COLDFUSION Developer's Journal

ColdFusionJournal.com

Out-Moding Modular Presentation with XML

Isaac Dealey 10

This is:

    [ ] Video

    [ ] Flash

    [ ] Don't bother me. I'm staring.

The line between Flash and Video is forever gone.
Travel with us to Scandinavia for a behind the scenes tour
of the award-winning, genre-bending Volvo V50 website.

macromedia.com/go/volvo

Writing Code... and Your Opinion

By Simon Horwith

I want to take this opportunity to announce a new service to the development community – "blog-n-play." Blog-n-play is a Web site launched by SYS-CON that allows anyone who'd like, to create his or her own blog – and it's free. Your blog is associated with one of the SYS-CON publications (anyone reading this editorial is most likely suited to creating a **CFDJ** blog). A unique subdomain of the *CFDJ* domain ("coldfusionjournal.com" in the case of *CFDJ*) is created for your blog. In addition to my personal blog at Horwith.com, you can also visit my CFDJ blog at http:/simon.coldfusionjournal.com.

The blog application is loaded with great features such as a rich text editor for authoring, creating links, skinning your blog, and much more. I recommend it to anyone who has not yet created a blog. It takes two minutes to set up and it's free, so you have nothing to lose. In addition to writing our opinions in blogs, we developers write code...and that's what this month's issue is all about.

ColdFusion developers require some degree of proficiency in many skills. The most basic skill is the ability to write lines of code. A ColdFusion developer's IDE (integrated development environment) is the most important tool in his or her arsenal. This month, *CFDJ* features articles relating to IDEs – software designed to help you write code.

Simeon Bateman and Stephen Milligan (a.k.a. "Spike") have written an excellent article introducing ColdFusion developers to CFEclipse. CFEclipse is a plug-in that adds support for CF development to the free open source Eclipse IDE (very popular with Java developers).

Jeff Fleitz's article introduces developers to Plum. No, I'm not talking about fruit. Plum is a brand-new ColdFusion IDE created by Productivity Enhancement, Inc. – founded by Team Macromedia member Adam Churvis. Plum is unique in that it's not *just* an IDE. Besides the features you'd expect from an IDE, Plum offers developers some of the benefits of an application framework, development methodology, and CMS. It's definitely something worth looking at, particularly for "newbies" and developers looking to make it easy to standardize applications. Don't take my word for it, read the article.

Of course, no issue devoted to IDEs would be complete without examining Macromedia's offerings. Jeffry Houser writes about the features in Dreamweaver MX. Matt Woodward tells about Flex Builder – the IDE Macromedia has given Flex developers.

Isaac Dealey tells how he used XML to create a powerful framework for defining the presentation tier in applications. James Edmunds writes about a ColdFusion Component he created for managing (Verity) searches in his applications.

This month we also have an article from Ben Forta on some of the new features in Blackstone, and a great article from Hal Helms about object-oriented CFML applications. Due to space constraints, we're not offering a Community Focus column this month, but next month's issue is chock-full of great content. The CFEverywhere series will continue soon – space constraints pushed it back slightly.

