do just about anything you need.

Try and Catch

Exception handling in ColdFusion is implemented using a *try/catch* interface, implemented (as you'd expect) using tags named <CFTRY> and <CFCATCH>. The basic flow is like this:

```
start code block
   CFML code goes here
   error handling block
      error handling code goes
      here
   end error handling block
end code block
```

Any code in which exception handling is to be used must be flagged – the start and end so noted. Exception-handling code is then placed at the end of the code block. Then, if an exception occurs within your code, processing is stopped and control is transferred to the error-handling code.

The CFML code looks like this:

```
<CFTRY>
   ... CFML code ...
   <CFCATCH>
      ... CFML code ...
   </CFCATCH>
</CFTRY>
```

<CFTRY> and </CFTRY> delimit the code in which exceptions are to be trapped, and ColdFusion tries (and thus <CFTRY>) to execute it. If an exception occurs (this is actually referred to as an error being *thrown*), it's caught by the exception-handling code (and thus <CF-CATCH>).

For example, if you want to send an e-mail to an administrator when an exception is thrown, you could do something like this:

```
<CFTRY>
   ... CFML code ...
   <CFCATCH>
      <CFMAIL
TO="admin@domain.com"
          FROM="
admin@domain.com"
          SUBJECT="Error
on #GetCurrentTemplatePath()#">
      TYPE:
      #CFCATCH.type#

      MESSAGE:
      #CFCATCH.message#
      </CFMAIL>
   </CFCATCH>
</CFTRY>
```

The <CFCATCH> block here contains a <CFMAIL> tag, so when an exception is thrown, an e-mail message is sent to the specified address and processing continues (to whatever code comes next).

Handling Specific Exceptions

As you can see, trapping and handling exceptions is not complicated at all. But what if you want to handle different types of exceptions in different ways? That's doable too – all you need is multiple <CFCATCH> blocks, one for each type of exception.

Look at the following code snippet:

```
<CFTRY>
   ...
   <CFCATCH TYPE="missingin
      clude">
      ...
   </CFCATCH>
   <CFCATCH TYPE="database">
      ...
   </CFCATCH>
   <CFCATCH TYPE="lock">
      ...
   </CFCATCH>
   <CFCATCH TYPE="any">
      ...
   </CFCATCH>
</CFTRY>
```

Here four catch blocks are used. The first catches failed <CFIN-CLUDE> tags, the second catches database problems (including ODBC and SQL errors), the third catches <CFLOCK> errors (including time-outs and missing locks), and the fourth catches all other errors. It's a good idea to always have one <CFCATCH> with TYPE="any" to catch any exceptions not specifically caught, and to specify it last (otherwise it'll catch exceptions that you might not want it to).

Other exception types are supported, including over 50 advanced types (used to catch very specific exceptions, like <CFPOP> authentication failures) and custom types used to define exception types of your own.

Using Exception Handling

We've only just scratched the surface here; there's lots more to exception handling and what you can do using this functionality. Here are some things to consider:

- The earlier examples used exception handling purely for notification purposes, but as you can execute any CFML code within <CFCATCH> blocks, you can do just about anything (including executing <CFQUERY> tags, writing log files, setting APPLICA-TION variables to point to other servers, and more).
- Exception handling can be nested. This allows you to provide high-level handling for complete pages, and more granular handling for specific operations. For example, placing a <CFQUERY> tag within its own <CFTRY> block would enable you to retry timed-out queries, or resubmit them to a backup database.
- When nesting exception handling, exceptions can be passed to outer <CFTRY> blocks by re-throwing them using the <CFRE-THROW> tag.
- Developers can throw (generate) their own exceptions using the <CFTHROW> tag. For example, an exception could be thrown if required parameters are not passed, so as to centralize processing and the displaying of error messages.
- The code in custom tags should have its own internal exception handling; mechanisms exist for passing exceptions to calling pages if needed.

And that's just the start of it.

Summary

At a minimum, every developer should be using <CFERROR> to override the default error screens with ones that better match the look and feel of the application. However, <CFERROR> is just the beginning. Exception handling is not difficult to implement and should be used to create better applications – better for you and better for your users.

Neo, which I'll be covering extensively starting next month, extends ColdFusion's try/catch handling even further. If you have yet to play with these tags, there couldn't be a better time to start.
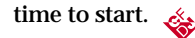
BEN@FORTA.COM

ABOUT THE
AUTHOR
Ben Forta is Macromedia's senior product evangelist and the author of numerous books, including the recently published ColdFusion 5 Web Application Construction Kit *and its sequel,* Advanced ColdFusion 5 Development. *For more information on Ben's books visit www.forta.com.*

Microsoft Access

Most ColdFusion developers who work with databases have a favorite, but it usually isn't Microsoft Access. This is not completely irrational and they can give you five reasons why it's unconstitutional to use Microsoft Access. This article attempts to explain when a developer *should* use Microsoft Access.

Consider the situation in which a client has hundreds of man-hours invested in an Access front end that performs everyday tasks, such as managing mailing lists or paper surveys. I propose that Access should be maintained as a front end and the data should be moved to an enterprise database like SQL Server or Oracle.

I'll explain how to use ODBC with Access and SQL Server to reduce development time and maintain the existing functionality of an Access front end. Fewer hours spent developing a ColdFusion alternative to a database's existing administration pages can greatly impact client relationships, keep the client's staff happy, and reduce the amount of coding or debugging you have to do.

Microsoft Access fits into the ColdFusion developer's toolbox because overengineering a project can kill both you and the job you're working on. I'll begin with the available options before summarizing the factors I consider when deciding what to build for a client.

Building Web site data administration pages

The Options for Administration Pages

If you develop the client's Web site using ColdFusion to display the information from the database, you have several options for editing the information in the database (which I'll refer to as administration pages):

- SQL Server using an Enterprise Management Console in which administrators would enter the data directly into the table
- ColdFusion/ASP or other server technology connecting to a SQL Server or Access database
- Microsoft Access using FTP to upload the entire database when there's a change in the data it contains
- Microsoft Access connecting to a SQL Server over ODBC to maintain the existing forms and functionality

Ain't That Bad!

There may be other options too. Let's look at each of the methods I've suggested in more detail.

SQL Server Enterprise Management Console

Create a database in SQL Server and import the client's existing tables and queries from Microsoft Access using the Import Wizard. Next, create a login for the client to access the database. The client should be allowed access to add, edit, and delete records from the tables in their database, but denied the ability to drop tables or modify table properties. They should also be denied access to other databases on the server. To do this, navigate to the "users" icon in the client's SQL Server database. Click New and enter a login name. Next select "db_datareader" and "db_data writer" from the list of roles and click OK to add the user.

Why bother creating a separate user login for the client? It prevents users from poking around in other databases and minimizes their potential to make critical mistakes. You'll need to install the SQL Server Enterprise Management Console on each computer that will be used to administer the data, train the staff on the interface, and explain what the error messages mean.

The key benefit of this solution is that strict validation can be enforced. Basically, users can't break what they can't touch. Another plus is the ability of the SQL Server database to support many more concurrent users and the ability to grow beyond 1GB in size.

On the negative side, the client machines need a reasonably fast Internet connection to access large chunks of data, and the learning curve for staff who struggle with Microsoft Access may be too steep – it doesn't come with the comfort of an Access GUI. Another issue is that there may be aspects of the client's original access database code that can't be built in SQL Server, for example, reports and mail merging.

ColdFusion

What about building in data administration pages in ColdFusion? Maybe you could use a tool like UltraDev to speed up the process. Look at the elements involved:
1. Convert the database to SQL Server (for a popular/high traffic site) or just use the original Microsoft Access database.
2. Define a new ODBC connection in the ColdFusion Administrator.
3. Create ColdFusion login pages.
4. Create ColdFusion pages to retrieve records.
5. Create ColdFusion pages to display records.
6. Create ColdFusion pages to validate user submissions.
7. Create ColdFusion pages to write changes to the database.
8. Trap and handle any potential Cold-Fusion errors.

Security is important with data administration pages to prevent the average Internet menace from adding bogus data or destroying your client's hard work. The display should at least be as straightforward to the administrator as the Microsoft Access administration pages, but ideally it should improve the administrator's experience.

At a minimum you should implement server-side validation to ensure that trash doesn't get into the database, and ideally you should try to trap the errors on the client-side with a scripting language such as JavaScript. With that in mind, ColdFusion errors should also be anticipated and trapped so, for example, incomplete database transactions are rolled back. To do this, make extensive use of cftry and cfcatch blocks in your code. Your final line of defense is to create and enforce strict rules in the SQL Server/Microsoft Access database.

Why all this fuss? A user can easily correct mistakes with simple steps in Microsoft Access, but a ColdFusion administration

	SQL Server	ColdFusion	Access FTP	Access ODBC
Existing functionality	Low	Low	High	High
Technical ability	High	Low	Medium	Low
Willingness to change	Yes	Yes	No	No
Number of visitors	High	High	Low	High
Frequency of changes	High	High	Low	High
Connection speed	High	Low	Medium/High	High
Number of editors	More than one	Many	1	More than one

TABLE 1 Recommendations for selecting an appropriate method to build administration capabilities for a Web site.

page must exist to allow users to correct mistakes not trapped in the validation routines. If they can't correct things immediately and the data is used to drive dynamic content, the Web site will be inaccurate until the error can be fixed or the ability to correct the error has been presented to the user.

Why ColdFusion?

Many editors may need to work on data administration at one time, but not necessarily at the same location. This approach works well for slow Internet connections since each page request is a small defined transaction and the only special software it requires is a browser. Unfortunately, the amount of design work required can be prohibitive in terms of time and complexity. Even tools like UltraDev can't eliminate the need to hand-code some of the administration pages. The staff will still need to be trained on the new interface and you still have the problem of implementing all the functionality from the Microsoft Access version.

Microsoft Access FTP

Not all Web sites are destined to become the next Amazon or as popular as Yahoo. Some people just want to make small adjustments to their Web site once or twice a year. They don't have many visitors per month, let alone per day, but when they do, the information needs to be up-to-date. In this scenario, the latest information is recorded in a small Microsoft Access database and the Web site displays information only (it doesn't capture any information to update the database).