I hope you enjoy this month's issue. There's nothing more challenging or more fun than installing and experimenting with a handful of different IDEs.
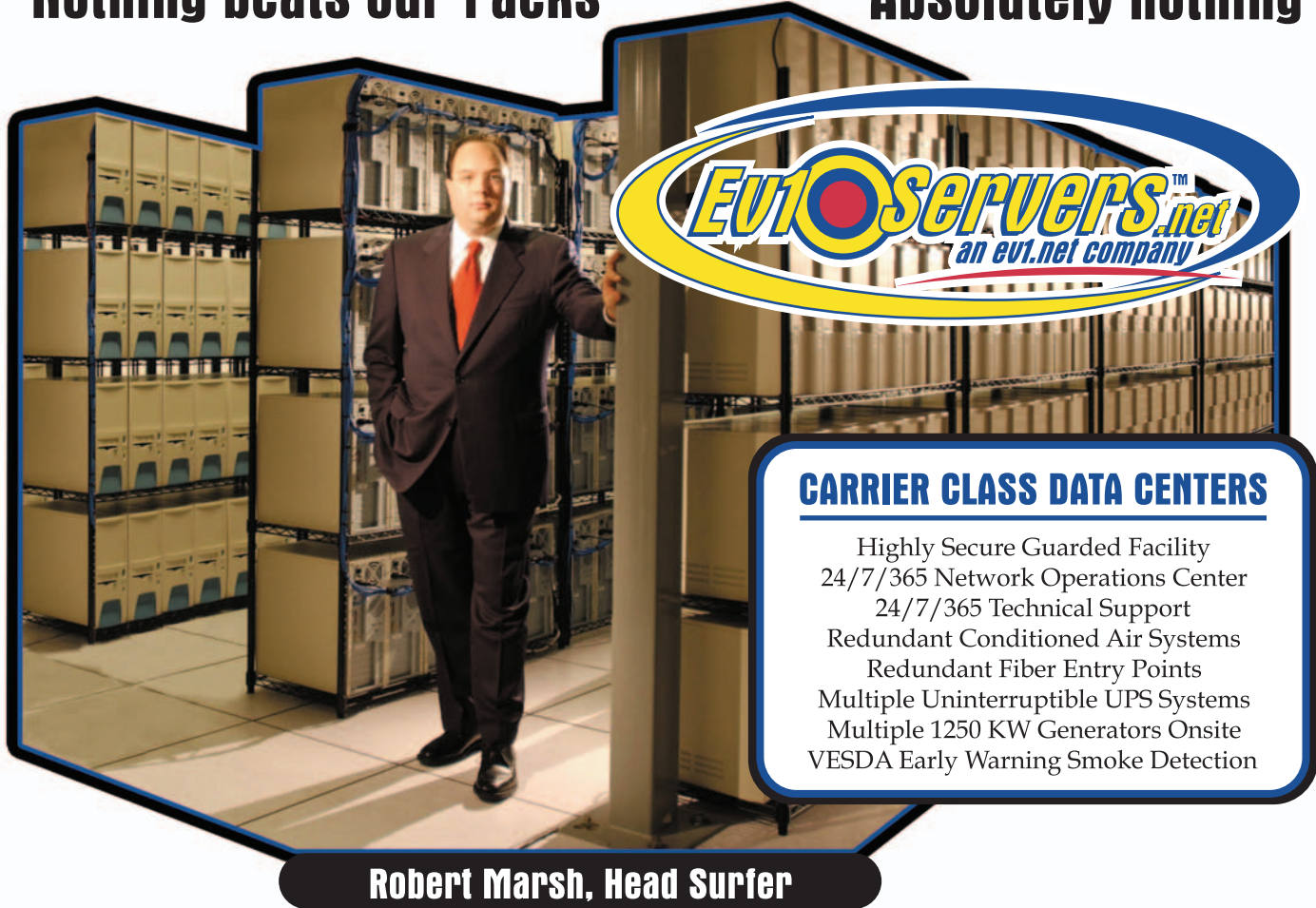
About the Author
Simon Horwith is the editor-in-chief of ColdFusion Developer's Journal and is the CIO at AboutWeb, LLC, a Washington, DC based company specializing in staff augmentation, consulting, and training. Simon is a Macromedia Certified Master Instructor and is a member of Team Macromedia. He has been using ColdFusion since version 1.5 and specializes in ColdFusion application architecture, including architecting applications that integrate with Java, Flash, Flex, and a myriad of other technologies. In addition to presenting at CFUGs and conferences around the world, he has also been a contributing author of several books and technical papers. You can read his blog at www.horwith.com.
simon@horwith.com

<bf>on<cf>

Blackstone and Better Leveraging Java

Making ColdFusion a better Java citizen

The next major version of ColdFusion, code-named "Blackstone," is getting ready to ship, and by gauging customer and partner reactions thus far, we have a winner on our hands.

By Ben Forta

Lots of Goodies

The marquee features in Blackstone have been discussed repeatedly over the past few months (especially at MAX in New Orleans). They include:

• Creating rich and engaging Flash forms using simple CFML tags
• Simplified access to industry standard XForms
• Painless creation of printable content (in FlashPaper and PDF formats)
• Reporting extensions (including a brand new Report Writer)
• Gateways that provide a mechanism to connect to just about any back ends and systems, with ColdFusion both initiating communications and responding to requests
• SMS support, enabling ColdFusion to generate SMS messages as well as to power SMS-based applications

All of these, and much more, promise to change the applications that developers build, as well as how they build them. (I'll be covering them in detail in future columns.) But there are other important features in Blackstone, features that have been getting a lot less coverage. I'd like to mention two specifically in this month's column.

Multiple Instances Revisited

ColdFusion MX Enterprise features support for multiple ColdFusion instances, although that is actually less a ColdFusion feature and more a J2EE server feature that ColdFusion users can take advantage of. ColdFusion MX, after all, is a Java application (deployed as an .ear or .war, like any other Java application). Simply put, using multiple ColdFusion instances provides greater security, greater stability, and greater scalability (almost like having ColdFusion installed on multiple physical servers, but all on one server).

The current shipping version of ColdFusion Enterprise supports the use of multiple instances. If you have an existing J2EE server, you can create multiple .ear or .war files, and can then deploy them as you would any other Java application. If you do not have an existing J2EE server, the ColdFusion installer can install JRun 4 for you, and in doing so will also create and deploy the first ColdFusion instance so that you can be up and running immediately.

When you want to deploy additional instances, then things get a little tricky for users without experience in J2EE server administration. You'll need to use the J2EE server administration tools to create a new server, run the ColdFusion installer to create the .ear or .war, expand the files (if using JRun), make tweaks to an XML file, then copy the expanded files into the server folders – doable, but not exactly a trivial process. (Unfortunately, this is why so many users have yet to deploy multiple instances.)

Blackstone will make this a whole lot simpler. In Blackstone you will have the same three installation options that are present in ColdFusion MX 6.1, but selecting the JRun+CF option in Blackstone installs additional administration screens that make

the deployment of instances (and even the creation of clusters of instances) as simple as any other ColdFusion administration process. You'll be able to simply fill in a form and hit a button to create a new instance, without needing to use the JRun management tools or the ColdFusion installer, without needing any XML tweaks, and without even knowing what an .ear or .war file is.

How could this be used? Consider these use cases:

1. You are deploying a brand new application, one that uses its own data sources and is built by a different development team (who need CF Administrator access), and you want the new application to be safely isolated from your existing production applications. Simply create a new instance, launch the ColdFusion Administrator for that new instance, define the data sources and any other needed settings, copy the code, and you are good to go.

2. You are about to deploy an update to your application code, and need to maintain the existing application as a fallback, just in case something goes wrong. Simply create a new instance (you could even create a .car file using the old instance to save data sources and any other needed configuration, launch the ColdFusion Administrator for the new instance, and import the .car file to import those settings), copy the code, associate your Web server to the new instance, stop the old instance (to prevent resources being used unnecessarily), and you're done. If you then need to roll back, start the old instance, and reassociate your Web server to it. Clean and simple.

3. You have an existing application that is seeing a spike in load (holiday shoppers maybe), and want an additional server running the same application (so that you can handle a greater load, and also provide failover in the event that a server problem occurs). Simply create a new instance, point to the Java package containing the code and settings used for the first instance, and let ColdFusion do its thing. You'll have a second instance created, configured like the first, and containing the same application as the first. You can then use a second screen to create a cluster (perhaps to enable session sharing between the instances). You get the idea.

Of course, for those who want more control, JRun will still be installed with its own management software just like it is now, and you can deploy and manage applications just as you can now. But for those of us who simply want to leverage what is undoubtedly the most significant benefit of ColdFusion Enterprise over ColdFusion Standard, Blackstone will make life much simpler.

Improved J2EE Deployment

ColdFusion (as of CFMX) is a Sun-verified Java application, and is installed on top of J2EE servers like any other Java applications. Well, kind of. ColdFusion MX (including ColdFusion MX 6.1) can indeed be deployed on top of a J2EE server, but the complete process is one that does not excite J2EE administrators.

In J2EE-land, administrators are typically given an application to deploy, and they don't pay a whole lot of attention to what that application is and how it works. Nor should they; developers worry about applications, and J2EE administrators worry about servers staying up and running well. How does this work? Applications to be deployed on a J2EE server are packaged up as a single file, a Java archive file (usually with a .ear or .war extension). The archive file contains everything needed for an application to run – source code, configuration settings, supporting files, everything. Once an application has been tested and is ready for deployment, it's packaged (and of course the package is test deployed), and handed off to the J2EE administrator who drops it onto the J2EE server (okay, so I'm simplifying things a bit, but the basic flow is accurate). What J2EE administrators don't do (or don't like doing) is run through a post-installation to-do list containing things like create a data source, set up some mappings, install these extensions, and so on.

And yet J2EE administrators deploying ColdFusion MX must do just that. ColdFusion itself (the core engine, compiler, and runtime services) can indeed be deployed like any other Java application, but that's just ColdFusion itself. Once ColdFusion is deployed someone still needs to move all of the .cfm and .cfc files over, and use the ColdFusion Administrator to define data sources and mappings and more. In other words, while ColdFusion itself is deployed like any other J2EE application, the total experience of deploying a ColdFusion application is not.

Blackstone changes this by allowing complete J2EE deployment packages to be built. Blackstone comes with a packaging tool that creates a complete .ear or .war file that can contain the ColdFusion runtime (with or without specific features), application code, data sources, and more. The tool can take some time to run (building a complete deployable .ear or .war is not a quick process), and when complete that package can be given to a J2EE administrator to be deployed just like any other Java applications. This means that ColdFusion applications can even be deployed on a J2EE server that is not running ColdFusion, because the ColdFusion engine will be packaged in the Java package file.

This is an important, and much needed enhancement. From a J2EE administrator's perspective, deploying Blackstone applications will be just like deploying any Java applications. Actually, they'll not even have to know that it's a ColdFusion application – it's Java, pure and simple.

Conclusion

Blackstone is an incredibly exciting release, and I'll be covering Blackstone features in depth in future columns. But beyond developer features, Blackstone also better leverages underlying Java and J2EE to simplify deployment, stability, and security.

About the Author

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

ben@forta.com

Out-Moding Modular Presentation with XML

Manage your application's presentation and provide a rich user experience

By Isaac Dealey

In my experience, the majority of client change requests for any given application have to do with the presentation layer. This makes sense if you're a client, because the presentation layer is the only part of the application you see.

Thus, it's the one part of the application you can judge and comment on. By and large, clients wouldn't even know how to comment on other parts of the application, except by proxy in seeing how those parts of the application influence the presentation layer. For instance, if a form submission performs an update of data, the client only knows that the functionality performing the update works if the presentation layer indicates the change. If the presentation layer doesn't reflect the change, the client will wonder if anything actually happened. This is because many clients don't have the luxury or know-how to open the database in another IDE, such as Enterprise Manager, to view the data. Even if clients did have this ability, most wouldn't know how or why they should use it.