Turning this database into a SQL Server database would be overkill, as would writing ColdFusion admin pages, since there's only one update every few months. My suggestion is to allow clients to continue working in the Microsoft Access database on their machines and either:
1. Develop a ColdFusion upload page in an NT password–protected directory of their Web site, or

2. Show them how to use FTP, maybe even write an MS-DOS batch file to execute all the upload commands for them

This option, although less robust than the other suggestions, has its benefits. Clients can continue using their old system; if their staff is capable of developing in Microsoft Access they can continue to add functionality. However, using FTP or ColdFusion to upload a database is a slow process for large databases. To this end, users must get in the habit of compacting the database prior to the upload (select Tools, Database Utilities, Compact Database in Microsoft Access). This approach means that occasionally (once or twice per year depending on the frequency of updates) a visitor to the site may experience some delay while the database is updated. If multiple copies of the database exist for editing purposes, the database administrator needs to understand the process of synchronizing database copies and should perform this before each upload.

Microsoft Access ODBC Option

What about clients with lots of Web site traffic who are either too reliant on their Microsoft Access systems or can't afford to develop a ColdFusion administration implementation? My suggestion combines SQL Server, ColdFusion, and Access. The same ODBC we've all used with ColdFusion can be used in Microsoft Access to present an interface to a remote SQL Server database.

The aim of this approach is to maintain the client's current interface in Access for data administration so he or she gets a cost-effective solution and has nothing new to learn. An ODBC-linked table can be used just like a regular table created in Access except you can't modify the table design. This means that if you name the linked tables the same as the original data tables, all the queries, forms, and reports will work the same. Users will see no change in their everyday interactions with

MACROMEDIA

www.macromedia.com/go/usergroups

the database but the Web site will be kept up-to-date. They can still create mailing labels from the same buttons in their Microsoft Access application and any custom Visual Basic code will still work.

Okay, enough of why should you do it – let's look at how: first you'll need to import the existing tables from their Microsoft Access database into SQL Server. Next, create a file-based ODBC Data Source Name (DSN) that connects to the SQL Server. For each table in the SQL Server database, create a linked table using the file-based DSN you defined. Once the tables are linked, change the names of the old tables. I suggest prefixing each table name with the word "old" so an alphabetical sort of the table names will group them together. Change the names for the new linked tables to the name used in the original Access database, then everything should work fine again. An optional final step would be to create a distribution package, in WinZip for example, that contains the database, the file-based DSN, and either an MS-DOS batch file or instructions for installation if the system needs to be placed on several machines.

It's as simple as that. To do this stage for a client typically takes around an hour to set up, compared to the several hours (minimally) required to develop Cold-Fusion Administrator pages, even with Dream-weaver. Next, I'll discuss some of the steps in a bit more detail.

Create the File-Based DSN

While a complete explanation of file-based DSNs is beyond the scope of this article, here's a rough outline of the process to create one:
1. Open data sources in Windows Control Panel (Admin Tools in Windows 2000/ NT).
2. Select File DSN.
3. Click Add.
4. Select SQL Server, enter a name, and click finish.
5. Describe the data source.
6. Enter server, e.g., (local) or if remote, e.g., sql.enterwhacked.com.
7. Pick SQL Server authorization.
8. Enter the loginID and password created in the SQL Server database for this user.
9. Hit Next (don't hit Enter here or the wizard will exit and you'll have to re-configure the settings it inserts by default).
10. Hit Next again and then finish.
11. Test the data source using the button it shows before clicking OK twice to finish the process.

Link Tables in Microsoft Access

You just created the ability to pull data from the SQL Server database. The next step is to give the users the interface to that data. Do this by creating a linked table in Microsoft Access:
1. Right-click in the database window while the "tables" tab is showing.
2. Select link tables from the pop-up context menu.
3. Select ODBC ISAM from the dropdown list of file types. The default should be Microsoft Access database – ODBC is usually the last entry in the list.
4. Select your file-based DSN from the list of DSNs.
5. The window now lists all the tables and views you can access. Place a check mark against any that you wish to link in this database.
6. Last, set any connection options. As mentioned earlier, it's generally a good idea not to set up the connection to automatically enter the username and password. Click Finish to close the wizard.

Working with ODBC in Microsoft Access

Now that your tables are linked, you can use them like any other table in the Microsoft Access database. You can:
- Use existing queries or create new ones that process data in the linked tables
- Use existing reports or create new ones driven by data in the linked tables
- Use existing forms or create new ones to present the linked tables' data using a suitable interface
- Add, edit, and delete records directly in the linked tables, in forms, or some queries

The key property of a linked table is that you can use it to do anything you could do with a regular Access table except modify the properties of the data (e.g., field length). This is a good thing since inexperienced users won't be able to accidentally modify or delete crucial tables.

Benefits of the Microsoft Access ODBC Approach

You or the users can make use of any Visual Basic for Access modules that existed in the original Access database before you created the links to SQL Server. If you changed the linked table names over to the old names, any old forms and VBA code that worked on the data will still work. Consequently, the only training required for the users is the added step to enter a username and password. Since the data is stored remotely, the Microsoft Access database can be compacted into

a very small file, which makes the interface easy to distribute by e-mail, floppy, or the intranet. The only other file you need to send is the file-based DSN. The GUI offered by Access is simple enough so you or the users can develop new interfaces to the data.

This approach does, however, rely on all the correct drivers being installed on client machines and you need a reasonably fast Internet connection for complex database options since queries are executed on the local machine, not on the SQL Server. You can process some queries on the SQL Server to reduce the amount of data sent over the connection. To do this, you need to open up an existing query and click on the dropdown for query type. You can either specify a "pass-through" query type and set connection options or re-create the queries as views in SQL Server.

Recommendations

We've covered four different approaches in this article, all of which can work in certain situations. The important point is to use whichever is most appropriate for the client. Table 1 summarizes my guidelines for selecting an approach to building data administration pages for a Web site.

Author Bio
Adam Howitt, a computer consultant in Atlanta, earned a master's in software engineering in Sheffield, England. He maintains a Web site for expats, www.connectedexpats.com.

@ ADAMHOWITT@YAHOO.COM

Ain't That Bad!

MACROMEDIA

www.macromedia.com/go/mastering

Prefab **ColdFusion Construction**

Build your ColdFusion applications modularly with prefabricated parts

BY
JOSHUA OSTER-MORRIS

**M**ost ColdFusion developers have noticed that there are many similarities between each application they write.

ABOUT THE AUTHOR

Joshua Oster-Morris has been writing software since age 6, when on his TRS-80 he began his first computer game. Since then he has deployed many mission-critical database applications using dBase, FoxPro, C++, and, most recently, ColdFusion and Java.

Most database tables require you to write a data entry screen, a form action page, and at least one SQL query. I can't tell you how many times I've typed in alternating <TR> and <TD> tags with <CFINPUT> and field descriptions between them. I've even gone so far as to print a screenshot of the data structures from my DBMS so I wouldn't miss any fields and I would get the validate and size settings correct. These redundant processes are repeated throughout the entire application development process. Wouldn't it be nice if all the tedium of building apps could be taken care of automatically and we would only have to deal with the complicated parts of our applications?

All great ideas begin with inspiration. My inspiration was Ben Forta. At the ColdFusion Southwest Mini Conference in April 2001, Mr. Forta and I discussed using SQL stored procedures throughout my applications rather than normal queries. He mused about writing applications without any <CFQUERY> tags, using <CFSTOREDPROC> instead. He then dismissed the notion somewhat because it's impractical to deal with the more complex syntax of <CFSTOREDPROC> for all the

queries associated with the average CF application. I wanted to make it practical enough so I could take advantage of all the benefits of using stored procedures (speed, security, etc.). I realized that all I needed to know to automate the creation of the stored procedures was the name of each column and its data type and size. That information is available through a SQL query, isn't it?

Manufacturing the Prefab Parts

With the information available in the sysobjects, syscolumns, and systypes tables in my MS-SQL Server, I set about creating a ColdFusion application factory. I selected Java as my language of choice to create this factory (see Figure 1). I prefer the state-aware nature of local applications to the state-unaware nature of Web apps. Also I thought it would be more useful for developers who don't run their own servers and may not have the ability to use the <CFFILE ACTION="WRITE">, which would be required to make the factory work. Not to mention that it will run on both Unix and Windows machines. Java is so cool!

The parts my factory manufactures are:
- **SQL**
  –Stored procedures (see Listing 1)
  –Triggers for returning the @@identity associated with INSERTs (see Listing 2)
- **ColdFusion**
  –Custom tags that call each stored procedure (see Listing 3)
  –Custom tags that call data entry screen widgets (see Listing 4)
  –Custom tags that call the data entry action page widgets (see Listing 5)
  –Custom tags that call Combo and List Box widgets (see Listing 6)

FIGURE 2: Custom tag snippets

- **ColdFusion Studio**
  –Snippets for all the custom tags created above so they can be easily entered into applications (see Figure 2)

Use of this factory would certainly not be limited to the templates that I created. I've designed a programming interface for the factory so you can design your own templates. The syntax for this template-generation interface is ColdFusion-friendly using tags like <query>, <loop>, and <if> (see Figure 3).

You may download a copy of my ColdFusion Factory from www.cf factory.com/.

Building the Application

Building the application from the prefab parts was a bit disconcerting the first time I tried. I felt quite strange without all that repetition while building all the screens for the application. Using my system, I was merely calling a series of custom
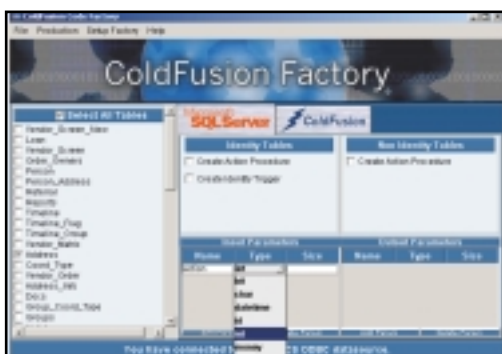
FIGURE 1: ColdFusion factory
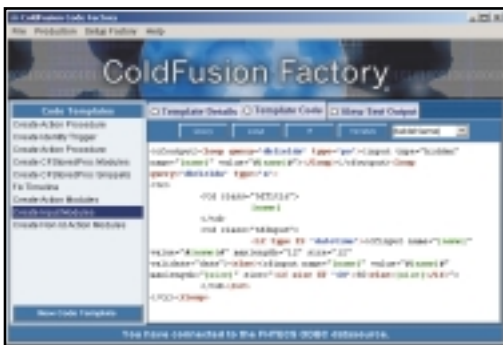
FIGURE 3: Code templates

tags to build the screens (see Listing 7). The action pages are even easier to make, though the same process applies (see Listing 8).

You could also easily use the factory to make all the fuses associated with data entry screens and queries within the framework of a Fusebox application.

Benefits of the Code Factory

Speed. Accuracy. I first developed this process and software in response to a request from a customer to write a mission-critical piece of software. When I asked about the time frame, the response was, "I need it in five weeks. Can you do it?"

Not wanting to turn away the business, I gave him the thumbs-up and quickly took my leave to start investigating the requirements of the new system. Once I had a general idea of the required data to store, I built the data structures. This is when I realized I was in trouble. Printing out the entity-relationship diagram showed me that my tables fit snuggly on six pages. I would still be doing busy work on the application in six weeks rather than installing it. I needed to do something fast.

I remembered my conversation with Mr. Forta about stored procedures and that I planned on writing a little something to deal with that problem. I realized it could be extended to deal with input modules as well as stored procedures. I spent a couple of sleepless days and nights working on my factory. When it was finished, I wasn't sure yet if it would help. I hit the "Go" button and waited breathlessly to see if it would work. It did. About two sec-

onds later I was ahead of my prior level of project completion by about five weeks. That left me time to deal with the more complicated work of writing the business logic, etc.

Flexibility. Then disaster struck. The customer had neglected to inform me of some rather important information that would change the data structures dramatically. After I my heart attack, I put these changes into the context of my new tool and realized that this might be only a two-second fix. Once again I held my breath and watched as more miracles happened. All those templates that would need to be rewritten, double-checked, and repaired, worked the first time. It was mind-boggling.

How to Make Your Own Factory

What's absolutely required to make this work is:
1. A completed database design with the structures built in your database server; this should be completed before you start any coding, even if you use more conventional development strategies.
2. A database server that offers you access to sysobjects and syscolumn style data; MS-SQL Server, Oracle, and Sybase are all good choices.

The key is to understand the sys* tables in your database server. I use MS-SQL Server; see Listing 9 for information about that product.

As you may have noticed, I've limited the sysobjects table to return only "user" tables, not the ColdFusion client-variable tables. The information afforded by these two simple queries allows you to write code to automate the creation of all the types of files I discussed.

The details of how I wrote my ColdFusion Factory are beyond the scope of this article, but all the information you need to follow in my footsteps is in your database right now.

JOSH@CFFACTORY.COM

Listing 1

```
CREATE PROCEDURE ShippersModify
  @ShipperID int ,
  @CompanyName nvarchar ,
  @Phone nvarchar ,
  @action int

AS

IF (@action = 0)
  BEGIN
    INSERT INTO Shippers (
    CompanyName ,
    Phone )
    VALUES (
    @CompanyName ,
    @Phone )
  END
ELSE
  BEGIN
    IF (@action = 1)
      BEGIN
        UPDATE Shippers
        SET
          CompanyName = @CompanyName ,
          Phone = @Phone
        WHERE
          ShipperID = @ShipperID
      END
    ELSE
      BEGIN
        IF (@action = -1)
          BEGIN
            DELETE FROM Shippers
            WHERE
              ShipperID = @ShipperID
          END
        ELSE
          BEGIN
            RAISERROR (50101,10,@action)
          END
  END
END
```

Listing 2

```
CREATE TRIGGER OrdersGetID ON dbo.Orders
FOR INSERT AS SELECT @@IDENTITY AS OrdersId
```

Listing 3

```
<cftry>
  <cfparam name="Attributes.ShipperID">
  <cfparam name="Attributes.CompanyName">
  <cfparam name="Attributes.Phone">
  <cfparam name="Attributes.action">
  <cfstoredproc procedure="ShippersModify">
    <cfprocparam type="in" dbvarname="@ShipperID"
      cfsqltype="CF_SQL_IDSTAMP"
      value="#Attributes.ShipperID#">
 ***Other Parameters***
    <cfprocresult name="caller.SPShippersModify">
  </cfstoredproc>
<cfcatch type="Any">
  ***Catch Code***
</cfcatch>
</cftry>
```

Listing 4

```
<cfoutput>
<input type="hidden" name="ShipperID"
value="#ShipperID#">
</cfoutput>
  <tr>
    <td class="tdTitle">
      CompanyName
    </td>
    <td class="tdInput">
      <cfinput type="text" name="CompanyName"
      maxlength="80" value="#CompanyName#" size="60">
    </td>
  </tr>
  <tr>
    <td class="tdTitle">
      Phone
    </td>
    <td class="tdInput">
      <cfinput type="text" name="Phone"
      maxlength="48" value="#Phone#" size="48">
    </td>
  </tr>
```

Listing 5

```
<cfif form.ShippersId is "">
  <cfset action = Application.spInsert>
<cfelse>
  <cfset action = Application.spUpdate>
</cfif>


<cfmodule template="/sp/ShippersModify.cfm"
ShipperID = "#form.ShipperID#"
CompanyName = "#form.CompanyName#"
Phone = "#form.Phone#" action = "#action#">


<cfif form.ShippersId is "">
  <cfset form.ShippersId =
  SPShippersModify.ShippersId >
</cfif>
```

Listing 6

```
<cfmodule template="/sp/businessCombo.cfm">
  <tr>
    <td class="tdTitle">
      businessId
    </td>
    <td class="tdInput">
      <cfselect name="businessId" value="value"
      display="display" query="SPbusinessCombo"
      selected="#businessId#"></cfselect>
    </td>
  </tr>
```

```
<cfmodule template="/sp/SPbusinessDetailsSelect.cfm"
businessId="0">
<cfform action="insBusiness_action.cfm">
 <table class="parenttable">
   <tr>
   <td>
   <table width="100%">
   <cfoutput query="SPbusinessDetailsSelect"
   group="businessId">
    <cfmodule template="/sp/businessTypeList.cfm">
    <cfinclude template="/im/businessTypeList.cfm">
    <cfinclude template="/im/businessInput.cfm">
    <cfinclude template="/im/addressInput.cfm">
   </cfoutput>
   </table>
   </td>
 </tr>
 </table>
</cfform>
```

Listing 8

```
<cfinclude template="/im/addressInputAction.cfm">
<cfinclude template="/im/businessInputAction.cfm">
<cfinclude template=
 "/im/business_businessTypeInputAction.cfm">
```

Listing 9

```
SELECT id , name FROM sysObjects WHERE type = 'U' AND name
not in ('dtproperties','CDATA','CGLOBAL')

id          name
----------- -------
885578193   message
133575514   users

SELECT sysColumns.name as colName , sysColumns.length as
colLength , sysColumns.colstat as colStat , sysTypes.name
as coltype FROM sysColumns , sysTypes WHERE
sysColumns.xtype = sysTypes.xtype AND sysTypes.name <>
'sysname' AND sysColumns.id = @theid ORDER BY colorder

colName     colLength colStat coltype
----------- --------- ------- -------
messageId   4         1       int
userId      4         0       int
message     2000      0       varchar
```

CODE LISTING
▸▸▸▸▸▸▸▸▸▸▸▸▸

The code listing for
this article is also located at
www.sys-con.com/coldfusion/sourcec.cfm

FuseTalk 3.0 from **e-Zone Media**

REVIEWED BY
ERON COHEN

One of the most common database-driven applications on the Internet is the discussion forum.

Internet forums are used for product support, organizational announcements, and corporate discussion areas, and as a place to talk about the theme of a given Web site. They're popular because they offer a convenient way to discuss interests and ideas while keeping a sense of history – something a typical e-mail list doesn't have. They also have the advantage of offering site administrators a high level of control over what's being talked about and how it's discussed.

If you've been considering adding a discussion forum to your Web site, you know there are many competitors. However, one of the best forums applications I've ever used is ColdFusion-based FuseTalk 3.0 from e-Zone Media. FuseTalk is excellent – it's inexpensive, feature-rich, easy to deploy, and a pleasure to maintain.

Installation

It took me about 20 minutes to install and configure FuseTalk. The installation comes as an executable .zip file. You get started by unzipping the software to an appropriate directory on your server and then setting up the preferences, such as the data source. I used ColdFusion 5 with SQL Server 2000 for the database back end. FuseTalk also supports MS Access, Oracle, and MySQL; however, if you use SQL Server, you'll have the option of using the full-text search feature.

After you run the Cold-Fusion-based DHTML installer, simply edit some ini files to finish the initial setup. After that you can use the administrator

menus to get your forums going. You can also customize the look and feel of your forums by editing the header and footer files provided in the "\includes\Custom" directory. This is a great place to add your own logos or advertising.

FuseTalk is a secure application that generally requires users to have some kind of account. There are several ways to handle this. The easiest, of course, is the default account management that's built in, but documentation is provided that describes how to use your current site accounts or Windows NT authentication.

Once you have the product installed, you'll want to familiarize yourself with the administrative options.

The e-Zone Media people have really thought of everything. For instance, the moderation and censoring features are very advanced. Moderators can be set up on a per-category basis, and there are levels of moderators that have global privileges. Furthermore, all moderator actions are automatically logged. Administrators can choose which words they want to censor by adding them to a table using the

admin screens. If all else fails, ill-behaved users can be banned by IP address, e-mail address, or username.

There are also a few bonus features that put this application over the top. For instance, the software supports clustering. You can even provide a clustering URL in the administrator so the forum cache can be cleared on the clustered server. There's also a chat server in which users can create their own chat rooms, which can be deleted automatically after a set period of inactivity. They even have a "buddy list" feature that lets you know when your friends are logged in to the forums.

Another terrific feature is the ability to post messages to the forums via e-mail. A reverse feature complements this one: users can choose to have all messages posted to the forum automatically e-mailed to them. There's even a survey engine (again, they've thought of everything). But if you feel the forums are missing something, don't worry: this version of FuseTalk takes plug-ins. For instance, e-Zone

FuseTalk 3.0 has many administrative options

e-Zones own forum features FuseTalk

Media has one available for purchase that allows businesses to use forums for enhanced online support. Others are in the works.