Unfortunately, despite the fact that it is the most crucial tool for communicating with clients, the presentation layer of an application is often not thoroughly planned, or, worse, simply neglected. I can think of many times I've seen a presentation of a framework, architecture, or other application and asked, "How does X handle the display?" only to hear, "It's very dynamic: you just place your templates in this directory!" This is as though the best answer to versatility in presentation is simply to have no plan at all. How is the application branded (for different clients using an ASP application)? How is localized content presented for multilingual sites and applications? How is presentation filtered for different users and roles? With a little finesse, it's possible to take the presentation layer even further: you can reduce your workload while integrating the interfaces of disparate applications, with little if any impact on their code.

A Magic Trick

These are bold claims, so I don't blame you for being skeptical. Until I faced the same challenge while designing the onTap framework, I too was a skeptic. The onTap framework meets this challenge by using its HTML library and its suite of XML parsing functions to build HTML library presentation. Here is a brief example. To test it, you will need to install the onTap framework from www.fusiontap.com (see Listing 1). All listings for this article can be downloaded from www.sys-con.com/coldfusion/sourcec.cfm.

Looking at the code in Listing 1, you might not think it looks very different from the display templates of most applications. You would be correct in that assessment. The primary difference is the addition of the CFMODULE tag calling the onTap framework's

HTML custom tag. This tag is tasked with converting your XHTML template into HTML library structures to be used later. By itself, this example isn't very impressive: it simply takes a string of XHTML with an XML namespace and displays HTML or XHTML. The impressive part is that the converted XML can now be modified ad hoc to become whatever you need. The possibilities even include simplifying some complex DHTML, which would be difficult to accomplish using XSL transformations.

Let me start with something easy. Insert the following line of code between the closing CFMODULE tag and the opening CFOUTPUT tag:

```
<cfset htlib.childSort(view.mainmenu)>
```

You should notice that the order of the links in the menu is altered so that the links appear in alphabetical order. Now, for something a bit more involved, add these lines:

```
<cfset htlib.style(view.mainmenu,"float","left")>
<cfset htlib.addChildContainers(view.mainmenu,"div",true)>
```

Figures 1 and 2 are before-and-after screen shots depicting the changes you should see in the display after adding these lines of code. You should notice that the menu previously distributed horizontally across the top is now distributed vertically on the left side – a change resulting from just these two additional lines of code.

Indulge me while I offer one more example. Comment out the previous two lines and add the code below:

```
<cfloop index="x" from="1" to="#htlib.children(view.mainmenu)#">
    <cfset mychild = htlib.childGet(view.mainmenu,x)>
    <cfset mychild = htlib.clone(mychild)>
    <cfset htlib.childAdd(view.pagefooter,mychild)>
</cfloop>
```

The HTML library offers many alternative ways to accomplish this task. This is certainly not the most sophisticated or elegant method, but it's an easy example that shows how these modular display elements can be easily reused. When these lines of code have been added to the example, the footer menu at the bottom of the page is populated with the same links used in the main menu at the top. The HTML library offers many other tools for display, including rich interface elements such as trees and tabsets (see Figure 3).

A Look Behind the Curtain

I've shown you how easy it can be to get the display of a Web site to roll over and go fetch. Now I'll explain how this functions.

In a nutshell, the HTML library works because HTML and XHTML are structured documents. As with any standardized format, the structure of the document lends itself to being managed because its syntax is predictable. When you know the structure of the language syntax, you can create routines to handle or manage that structure. In essence, this is very similar to writing routines to import or export tab-delimited files or CSVs. The format of the file provides all the information needed to manage the import/export process. In the case of our HTML or XHTML, the process is made a little easier because many of the low-level formatting concerns (e.g., converting escaped characters) are handled for us by the ColdFusion Native XMLParse() function. But in other ways it is more complex. Instead of a linear series of records to insert into a database, the XML document is a nested hierarchy of elements, like a tree. Each has a collection of many disparate properties needing unique handling.

As previously mentioned, reading an XML document is a reasonably easy task because an XML document is consistently structured. Once read-in using the ColdFusion native XMLParse() function, the series of structures and arrays that represent an XML document can be used in many ways. The custom tag used in these examples chooses to convert them further, from XML documents into native ColdFusion structures and arrays specially formatted to work with the HTML library. This extra conversion is done for two reasons. First, although using an existing XML document within ColdFusion isn't very hard, modifying XML documents can be surprisingly difficult. This is particularly true with regard to the addition of child nodes. The native structures and arrays don't have this limitation. Second, an HTML or XHTML document is actually a much more complex document than an "ordinary" XML document, having both CSS and JavaScript features. Although XSLT provides support for CSS, it has no mechanisms for managing JavaScript and can be a daunting shift in thinking for traditional ColdFusion developers. In comparison to structures and arrays, XSLT can also require a significant amount of server resources when making small changes to a document. So XSLT isn't necessarily the most efficient mechanism to use for subtle changes such as adding or removing a single link from a menu in a large HTML document.
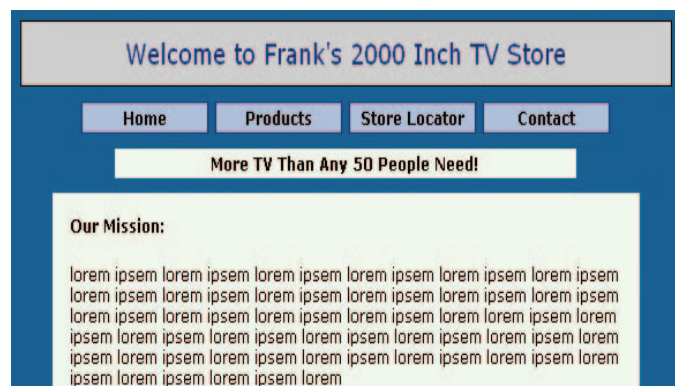
Figure 1: Before-menu at the top