The User Interface

As mentioned before, the user interface is sleek and well designed. Like the administrators, users will be pleased with the features. For instance, individual users can customize the environment colors and see only the categories of messages that interest them. Fun features like "author icons" let each author choose or upload an avatar to convey his or her personality, and insert "emotions" to convey a mood for a particular message posting. Users will also enjoy the ranking feature of FuseTalk – the more they post, the higher rank and credibility they obtain.

The only thing that I feel could be improved, on both the user and administrative sides, is the online help and documentation. It's not bad, but it could be better. To the credit of the FuseTalk developers, I would dub most of the application "self-explanatory"; I configured most of the application without ever needing the docs.

Pricing

Pricing for FuseTalk is reasonable. If you need only one forum and don't mind encrypted source code, $249 gets you going with the Professional version. If you need to have more than one forum, you'll want to look into the Enterprise version; this allows you 25 forums on one server for $999. As you'd expect, it gets more expensive if you want the unencrypted source code. The Professional version is $369, and the Enterprise version is $1,999. The code is well written and neatly formatted, although somewhat underdocumented, so if you do decide to go with the unencrypted version, you'll be glad to know you won't have to dig through spaghetti code to make your modifications. Demo versions are available for download from the e-Zone Media Web site. You may have already seen this product from a user's perspective – Macromedia uses it to power the ColdFusion forums on their Web site.

ABOUT THE
AUTHOR
Eron Cohen, a ColdFusion developer in Silver Spring, Maryland, has worked with ColdFusion since 1996.

ERON.COHEN@YAHOO.COM

ColdFusion and **Section 508**

BY
MICHAEL SMITH
AND
JON BRUNDAGE

Web site compliance with disabled access requirements

There's a new federal law, known as Section 508, that requires disabled accessibility to government Web sites.

We'll explain what you have to do to comply, how ColdFusion can help (or hinder), and why 508 isn't a bad thing after all.

Imagine

Imagine you can't see and you have a friend read you the HTML source for a Web page over the phone so you can understand the site. That's what viewing a site with a screen reader is like. Imagine minutes of hearing header HTML code for navigation – with cryptic image filenames and no English description. Imagine complex table layouts with no clue as to what each number means. Imagine applets that you don't know anything about because there's no HTML source for them.

Welcome to the world of blind Web users. Section 508 is aimed at making it easier for them to read your Web pages.

Section 508 Basics

As many programmers for the federal government already know, there are new standards for Web page accessibility. Currently these standards are mandatory for all federally sponsored sites, but who knows when the law might be extended to all sites, as ADA (Americans with Disabilities Act) was extended to all businesses. We suggest you check out disabled accessibility and get the benefits of better Web design now. Dealing with accessibility gets you thinking out of the box. For instance, can your page function without a mouse? Plus you'll reap the benefits of targeting your site to cell phone browsers, as similar issues apply.

Section 508 ensures that the disabled can access the same public information as everyone else. In the last census 54 million Americans had some disability, including 10 million blind or visually impaired. People with disabilities use tools such as screen and Braille readers. Think of their tools as alternative browsers. Just as you must consider differences between Internet Explorer and Netscape, these browsers must be considered when you code.

Section 508 is hot – both Macromedia and Microsoft have Web pages listing their 508 compliance initiatives, and Macromedia has released an extension to Dreamweaver with accessibility tools. Designing for maximum access to Web content is spreading to education, local governments, and corporations.

The Regulations

Some of the requirements are listed in Table 1 (for a full review, visit www.access-board.gov/sec508/guide/1194.22.htm).

ColdFusion presents solutions to help you comply, and also presents some problems for you to overcome.

Help from ColdFusion

How can ColdFusion assist you? Let's look at an example of how to comply with Rule A: "A text equivalent for every nontext element shall be provided…via alt."

If you're displaying a record with an associated image (perhaps the description of a house with a photo), list the image path and

RULE	
A	A text equivalent must be supplied for every image.
B	Alternatives for multimedia presentations must be synchronized with the presentation.
C	All information conveyed with color (e.g., red for a negative number) must be provided in an alternative form.
D	Documents must be readable without an associated style sheet.
E	Redundant text links must be provided for each active region of a server-side image map.
F	Client-side image maps should be provided instead of server-side image maps.
G, H	Data tables must associate row and column headers with the data contained in them.
I	Frames must have readable names.
J	Avoid causing screen flicker.
K	If there's no way to comply with 508 standards, provide a text-only page, and update it whenever the primary page changes.
L	Any information provided by scripting languages must also be supplied in text readable with assistive technology.
M	If an applet, plug-in, or other application must be present on the client, provide a link to a compliant plug-in or applet .
N	Online forms must allow assistive technology to access elements required for completion and submission.
O	Users must be able to skip repetitive navigation links.
P	When a timed response is required, the user must be alerted and given enough time to indicate that more time is required.

TABLE 1 Our take on some 508 rules

> " Section 508 is hot – both Macromedia and Microsoft have Web pages listing their 508 compliance initiatives, and Macromedia has released an extension to Dreamweaver with accessibility tools "

image description as fields in the record, and then output the image as part of the dynamic data (see Listing 1).

Data Tables

A common ColdFusion programming technique is to build dynamic tables from record sets. Good news! You can build dynamic tables that are Section 508–compliant and populated with records from a CFQUERY using the ID and HEADER tags. Say you wish to query a data source containing information about your CD collec-tion, with the records displayed in a table listing artist and CD title (see Listing 2). A screen reader would speak the following: "CD Collection. This table lists the artists and titles of my music collection. Artist, Title…" and then would speak the contents of each cell.

Things to Watch Out For

ColdFusion is an excellent tool for creating compliant Web pages. However, there are some things you should consider. CFFORM, CFIN-PUT, CFSELECT, and other Java-Script-producing tags can't control

SUGGESTION	
A	All images, except shims, need an ALT tag.
B	Provide a link to a text version, synchronizing captions with audio.
C	Use a minus sign or other text information too.
D	Content must be readable if the style sheet is removed.
E	Use redundant links because readers can't see URLs on server-side image maps.
F	Client is better than server, as readers can see the URLs before clicking. Use the ALT tag on the areas.
G, H	See Listing 2 and visit www.w3.org/TR/WCAG10-HTML-TECHS/.
I	Identify frames with plain English names when used, but it's best to avoid their use.
J	Don't loop your animation or use <BLINK> tag.
K	This is easy if the text is in a database.
L	Avoid pop-up windows and JavaScript alerts. When using JavaScript for DHTML, use the <NOSCRIPT> tag just after a closing </SCRIPT> to provide a text description of the action.
M	Avoid using noncompliant applets or plug-ins. The new PDF and Flash players will be compliant.
N	Provide hotkey access to form elements and make a TABINDEX order for the form fields. Use LABEL tag.
O	Place a "jump to content" link just after the <BODY> tag that jumps to an anchor placed at the beginning of the page's content.
P	Roll your own session-timeout code to allow longer timeout for disabled users.

For the complete text, visit www.access-board.gov/sec508/guide/1194.22.htm.

ABOUT THE AUTHORS

Jon Brundage is a government consultant, ColdFusion programmer, and an MDCFUG speaker and author.

Michael Smith is president of TeraTech (www.teratech.com/), a Rockville, MD–based consulting company specializing in ColdFusion, database, and Visual Basic development. Michael runs the MDCFUG and recently organized the CF2001 conference.

> ❝ By adapting coding practices required by the guidelines for accessibilty, you'll be in a win-win situation ❞

placement of the <NOSCRIPT> tag. You'll need to browse your pages and look at the source code produced to determine placement of this tag. (Macromedia is aware of this issue and will address it in the future.) Another consideration is the use of CFINCLUDE. If the included template is used solely as an inserted page component, strip the file of all <HTML>, <HEAD>, and <TITLE> opening and closing tags. Otherwise, pages may confuse screen readers with extra opening and closing tags.

Other problem tags that use Java applets include:
- CFGRID
- CFTREE
- CFSLIDER

To be compliant you'll need to provide the data in these tags in a separate format such as HTML tables or lists. For example, for CFGRID you can provide a table of input boxes with names made unique by adding the row number to them (see Listing 3).

In the action page we can again loop over the rows, ignoring any blank ones (see Listing 4).

Timeout

Do you have members-only login sections on your site? Then you may have timeout issues. Session variables, by default, last only 20 minutes. That may not be long enough for disabled users to view the material. We suggest that you code the timeout and increase the session timeout to an hour or more. Then you can let individual users increase the timeout for themselves.

Positive Effects of Compliance

By adapting coding practices required by the guidelines for accessibility, you'll be in a win-win situation. First, your disabled visitors are winners because they'll be able to access your site. Second, your code will be streamlined and clean, so it will download fast on dial-up connections and will be easier to use from PDAs too.