Figure 2: After-menu to the left
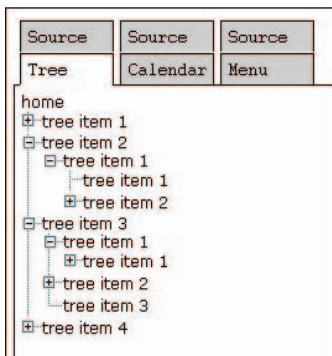
Figure 3: Rich interface elements

When the HTML library was first released, many developers found it to be rather intimidating. Full pages of pure CFSCRIPT to generate the presentation elements looked very different from the traditional template-driven displays most of us are used to seeing. The new HTML custom tag helps to bridge this gap by allowing existing templates to be converted so that they can use the HTML library with little or no modification. Older templates must be formatted as valid XML (no unclosed tags, no unquoted attributes, no value-less attributes, and proper use of XML entities), however this is far less work than converting to pure CFSCRIPT. The XML syntax is only half the story. Many of the more powerful features of the library require an understanding of library's underpinnings, which support the XML syntax.

The good news is that the underlying architecture of the HTML library is not as complicated as it might appear at first glance. In truth, it is quite simple because it is based on the standards of HTML, the format that made Web technology accessible to everyone. In an HTML document, each tag (aka, node or element) has several features:

- *Element:* This is the name of the HTML element type being created, such as "div," "span," or "input." An element must have both an opening tag (<div>) and a closing tag (</div>), or it must be closed in the opening tag (<div />).
- *Children:* This is the array of HTML tags between a tag or node opening and the closing tag. The children of a given HTML element may be manipulated using the html.childX functions in the HTML library, such as html.children(), html.childAdd(), html.childMove(), and html.childRemove.
- *Attributes:* These are the name/value pairs found within the opening tag of a node, such as the name attribute of an input element or the src attribute of an image tag. Every HTML element may have an ID attribute, class attribute, and style attribute. Using the HTML library of the onTap frameworks, these values can be managed with the collection of html.attributeX functions in the HTML library, such as html.attribute(), html.attributeAppend(), and html.attributeRemove().
- *Style:* CSS style sheets control the display of HTML elements once the browser receives the document. They can be applied to HTML elements in three ways: (a) using external style sheets (attached to the document using a <LINK> tag), (b) using the <STYLE> tag, or (c) using in-line style elements that are applied as a "style" attribute in the target HTML element. With an HTML library, in-line style elements for a given HTML element can be managed using the collection of html.styleX functions in the HTML library, such as html.style(), html.styleAppend(), and html.styleRemove().
- *Events:* The Document Object Model (DOM) for HTML offers a host of event handlers. These include special attributes, which are applied to HTML nodes to provide dynamic presentation within the browser, such as "onclick," "onmouseover," and "onchange." JavaScript is the most common client-scripting language used to implement these event handlers. Unfortunately, JavaScript has no XML syntax and must therefore exist as plain text, outside of our wonderfully structured XML. The HTML library and the new XML syntax for creating HTML library elements seek to provide better management for client-side JavaScript. Events can be managed using the collection of html.eventX functions, such as html.event(), html.eventsShow(), and html.eventRemove().

Beyond the Document

In addition to managing client DOM events, the HTML library also includes a number of additional "server events" that allow the presentation of these elements to be modified at different stages during a ColdFusion page request. Most of the library's server events are triggered when the html.show() function is used to convert the series of structures and arrays representing the HTML into a string of HTML. The following events are available to all elements.

- *Pre:* Occurs prior to the opening tag of the HTML node.
- *Post:* Occurs after the closing tag of the HTML node.
- *Open:* Occurs inside the opening tag of the HTML node, before any children or prechild events are displayed.
- *Close:* Occurs inside the closing tag of the HTML node, after all children and all postchild events.
- *Prechild:* Occurs prior to the display of each child element.
- *Postchild:* Occurs after the display of each child element.
- *Childdelimiter:* Occurs between multiple children (after the postchild event of the previous child and prior to the prechild event of the next child). If an element has less than two children, this event will not execute.

These server events can be populated with a variety of elements: functions that can modify the tag and/or return content to the display, plaintext strings that are injected directly into the HTML output, or other HTML elements. This article has already shown an example using a plaintext value. If you examine the XML syntax in Listing 1, you'll notice that the pagefooter node of the XML contains the attribute tap:childdelimiter="|". This inserts the pipe character between each of the child nodes when the footer is displayed. Thus, it prevents repeated instances of the value in the code that might need to be updated if you decide to change the character used to delimit the menu.