Summary

In this article we've introduced you to accessibility issues and new Section 508 requirements for federal Web sites and showed how Cold-Fusion's flexibility makes it your ally for compliance. An accessible Web site is a well-designed site. ▨

Resources
- www.w3.org W3
- *Popular Bobby checker for examining pages for accessibility:* www.cast.org/bobby
- *Federal Section 508 – a ColdFusion site:* www.section508.gov/
- *Henter-Joyce, makers of the JAWS screen reader:* www.hj.com
- *HiSoftware, producers of verification tools:* www.hisoftware.com
- *W3 accessibility guidelines:* www.w3.org/TR/WCAG10-TECHS
- *DIBOC, an organization with some excellent examples:* www.diboc.com/wrac.cfm
- *Tools download page:* www.macromedia.com/macromedia/accessibility/

@ JONWIND@WINDCOMPANY.COM

@ MICHAEL@TERATECH.COM

Listing 1

```coldfusion
<cfquery name="house_data" datasource="houses">
SELECT price, address, image_path, image_description
FROM   tblHouses
</cfquery>


<CFOUTPUT QUERY="house_data">
  Address: #address#    Price:  #price#
<img src="#image_path#" alt="#image_description#">
</CFOUTPUT>
```

Listing 2

```coldfusion
<table summary="This table lists the artists and titles of
my music collection" border="1" >
 <caption>C D collection</caption>
  <tr>
    <th id="artist">artist</th>
    <th id="title">title</th>
  </tr>
  <CFLOOP QUERY="CD_collection" >
   <CFOUTPUT>
<tr>
    <td headers="artist">#artist#</td>
    <td headers="title">#title#</td>
</tr>
   </CFOUTPUT>
  </CFLOOP>
</table>
```

Listing 3

```coldfusion
<cfset numrows = 5>
<cfoutput>
<input type="hidden" name="numrows" value="#numrows#">
</cfoutput>
<TABLE>
<TR><td>Series</td><td>entity</td><td>start_date</td></tr>
<CFLOOP index="row" from="1" to="form.
<cfoutput>
<TR>
<td><CFINPUT TYPE="text" Name="Series_#row#"></td>
<td><CFINPUT TYPE="text" Name="entity_#row#"></td>
<td><cfinput type="Text" name="start_date_#row#"></td>
</tr>
</cfoutput>
</cfloop>
</table>
<input type="submit" name="submit" value="Submit Request">
```

Listing 4

```coldfusion
<CFLOOP INDEX = "Counter" FROM = "1" TO = Form.numrows>
 <CFIF
  evaluate("FORM.SERIES_#COUNTER#") IS NOT "" AND
  evaluate("FORM.ENTITY_#COUNTER#") IS NOT "" AND
  evaluate("FORM.START_DATE_#COUNTER#") IS NOT "" >

  Update or Insert record here
 </cfif>
</cfloop>
```

CODE LISTING ▶▶▶▶▶▶▶▶▶▶▶▶▶

The code listing for this article is also located at
www.sys-con.com/coldfusion/sourcec.cfm

Ask the **Training Staff**

A source for your CF-related questions

BY
**BRUCE
VAN HORN**

Sometimes I take for granted just how cool ColdFusion really is! The other day one of my programmers asked me how to do what I thought was a fairly easy task.

When I showed him the four lines of CF code he needed, he said, "ColdFusion rocks! It would have taken me 50 lines in PHP to do the same thing!" What could I say, other than "Yep"? Below are a few lines of code that I hope will make your life easier.

Q: *How can we prevent a browser from caching the content of our Web pages? For example, our content is constantly changing; if a user browses a page, goes on to other pages, and then uses the back button to return to a previous page, we don't want the user seeing the same content he or she saw the first time – we want the user to see the updated content on that page. Can we do this?*

A: Absolutely! Most browsers do cache page content to enhance the performance of site navigation. However, you can put some code into your pages that will instruct the browser not to cache their content. Unfortunately, different browsers require different no-cache parameters, so I use three tags to make sure it works in all browser versions. Place the following code at the top of each page you don't want cached:

```
<CFHEADER NAME="Expires"
  VALUE="#Now()#">
<CFHEADER NAME="Pragma"
  VALUE="no-cache">
<CFHEADER NAME="cache-control"
  VALUE="no-cache, no-store,
  must-revalidate">
```

Q: *Our application needs to send out an e-mail message on behalf of our users; however, our mail server doesn't allow relaying from e-mail addresses outside our domain. We want the message to come from one of our e-mail accounts, but have a reply-to set to the address of the user who sent the message. I don't see any way for CFMAIL to set a "Reply-To" attribute. Can this be done?*

A: This is a great question! Yes, it can be done now that CFMAIL can use the CFMAILPARAM tag. You may be aware that you can use the CF-MAILPARAM tag to send multiple file attachments, but you can also use it to send mail header information (like "Reply-To"). Here's all you have to do:

```
<cfmail to="paul@domain.com"
        from="noreply@abcgreet
          ings.com"
        subject="You have
          received a Birthday
          Greeting!">
<cfmailparam name="Reply-To"
  value="yourfriend@somedomain.com">
Message body goes here...

</cfmail>
```

Q: *Is there a way to program ColdFusion to restart after so many database connection errors? And, more generally, is there a way of programmatically restarting ColdFusion?*

A: Yes and no. Yes in the CF Administrator – you can have CF restart after a certain number of failed requests (page or DB requests). You can get it to restart by checking the "Timeout requests after ___ seconds" box in the Settings section (be sure to set a timeout in seconds – do not leave it set to 0). Also check the "Restart at ___ unresponsive requests" box and plug in a number. Don't set either of these settings too low or CF will be restarting all the time.

There is no CF tag or function to restart the CF service, but you can write a .bat file to issue an NT command to stop or start the service:

```
net stop "World Wide Web
  Publishing Service"
net stop "Cold Fusion
  Executive"
net stop "Cold Fusion RDS"
net stop "Cold Fusion
  Application Server"
net start "Cold Fusion
  Application Server"
net start "Cold Fusion RDS"
net start "Cold Fusion Executive"
net start "World Wide Web
  Publishing Service"
```

and then call that .bat file using CFEXECUTE:

```
<!--- build a form that has a
submit button on it and points
to this action page --->
<cfif isDefined("Form.restart")>
 <cfexecute
  NAME="C:\websrvr\htdocs\restart
  svcs.bat"/>
 All webservices (IIS & CF)
  will be restarted. Please
  check back in 30 seconds.
</cfif>
```

• • •

Please send your questions about ColdFusion (CFML, CF Server, or CF Studio) to AskCFDJ@sys-con.com. Visit our archive site at www.netsite dynamics.com/AskCFDJ.

@ BRUCE@NETSITEDYNAMICS.COM

ABOUT THE
AUTHOR
Bruce Van Horn is president of Netsite Dynamics, LLC, a Macromedia certified instructor, and a member of the CFDJ International Advisory Board.

Letters to the Editor. . .

User-Defined Functions

I was reading the October issue of **CFDJ** (Vol 3, issue 10) and came across Bruce Van Horn's **Q&A** column. One question asked how to get the file size, and he said it would be great to have a UDF for this. Well, the following link has the very UDF he proposed: www.cflib.org/library. cfm?ID=8.

BTW, his column is one of the first I read when I get my copy of *CFDJ*.

Robert Barrett
rbarre@acxiom.com

Thanks. I found it too just a few days after I wrote the column!

Bruce Van Horn
bruce@netsitedynamics.com

Relative Paths Can Be Problematic

I loved the article, "A Cure for Arachnophobia," (Vol. 4, issue 1), but ran into a challenge with my images. The pictures work fine in the standard pages but (I'm guessing here) since the URL has changed, it can no longer find the pictures. Is there any way to make the <img> tag recognize the URL only up to the <file>.cfm part?

Chris Most
via e-mail

This is a valid concern, and one that may or may not be an issue depending on the Web server being used. Yours is one of several messages I received asking the same question. Relative paths can be very problematic if you tinker with URLs like I did in the column.

One solution (and I know it's not an easy one to retrofit into an existing site) is to build URLs using a cf variable as a prefix, like this:

```
<IMG
SRC="#prefix#../../file.jpg">
```

and just set that prefix as needed, starting with a /. This gives you flexibility so you don't have to hard-code all paths (you hard-code only the one variable) and the ability to play tricks with URLs, too.

Another option is to use absolute paths instead of relative ones, and to build them on the fly (so they're still relative). This is actually not as complex as it sounds; you just need to determine the current file name (there are CGI variables and functions that can help with this) and then programmatically build the real path, which can then be embedded in the <IMG> tags. You could do this just once per page (or even in a custom tag) and then refer to that path in the <IMG> tag.

Ben Forta
ben@forta.com

A Road Fraught with Danger!

Before I move on to my next project I want to say thanks for Norman Elton's article "Excel's Web Query" (Vol. 3, issue 12). Just like him, I find that the users who log into our ASP are much more comfortable with reports in Excel. So I started down the Microsoft road, using Office as a COM object to create spreadsheets, but that's a road fraught with many dangers, including having to restart the Office server if a macro goes bad!

Thanks and keep up the good work!

Dave
via e-mail

I enjoyed the article, "Adding New Help Topics to Studio" by Charlie Arehart (Vol. 4, issue 1), on adding help to ColdFusion Studio.

One source for CF Studio-ready documentation is the OpenWDDX SDK. The entire developer kit documentation can be integrated into Studio. Download it from www.open-wddx.org/.

Michael Mongeau
mmongeau@yahoo.com

Thanks, Michael, for pointing this out to me. For those interested, you'll find the instructions for this in the Help files of the SDK itself, in the first chapter. Here's the key point:

"Just unpack the SDK .zip archive into Studio/HomeSite's Help directory, then click the Help icon in the Resource Tab. A new 'book' labeled Wddx SDK will appear. (You may have to hit F5 once to refresh the display.)"

Charlie Arehart
carehart@systemanage.com

2001

READERS' CHOICE
A W A R D S

The ColdFusion Readers' Choice Awards Acknowledge the Pursuit of Excellence in the CF Software Industry

The Best of the Year Are Here!

Once again ***ColdFusion Developer's Journal*** is delighted to announce its ColdFusion Readers' Choice Awards. For the second year in a row we're proud to be the only publication to present such awards. CF developers and their companies are finally being acknowledged for their product design and excellence. It's recognition that we at *CFDJ* feel is long overdue and well deserved.

How We Did It…

Our readers nominated products in 15 categories; we then put them up for vote on the *CFDJ* Web site. The response was immediate and vocal. No one ever said that the CF professional was shy or without an opinion! On the contrary, our savvy readers displayed once again that they keep their ears to the ground when it comes to trends and leading-edge product developments. They keep us on our toes, too, informing us about what they feel truly merits the ***CFDJ*** seal of approval. These awards are merely an extension of ongoing dialogue. The stuff either works or doesn't. Period.

Free of Outside Influences

It's out in the field that a product earns its stripes. That's why we created the ColdFusion Readers' Choice Awards. You know that when a product and/or company displays the award, they have passed a rigorous and painstaking inspection of their product or service. Free of outside influences, the awards are a true reflection of what ColdFusion developers are using.

Congratulations!

Congratulations to all the category winners and finalists! Many thanks to the readers of *ColdFusion Developer's Journal* who made these awards possible by casting their vote on our Web site during the past few months. These awards are the only ColdFusion awards in the industry, and they are given to products nominated and selected by the largest number of professional ColdFusion developers ever assembled for such a purpose.

Once again, congratulations! **SYS-CON Media** wishes each of these companies and its employees continued excellence in the software industry. Following are the products that you, our readers, selected as our second annual ColdFusion winners.