Smart Upgrades Through Extensibility

As Web application developers, the most significant portion of our time is likely spent developing forms to allow our users to manage their databases: create/read/update/delete (CRUD). Particularly with regard to updates, this can often be a tedious task. When an update form is presented, it must be prepopulated with existing data. One must determine where the data is gathered from and how it's used to populate the form. Especially when working with applications inherited from previous developers, we often find that the data used to populate forms hasn't been properly escaped with the HTMLEditFormat() function.

Web (r)evolution:

KTML

Let your clients edit their Web sites content through the browser

Visually use CSS styles (Word like)
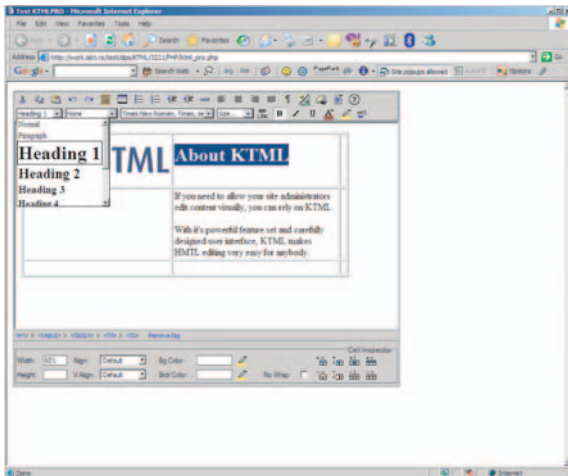
Native Dreamweaver MX integration
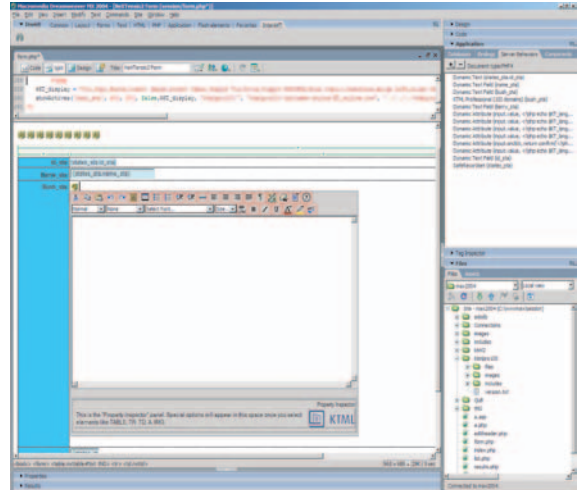
Easy-to-use image explorer
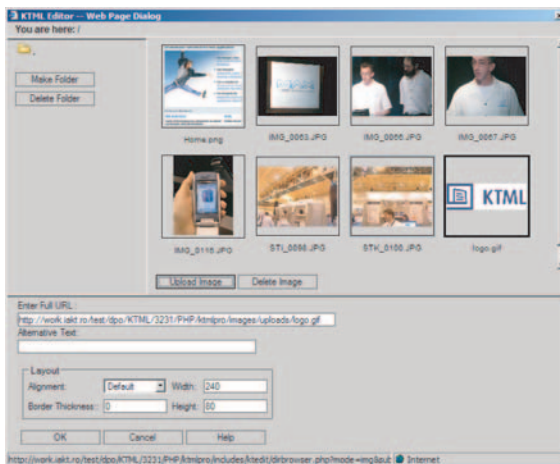
Multi-language spell-checker
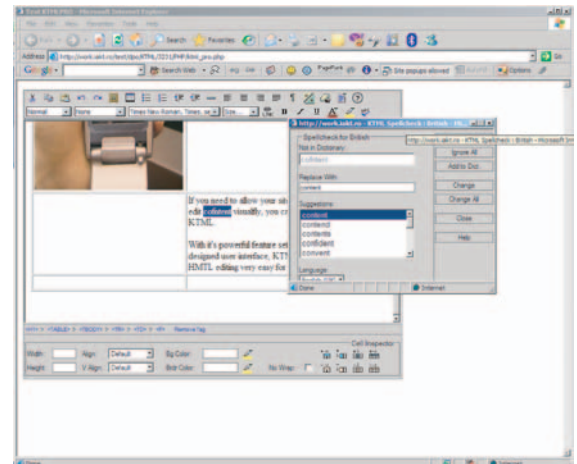
KTML 3.5 - HTML online editor

See all the features, benefits and a live demo at:
http://ktml.interaktonline.com/

Abstractions

Understanding their role in application design

I've been busy lately providing training in implementing object orientation (OO) with ColdFusion components (CFCs) to several companies. I've found that most ColdFusion developers approach OO as something to "layer on" over their traditional programming practices. Even many Java developers make this mistake, which means that they don't often see many benefits from a foray into object orientation. The error stems from a lack of understanding. OO design is fundamentally different from procedural design, so a major shift in thinking is required in order to gain the benefits of greater code maintainability and reuse.

By Hal Helms

The concept of abstraction underpins all other principles and practices of object orientation. In his book, *Object-Oriented Design with Applications*, Grady Booch (one of the authors of the Unified Modeling Language) defines abstraction in the following way:

"An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of object and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer."