— 2nd Runner-up —
ERWIN
from Computer Associates

ERwin is a data modeling solution that creates and maintains databases, data warehouses, and enterprise data resource models. ERwin models visualize data structures, which helps organize, manage, and even mitigate the complexities of data, database technologies, and the deployment environment.

www.ca.com

BEST **Design Service**
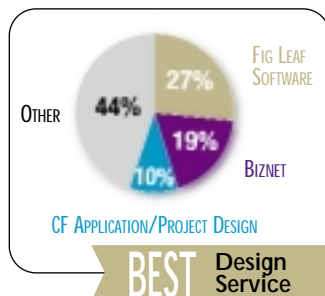
— Winner —
FIG LEAF SOFTWARE

Fig Leaf's creative media department has designed and developed Web-based media for some of the most recognized companies and organizations in the country. Comprised of a team of award-winning artists, programmers, writers, and instructional technologists, the staff strives to create ingenious and innovative solutions. Their graphic Web site design, interactive media design, corporate branding, and usability consulting services have helped build successful online solutions.

www.figleaf.com

— 1st Runner-up —
BIZNET

Biznet's award-winning team of graphic designers seeks to provide a creative working environment in which to foster talent and attract the best professionals across all areas. Biznet hopes to have all their designers and developers Macromedia certified by the end of the year.

www.biznet-solutions.com

— 2nd Runner-up —
CF APPLICATION/ PROJECT DESIGN
from Web Panache

Web Panache's team is trained in the latest techniques and keeps abreast of the changing requirements in this field. All of Web Panache's partners have extensive backgrounds in computer science and years of personal experience supporting people and projects of all kinds. *Panache* describes the "out of the ordinary service" they extend to every customer.

www.webpanache.com

BEST **E-Business Software**
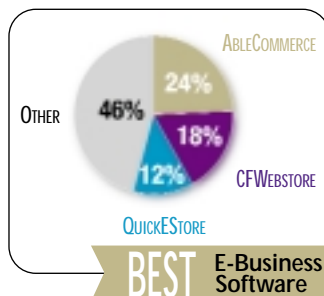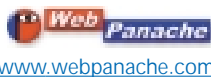
— Winner —
ABLECOMMERCE
from AbleCommerce Software

AbleCommerce enables users to generate secure online storefronts with a robust feature-set that includes international currencies, bundled product "Kits," automated shipping, and multitier taxation. Additional capabilities include cross-selling, membership pricing, and the ability to create targeted discounts. The source code version offers developers scalability, adaptability, and the flexibility to meet future needs.

www.ablecommerce.com

— 1st Runner-up —
CFWEBSTORE
from Dogpatch Software

CFWebstore is a completely integrated Web storefront and e-commerce solution. CFWebstore allows you to customize the templates and utilize the built-in store settings to create your own custom store.

It's designed for small- to medium-size merchants and Web developers who want plenty of features in a product they can easily customize as well. Features include a wide range of discounts such as coupons and gift certificates, inventory management and cross-selling, an easy-to-use Web-based administrator, built-in help files, unlimited number and level of categories and products, custom product fields, and special quick-edit screens for products and categories.

www.cfwebstore.com

— 2nd Runner-up —
QUICKESTORE
from QuickEStore E-Commerce Software

QuickEStore is versatile, easy-to-use, and will have you doing business on the Internet in no time. QuickEStore can be installed in minutes and includes four "one-click" storefront wizards, inventory management and reporting, flexible sales tax and shipping options, and a comprehensive, browser-based admin console that makes managing your store simple and secure. Optional integration with popular online credit card processors is built right in. It also includes customer relationship management tools and QuickEChat.

www.quickestore.com

BEST **Education & Training**
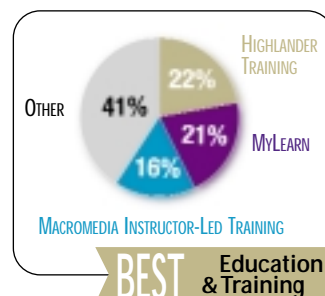
— Winner —
HIGHLANDER TRAINING
from Highlander PLC

Highlander Training, a training center for e-business solutions, has won a number of awards for both its trainers and courses, and has trained some of the world's most successful companies in e-

business technology. Highlander runs a range of Macromedia authorized courses including programs for Web site developer, Java Web developer, ColdFusion developer, and ColdFusion administrator. There are also courses in Macromedia Spectra for Developers, Macromedia Spectra Methodology, and Professional XML Training.

www.highlander.co.uk

— 1st Runner-up —
MYLEARN
from Documentum

myLearn is a full-featured learning management system (LMS) used to provide personalized training programs to Documentum's customers, partners, and employees. myLearn was live after two days of development and mission-critical after 30. myLearn has improved Documentum's education information infrastructure and introduced thousands to e-learning.

http://mylearn.documentum.com

— 2nd Runner-up —
MACROMEDIA INSTRUCTOR-LED TRAINING
from Fig Leaf Software

Fig Leaf's certified instructors leverage years of development expertise to provide insights into your Web challenges. Our classes are held in Washington, DC, Atlanta, Chicago, Baltimore, and northern Virginia, as well as client sites. Fig Leaf training developed the Fast Track to JavaScript, Fast Track to SQL, and Spectra for Developers courseware for Macromedia, as well as authored the book *Inside Flash,* available from New Riders. Our new "RealWorld CF" seminars cover topical issues related to Web development, including securing ColdFusion applications, printing, reporting, graphing, and Web design strategies.

http://training.figleaf.com
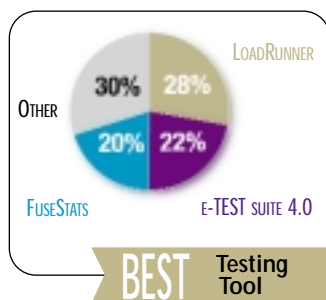
BEST Testing Tool

— Winner —
LOADRUNNER
from Mercury Interactive

LoadRunner is the industry-standard load-testing tool used to predict system behavior and performance. To identify and isolate problems, it exercises an entire infrastructure by emulating thousands of users. LoadRunner's integrated real-time monitors enable organizations to minimize test cycles, optimize performance, and accelerate deployment.

MERCURY INTERACTIVE

www.mercuryinteractive.com

— 1st Runner-up —
E-TEST SUITE 4.0
from Empirix

Empirix e-TEST suite 4.0 is an integrated test solution optimized for Web applications. e-Test suite includes e-Tester for functional/regression testing, e-Load for load/scalability testing, and e-Monitor for 24/7 quality monitoring of deployed applications.

All three tools are powered by a common set of Visual Scripts and don't require any programming. There are no proprietary languages to learn, no special proxies to set up, and no training classes required. As your application changes, any differences in your tests are highlighted in the Visual Scripts and can be automatically updated in place, so your regression, load, and monitoring tests will stay synchronized with your application.

empirix

www.empirix.com

— 2nd Runner-up —
FUSESTATS
from FuseLogic

FuseLogic offers cutting-edge solutions for ColdFusion developers. FuseStats, a set of ColdFusion custom tags, allows developers to integrate Web traffic logging and reporting capabilities into Web applications. Versions are available for both generic and fusebox-based applications.

fuselogic

www.fuselogic.com

BEST Web Development Tool
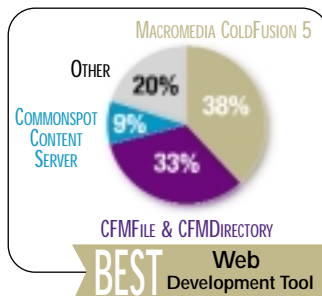
— Winner —
MACROMEDIA COLDFUSION 5
from Macromedia

Macromedia ColdFusion 5 is a cross-platform Web application server that delivers increased productivity, simplified management, and enhanced performance to the award-winning ColdFusion environment. This development tool contains a robust development environment for building Web applications. It gives you a suite of integrated tools for quickly and easily developing applications.

www.macromedia.com

— 1st Runner-up —
CFMFILE & CFMDIRECTORY
from CFM-Resources.com

CFMFile and CFMDirectory are the alternatives to ColdFusion's built-in CFFile and CFDirectory. They perform all the same commands and give you the same amount of power – with one difference: better security in a shared environment. CFMFile and CFMDirectory enable you to lock users within their directories, so

you're not required to install and configure "ColdFusion Advanced Server Security." These tags allow you to implement CFFile- and CFDirectory-like actions for your users in a secure way.

CFM-Resources
"Creating A Better ColdFusion Powered Web, One Free Site @ A Time!"

www.cfm-resources.com/cfmfile-directory.cfm

— 2nd Runner-up —
COMMONSPOT CONTENT SERVER
from PaperThin, Inc.

PaperThin's CommonSpot Content Server is a full-featured, browser-based, out-of-the box Web publishing and dynamic content management solution. It allows content producers within an organization to participate in the creation, management, and delivery of Web content.

CommonSpot Content Server can be implemented in weeks, not months, saving time and development expense. Its flexible template and data-driven architecture, along with granular-access controls and a

multilevel approval workflow process, gives content managers control over the integrity and structure of the site, while simplifying overall content management.

CommonSpot

www.paperthin.com

BEST Web Hosting
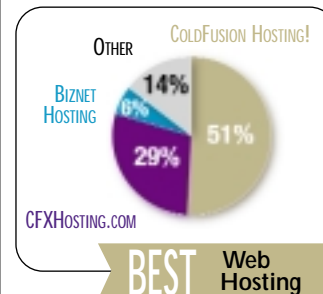
— Winner —
CFM-RESOURCES.COM COLDFUSION HOSTING!
from CFM-Resources.com

CFM-Resources.Com offers both free and affordable ColdFusion hosting solutions. They believe that developers should be able to develop ColdFusion applications in a secure and affordable environment.

EKTRON
www.ektron.com/demo

The company is fully protected by a Cisco 525 Pix firewall, full Internet bandwidth redundancy, power redundancy with a built-in high capacity generator, and a team of ColdFusion developers to ensure you receive the support you need.

www.cfm-resources.com

— 1st Runner-up —
CFXHOSTING.COM
from CFXHosting.com

CFXHosting, founded by Cold-Fusion developers, focuses exclusively on ColdFusion hosting and offers ColdFusion administrator access. In 2000, they were awarded the *ColdFusion Developer's Journal Readers' Choice Award* for excellence in ColdFusion Web hosting.