Abstractions are concerned with the simplification of reality, sparing users the myriad of details that may be associated with that reality. All abstractions have two things in common: an interface and one or more implementations. Users of the abstraction simply need to learn the interface, ignoring what may be a bewildering level of detail in the implementation. Abstractions make it possible for people like me to drive cars and run dishwashers.

Abstractions are not new to object orientation. In fact, you're very familiar with one particular abstraction: ColdFusion. When you want to send an e-mail, for example, you use the <cfmail> tag. CFML is an abstraction of a lower-level language, Java. Java itself is an abstraction of a still lower-level language. There's nothing you can do in ColdFusion that you cannot do in Java, but we value ColdFusion because it provides a simple interface to the complexities of the underlying languages.

When we undertake the creation of an object-oriented design, we engage in the process of creating and defining abstractions – at least we do if we want a good design. But people engaged in application design too often mistake their mission: they think they're writing code instead of designing abstraction layers. Thus, they concentrate on the implementation and neglect to abstract the details into a simpler interface.

Let me provide an example. Recently, I spent a week at a large bank helping developers "get their heads around" object orientation. I find it helpful to use examples related to the client's domain, so we began by talking about banks and the banking system. What I wanted them to derive was a domain model – an abstraction of their world that could be modeled in software. What they quickly began work on was a data schema for a relational database.

I expected this: most developers look to the structure of the relational model to provide structure to their code. But databases – at least in the OO world – exist only to make it possible to persist objects. It's the objects – those abstractions of the real world – that provide an OO program with its ability to accomplish tasks.

The trouble with viewing the application through the lens of the database schema is that, while the relational model is an excellent one for storing data, data is the wrong abstraction layer for application design. (Data itself is an abstraction: we don't store actual dollars in a database, we store numbers that stand for dollars.)

In tackling the challenge of a complex application, we need to employ a "divide and conquer" strategy. Large problems are composed of several constituent problems; each of these smaller problems can be further broken down. Every level is an abstraction, with its own stable interface and an implementation that is subject to change. This is the essence of object-oriented design.

Our challenge in writing applications is to derive abstractions on multiple levels. The needs of the system's users must be abstracted, the details of the system itself must be abstracted, and, finally, the underlying business area or domain must be abstracted. This threefold abstraction process has been formalized into a design pattern called Model-View-Controller,

When was the last time your Intranet was updated?

Announcing the Macromedia® Web Publishing System. You build and manage the site. Users keep content fresh.

With Macromedia® Studio MX 2004, Contribute™ 3 and Contribute Publishing Services, you can affordably build, manage, and publish enterprise sites. And unlike the average content management system, this works. You determine who can edit and who can publish. Then users simply point and click to update any page. In no time, dated content will be a thing of the past. Learn more at **www.webpublishingsystem.com**

macromedia®
WEB PUBLISHING
SYSTEM

into which each component of an application fits. User interfaces are the purview of the view layer, the workings of the base application machinery fall under the command of the controller, and the business domain is the responsibility of the model.

View components are usually the least interesting to programmers, although they are the most interesting to actual clients. The controller is often implemented as a framework. In the ColdFusion world, both Fusebox and Mach-II are controllers for applications. The model layer must be individualized for each business and each domain. The creation of a domain model consists of abstracting base elements, from which we build up larger elements until we have a good abstraction layer that can serve many individual applications.

Let's go back to the bank example. One of the base elements of any banking system is the idea of a currency issued by a country. Procedural programmers might represent this by a string such as "euro" or by a number that represents the currency. In OO land, we abstract the notion of a currency as an abstract data type.

We're used to thinking of data types in terms of boolean values, numbers, or strings. But OO's data types tell us what kind of behavior the data type is capable of in addition to what kind of information the data type can hold. In other words, a data type has both properties and methods. If this sounds like the description of a ColdFusion component (CFC), you're quite right: CFCs are

Figure 1: The UML class diagram for a simple banking domain model

abstract data types. To model a currency, we would create a Currency CFC that looks something like this:

```
<cfcomponent displayname="Currency">
    <cfset variables.name =  "null" />
    <cfset variables.country = "null" />

    <cffunction name="init" access="public"
      returntype="Currency" output="false">
            <cfargument name="name"
              type="string" required="true" />
            <cfargument name="country"
             type="string" required="true" />
            <cfset setName(arguments.name) />
            <cfset setCountry(arguments.
              country) />
            <cfreturn this />
    </cffunction>

    <!--- getters and setters --->
    <cffunction name="getName" access="public"
      returntype="string" output="false">
            <cfreturn variables.name />
    </cffunction>
    <cffunction name="setName" access="public"
      returntype="void" output="false">
            <cfargument name="name"
              type="string" required="true" />
            <cfset variables.name = arguments.
              name />
    </cffunction>

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

When I showed this to the class, most people thought I had missed an amount property. Otherwise, how could we model something like depositing $500? All that the Currency CFC should

do, however, is abstract the idea of a currency – not a specific number of dollars. To reflect an actual transaction, such as a deposit, we should create an abstraction of a transaction.

To do this, we ask ourselves, "What is common to all transactions?" We might decide that all transactions have a currency, a specific quantity of that currency, and a bank account. We would then create another abstraction/CFC: Transaction.