CFXHosting offers a wide array of plans ranging from shared to dedicated hosting. Their plan features, such as ColdFusion tags and ColdFusion Administrator functions, set them apart from their competition. CFXHosting also provides unlimited POP3 e-mail accounts, WAP (WML), Live Stats, and e-commerce. They use a Cisco-certified data center, which affords clients a high level of reliability, and allows them to offer clients a 99% uptime guarantee.

www.CFXHosting.com

— 2nd Runner-up —
BIZNET HOSTING
from Biznet

Biznet knows that handling online requirements can be a real challenge. Biznet provides clients with a complete range of hosting and colocation solutions that combine state-of-the-art technology with stringent security systems. They also recognize that hosting service levels vary from client to client.

Biznet has a range of packages tailored to suit your specific needs. As a Macromedia Premier Hosting Partner, Biznet Hosting specializes in ColdFusion and offers virtual, dedicated, and colocated hosting.

www.biznet-hosting.com

FUSESTATS 9%
WEBTRENDS REPORTING CENTER 11%
80%

THE COLDFUSION CONTROL PANEL

BEST Web Performance Tool
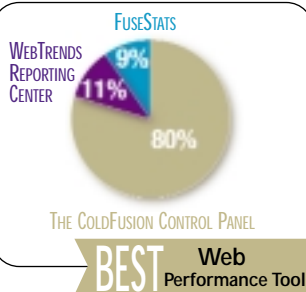
— Winner —
THE COLDFUSION CONTROL PANEL
from CFM-Resources.com

The ColdFusion Control Panel provides ColdFusion hosting companies with a tool that offers customers full control of their Web hosting account and full access to custom tag additions, CFX tag registrations, ColdFusion mappings, and a complete file manager, allowing customers absolute control of their Web site. It also integrates mail management, account information, and much more.

www.cfm-resources.com/cpanellicense.cfm

— 1st Runner-up —
WEBTRENDS REPORTING CENTER
from WebTrends

WebTrends Reporting Center provides analysis of Web site activity to multiple decision makers throughout an organization via a browser-based interface, improving the effectiveness and return on investment across an organization's e-business initiatives. WebTrends Reporting Center is a scalable solution designed to deliver the features and functionality required for customers ranging from mid-market businesses to large enterprises and service providers.

www.webtrends.com

— 2nd Runner-up —
FUSESTATS
from FuseLogic

FuseLogic offers cutting-edge solutions for ColdFusion developers. FuseStats, a set of ColdFusion custom tags, allows developers

---

to easily integrate Web traffic logging and reporting capabilities into Web applications. Versions are available for both generic and fusebox-based applications.

www.fuselogic.com

COMMONSPOT CONTENT SERVER 26%
OTHER 43%
19% **soEDITOR**
12%
EKTRON CMS100

BEST Content Management Tool

— Winner —
COMMONSPOT CONTENT SERVER
from PaperThin, Inc.

PaperThin's CommonSpot Content Server is a full-featured, browser-based, out-of-the box Web publishing and dynamic content management solution. It allows content producers within an organization to participate in the creation, management, and delivery of Web content.

CommonSpot Content Server can be implemented in weeks, not months, saving time and development expense. Its flexible template and data-driven architecture, along with granular-access controls and a multilevel approval workflow process, gives content managers control over the integrity and structure of the site, while simplifying overall content management.

www.paperthin.com

— 1st Runner-up —
SOEDITOR
from SiteObjects, Inc.

SiteObjects soEditor 2.1 (formerly ezEdit) is a ColdFusion custom tag that turns an ordinary <textarea> form field into a dynamic WYSIWYG browser-based HTML editor. soEditor makes it possible for anyone to create, edit, and maintain Web content without the need for HTML knowledge or programming expertise.

www.siteobjects.com

— 2nd Runner-up —
EKTRON CMS100
from Ektron Inc.

Ektron CMS100 is an authoring and publishing application built around eWebEditPro. It was designed to integrate into your existing site in just a few hours, and enable everyone in your organization to author Web content in a word processor-like environment.

CMS100 is a scaled-down version of Ektron's more robust eMPower and Ektron CMS200 applications, offering basic, yet powerful, functionality.

www.ektron.com

BEN FORTA'S COLDFUSION SITE 27%
OTHER 48%
15% **CFDEV.COM**
10%
HOUSE OF FUSION

BEST Web Site
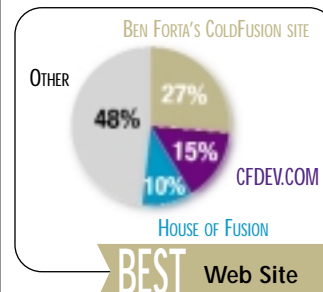
— Winner —
BEN FORTA'S COLDFUSION SITE

www.forta.com is one of the most popular online destinations for ColdFusion enthusiasts (and future enthusiasts). Hosted by Ben Forta, Macromedia's product evangelist and author of the best-selling *ColdFusion Web Application Construction Kit*, this site features a tip-of-the-day, ISP listings, lists of CF-powered sites, presentations, columns, and other valuable resources. Support is also provided for Ben's books, and there are links to other books that will be of interest to CF developers.

www.forta.com

— 1st Runner-up —
CFDEV.COM

cfdev.com, creator of ActivEdit, the award-winning browser-based WYSIWYG HTML Web editor, contains an array of services for the ColdFusion developer. They offer an impressive list of prod-

---

ucts, including the ever-popular ActivEdit, ActivMail, ActivScan, ActivSpell, and DataSource. Visitors to the CFDEV Web site can also choose from a wide range of free ColdFusion tools, such as the CFX XSLT and the CF Alert. Their resources incorporate several different resource centers, references, articles, instructions on software installations, links, and archives. This Web site also includes a support center, press box, job postings, a monthly newsletter, event information, and contact information.

www.cfdev.com

— 2nd Runner-up —
HOUSE OF FUSION

The House of Fusion Web site, run by ColdFusion veteran Michael Dinowitz, is designed to help anyone, no matter what their level or programming background, develop their ColdFusion knowledge. All aspects of the site were created for ease of use and maximum accessibility to important information. The on-site articles teach people the nuances, as well as the basics, of the language. Any information not covered in House of Fusion's own articles can be found in the site's numerous links to other key ColdFusion resources. House of Fusion is probably most famous for being the home of CF-Talk and close to two dozen other ColdFusion-related mailing lists.

www.houseoffusion.com

BEST **Web Application**
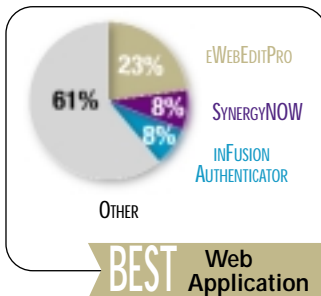
— Winner —
EWEBEDITPRO
from Ektron, Inc.

At the core of the Ektron family of products is eWebEditPro, a word

processor in a Web page – empowering business users to author and maintain Web content. Web developers can integrate eWebEditPro into a database-driven Web site running on any Web application server platform. It's frequently integrated into content management systems (both custom and off-the-shelf), as well as applications for e-learning, discussion forums, HTML mail, and more.

www.ektron.com

— 1st Runner-up —
SYNERGYNOW
from evolutionB Synergy

SynergyNow is a complete solution for developing and hosting run-anywhere applications. It delivers breakthrough value to ASPs by reducing their deployment cost and time-to-market while giving them a large and growing selection of ready-to-rent applications for the wired and wireless Web.


www.evolutionb.com

— 2nd Runner-up —
INFUSION AUTHENTICATOR
from On-Line Data Solutions, Inc.

inFusion Authenticator is an add-on DLL for Microsoft IIS, Gordano NT Mail, and O'Reilly Website and Website Pro. iAuth is an authentication module that allows you to use the power of ColdFusion to authenticate users of your Web site. iAuth templates allow you to protect any area of your Web site from unauthorized use. Since you control the templates, you can authorize on criteria associated with your applications, such as group membership, time of day, and subscription length.

www.coolfusion.com

MOST **Innovative CF Application**
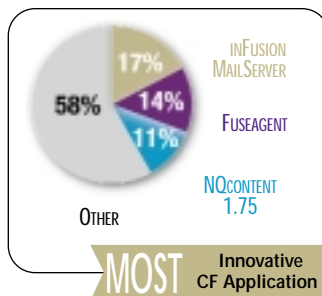
— Winner —
INFUSION MAIL SERVER
from On-Line Data Solutions, Inc.

inFusion Mail Server (iMS) is an e-mail server designed specifically for ColdFusion developers. By utilizing ColdFusion templates to control all mail functions, iMS lets you create customized, dynamic applications such as list servers, Web mail, autoresponders, and e-CRM solutions for both incoming and outgoing mail. The list of features is unlimited because you can add to them using CFML. The patent-pending architecture of iMS provides an open-development platform, offering seamless integration with application servers, MS SQL server, and hundreds of other applications. It's a scalable mail server solution that's tailored to meet your business needs.

www.coolfusion.com

— 1st Runner-up —
FUSEAGENT
from FuseLogic

FuseLogic offers cutting-edge solutions for ColdFusion developers. FuseAgent is the command-line user interface for the Internet age. If you're tired of filling out Web-based forms, FuseAgent is the solution. When scheduling the next meeting with your boss using your company's Web-based collaboration environment, simply enter "schedule a meeting with my boss" into FuseAgent's command-line entry point, and FuseAgent will take care of the rest.

www.fuselogic.com

— 2nd Runner-up —
NQCONTENT 1.75
from Netquest

NQcontent 1.75 is an innovative Web content management solution that addresses the site management and administration needs. Having both developers' and end users' needs in mind, NQcontent simplifies the process of content and data management. It's the ideal tool to use to build and update dynamic sites. With NQcontent "time to Web" is measured in weeks, not months.

www.nqcontent.com

THE LARGEST INDEPENDENT JAVA DEVELOPER CONFERENCE IN THE WORLD!

JDJ EDGE conference & expo

JUNE 24-27
JACOB JAVITS CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3
SAN JOSE CONVENTION CENTER
SAN JOSE, CA

JAVA IN JUNE
ESPECIALLY IN NEW YORK

REGISTER ONLINE TODAY
FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

VISIT WWW.SYS-CON.COM

Focus on Java

Java, now mainstream, is the dominant back-end technology upon which next-generation technologies are evolving.

Hear from the leading minds in Java how this essential technology offers robust solutions to *i*-technology professionals and senior IT/IS strategy decision makers.

The Java Fast Paths on June 24, a Java-focused CEO Keynote Panel, and comprehensive conference sessions will focus you on Java every information-packed day!

A Sampling of Java-Focused Sessions

- JAVA 1.4: WHAT'S NEW?
- BUILDING TRULY PORTABLE J2EE APPLICATIONS
- JAVA TOOLS FOR EXTREME PROGRAMMING
- BUILDING ASYNCHRONOUS APPLICATIONS USING JAVA MESSAGING
- .NET VS. J2EE
- J2EE: SETTING UP THE DEVELOPMENT ENVIRONMENT
- BUILDING WEB SERVICES WITH J2EE
- DETECTING, DIAGNOSING, AND OVERCOMING THE FIVE MOST COMMON J2EE APPLICATION PERFORMANCE OBSTACLES
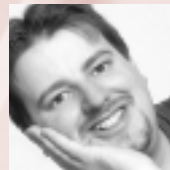
ALAN WILLIAMSON
JAVA CHAIR • EDITOR-IN-CHIEF
JAVA DEVELOPER'S JOURNAL

Featuring…

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 *i*-TECHNOLOGY PROFESSIONALS

Who Should Attend…

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- *i*-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

Hear these thought leaders in interactive, cutting-edge keynote addresses and panels…
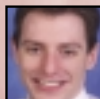
TYLER JEWELL
PRINCIPAL TECH.
EVANGELIST • BEA

GREGG KIESSLING
CEO
SITRAKA

DAVID LITWACK
CEO
SILVERSTREAM

BARRY MORRIS
CEO
IONA

GREGG O'CONNOR
PRESIDENT
SONIC SOFTWARE

RICK ROSS
FOUNDER
JAVA LOBBY

JAMES DUNCAN DAVIDSON
FATHER OF ANT

For Exhibit information
CONTACT: MICHAEL PESICK
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3057
MICHAEL@SYS-CON.COM

SPONSORED BY:

IONA | END 2 ANYWHERE
ADOS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake Enterprise Strength XML
ALTOVA

MEDIA SPONSORS

Federal Computer Week
Webservices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
WIRE
XML MAG
Wrox
jCert
SD Times
OASIS
wireless BUSINESS&TECHNOLOGY
JAVA DEVELOPER'S JOURNAL
BEA WebLogic DEVELOPER'S JOURNAL
XML JOURNAL
WebSphere DEVELOPER'S JOURNAL
ColdFusion Developer's Journal
WebServices JOURNAL

OWNED AND PRODUCED BY
SYS-CON MEDIA   SYS-CON EVENTS

THE LARGEST INTERNATIONAL
XML
CONFERENCE & EXPO
IN THE WORLD!

XML EDGE conference&expo

JUNE 24-27
JACOB JAVITS
CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3
SAN JOSE
CONVENTION CENTER
SAN JOSE, CA

XML-NEXT G
OF ENTERPRISE DEPLOYMENT

REGISTER ONLINE TODAY
FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

VISIT WWW.SYS-CON.COM

Focus on XML

XML is today's essential technology to develop and deploy Web services. Here's your chance to learn from expert practitioners how XML is making the next phase of the Web a reality.

Focus on standards, interoperability, content management, and today's new quest for more efficient and cost-effective Internet and intranet-enabled business process integration.

Focus on XML during information-packed days while you enjoy an XML/Web Services Keynote panel, comprehensive conference sessions, and an unparalleled opportunity to exchange ideas with peers and industry leaders.

A Sampling of XML-Focused Sessions

- XML STANDARDS - AN OVERVIEW
- OASIS STANDARDS UPDATE
- UBL - A UNIVERSAL BUSINESS LANGUAGE
- BRINGING XML TO PKI
- LINK MANAGEMENT WITH XLINK
- PRACTICAL XSLT AND XPATH
- ENTERPRISE CONTENT MANAGEMENT WITH XML
- XML IN THE ENTERPRISE AND INTER-ENTERPRISE WORLD

NORBERT MIKULA
XML CHAIR
BOARD OF DIRECTORS, OASIS
INDUSTRY EDITOR
WEB SERVICES JOURNAL

Featuring…

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 *i*-TECHNOLOGY PROFESSIONALS

Who Should Attend…

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- *i*-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

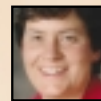## Hear these thought leaders in interactive, cutting-edge keynote addresses and panels…

JEAN FRANCOIS ABRAMATIC
SENIOR VP R&D, FORMER
CHAIRMAN, W3C • ILOG
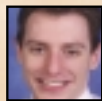
TYLER JEWELL
PRINCIPAL TECH.
EVANGELIST • BEA

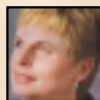**DAVID LITWACK**
CEO
SILVERSTREAM

ANNE THOMAS MANES
CTO
SYSTENET

BARRY MORRIS
CEO
IONA

RICK ROSS
FOUNDER
JAVA LOBBY

PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

For Exhibit information

CONTACT: RICHARD ANDERSON
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3056
RICHARD@SYS-CON.COM

SPONSORED BY:

IONA | END 2 ANYWHERE™
ADOS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
Enterprise Strength XML
ALTOVA

MEDIA SPONSORS

Federal Computer Week
WebServices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
NewsWIRE
jCert
Wrox
OASIS
XML-MAG
SD Times
JAVA DEVELOPER'S JOURNAL
Wireless
XML JOURNAL
BEA WebLogic DEVELOPER'S JOURNAL
WebServices JOURNAL
WebSphere DEVELOPER'S JOURNAL
ColdFusion Developer's Journal

OWNED AND PRODUCED BY

SYS-CON MEDIA   SYS-CON EVENTS

CFDJnews

Synthis Strengthens Commitment to Fusebox

(Atlanta) – Synthis has released Adalon 2.1, with enhanced support for the Fusebox Web Application Standard. A free 30-day evaluation is available for download from www.synthis.com.

Ericom Announces Support for Macromedia ColdFusion

(Hackensack, NJ) – Ericom Software, Inc., a provider of enterprise connectivity and e-business solutions, has announced that PowerTerm Host Publisher now supports the Macromedia ColdFusion Web Application Server.

PowerTerm Host Publisher provides an easy-to-use toolkit to integrate legacy applica-

tions to the Web. Adding support for the ColdFusion environment enables organizations to leverage corporate data and business logic while developing Internet applications for e-business.
www.ericom.com

SYS-CON Media Selects FuseTalk

(Montvale, NJ) – e-Zone Media Inc. has announced that **SYS-CON Media**, the world's leading publisher of *i*-technology print magazines, including *CFDJ*, and producer of developer conferences and accompanying *i*-technology Web portals, has selected FuseTalk to power their worldwide discussion forums.

FuseTalk is a fully customizable and scalable forums solution based on the ColdFusion development platform.

With FuseTalk, **SYS-CON** will be able to provide enhanced discussion capabilities to developers around the world. **SYS-CON** publishes eight monthly *i*-technology journals. Each publication's accompanying interactive portal allows developers to ask questions, share information, and problem-solve online, an important ingredient in **SYS-CON's** mandate to foster developer communities.
www.e-zonemedia.com
www.sys-con.com

Macromedia Elected to JCP Executive Committee

(San Francisco) – The membership of the Java community has elected Macromedia, Inc., to serve a three-year term on the Java Community Process Executive Committee. This committee will shape the future of the Java 2 Platform, Standard and Enterprise

Editions. The 16-member committee makes key decisions concerning Java technology specifications as they progress through the JCP program.
http://jcp.org
www.macromedia.comwww.cfconf.org/CFNorth/

CFDJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
ACTIVE PDF	WWW.ACTIVEPDF.COM	888.389.1653	2
CFDYNAMICS	WWW.CFDYNAMICS.COM	800.422.7957	13
CFNORTH DEVELOPER'S CONFERENCE	WWW.CFNORTH.COM	416.922.1600 ext251	4
COLD FUSION DEVELOPER'S JOURNAL	WWW.SYS-CON.COM	800.513.7111	33
EKTRON	WWW.EKTRON.COM/DEMO	603.594.0249	43
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM	877.215.HOST	31
INTERMEDIA.NET	WWW.INTERMEDIA.NET	800.379.7729	52
JAVA DEVELOPER'S JOURNAL	WWW.SYS-CON.COM	800.513.7111	15
JDJ EDGE CONFERENCE & EXPO	WWW.SYS-CON.COM	201.802.3069	47
JDJ STORE	WWW.JDJSTORE.COM	888.303.JAVA	26, 27
MACROMEDIA	WWW.MACROMEDIA.COM/GO/USERGROUPS	877.460.8679	23
MACROMEDIA	WWW.MACROMEDIA.COM/GO/MASTERING	877.460.8679	25
MACROMEDIA	WWW.MACROMEDIA/DOWNLOADS	877.460.8679	51
PACIFIC ONLINE	WWW.PACONLINE.NET	877.503.9870	39
PAPERTHIN	WWW.PAPERTHIN.COM	800.940.3087	17
QUILL DESIGN	WWW.QUILLDESIGN.COM		8, 11
RACKSPACE	WWW.RACKSPACE.COM	888.850.1470	3
WEB SERVICES EDGE CONFERENCE & EXPO	WWW.SYS-CON.COM	201.802.3069	35, 45
WEB SERVICES JOURNAL	WWW.WSJ2.COM	800.513.7111	36, 37
WEB SPHERE DEVELOPER'S JOURNAL	WWW.SYS-CON.COM	800.513.7111	48
WEBLOGIC DEVELOPERS JOURNAL	WWW.SYS-CON.COM/WEBLOGIC	800.513.7111	48
XML EDGE CONFERENCE & EXPO	WWW.SYS-CON.COM	201.802.3068	49
XML JOURNAL	WWW.SYS-CON.COM	800.513.7111	29

Macromedia
www.macromedia/downloads

Intermedia.net
www.intermedia.net