```
<cfcomponent displayname="Transaction">
    <cfset variables.currency = "null" />
    <cfset variables.quantity = 0 />
    <cfset variables.bankAccount = "null" />

    <cffunction name="getCurrency"
      access="public" returntype="Currency"
output="false">
    <cfreturn variables.currency  />
  </cffunction>
  <cffunction name="setCurrency"
    access="public" returntype="void"
      output="false">
  <cfargument name="currency" type="Currency"
    required="true" />
  <cfset variables.currency  = arguments.
    currency />
  </cffunction>

  <cffunction name="getQuantity"
    access="public" returntype="numeric"
    output="false">
  <cfreturn variables.quantity  />
  </cffunction>
  <cffunction name="setQuantity"
    access="public" returntype="void"
    output="false">
  <cfargument name="quantity" type="numeric"
   required="true" />
  <cfset variables.quantity  = arguments.
    quantity />
  </cffunction>

  <cffunction name="getBankAccount"
    access="public" returntype="BankAccount"
      output="false">
  <cfreturn variables.bankAccount  />
  </cffunction>
  <cffunction name="setBankAccount"
    access="public" returntype="void"
    output="false">
  <cfargument name="bankAccount"
   type="BankAccount" required="true" />
  <cfset variables.bankAccount  = arguments.
    bankAccount  />
```

```
        </cffunction>
</cfcomponent>
```

Notice that the bankAccount property is not a string or number, but an object of type BankAccount – another abstraction.

The class thought that I had also missed another important aspect: the type of transaction, such as "deposit" or "withdrawal." They expected to see a property labeled transactionType. While we might do this in procedural programming, in object-oriented programming we rely on types rather than labels so we would subtype Transaction with a Deposit CFC and a Withdrawal CFC. The simple UML diagram showing the relationship between components for a simple banking model might look like Figure 1.

If this strikes you as a lot of "stuff" for something so simple, you're not alone: most of my students thought that I was exaggerating abstraction to make a point. Even a humble address is abstracted into its own abstract data type. Why? First, abstracting the idea of an address into its own type allows for consistency of addresses (as well as code reuse) for any component that needs an address. Second, is it not likely that we will have to accomodate different types of addresses? The one I've shown works for U.S. locations, but other areas organize their addresses differently. I can now subtype Address to reflect these different address types.

When we design around abstractions, we help ensure that the inevitable changes that will occur over the life of a program are localized to a few components – and that code that relies on the component's interface will continue to function. With the release of ColdFusion MX, ColdFusion's implementation underwent a drastic change, but the interface represented by its tags and functions remained the same – and the applications that were built to rely on that interface continued to run.

Thinking in terms of abstractions is not a natural process for most of us. Since it's code that makes the application work, we tend to think of ourselves as suppliers of code – coders. But defining ourselves in this way devalues our work and creates code that is fragile against the realities of business needs that continuously evolve. Abstraction layers provide a bulwark against the dangers of raging complexity.

When programmers (aka, "abstraction builders") struggle with object orientation, it's seldom the syntax of any particular OO language that trips them up. Rather, the difficulty usually lies in coming to terms with abstraction itself. But the benefits are well worth the work.

About the Author

Hal Helms is the author of several books on programming. Hal teaches classes in Java, C#, .NET, OO Programming with CFCs, Design Patterns in CFCs, ColdFusion Foundations, Mach-II, and Fusebox. He's the author of the popular "Occasional Newsletter." Hal's Web site is www.halhelms.com.

hal@halhelms.com.

Macromedia Flex Builder 1.5

Visual Flex development and powerful debugging in a familiar environment

By Matt Woodward

Over the past couple of months I've been introducing you to some of the fantastic possibilities Macromedia Flex offers for building Rich Internet Applications (RIAs) with ColdFusion ("Flex Your ColdFusion Muscles," *CFDJ* Vol. 6, issue 12 and "Data Presentation with Macromedia Flex," *CFDJ* Vol. 7, issue 1), and I hope you've been motivated to begin experimenting with Flex yourself.

What I haven't yet addressed is the set of development tools available for Flex. One of the great things about Flex is that since Flex development is entirely text based, you can use the text editor of your choice to write your Flex code.

A good IDE that's designed to work in conjunction with a specific technology, however, can make your development life a bit easier by offering features such as syntax highlighting, code hinting, debugging, and even drag and drop, WYSIWYG UI development capabilities. If you've been doing Flex development with a generic IDE and have been searching for a better alternative, Macromedia Flex Builder 1.5 might be just what you're looking for.

Flex Builder offers all the capabilities I outlined above, is tightly integrated with the Flex server for previewing and testing, and since it's built on the Dreamweaver codebase, it will be quite familiar to ColdFusion developers who use Dreamweaver for ColdFusion development. Other than the addition of Flex capabilities, Flex Builder behaves exactly as Dreamweaver does, so it can handle all your ColdFusion projects just as you may be accustomed to with Dreamweaver. If you're doing both Flex and ColdFusion development, you may quickly find Flex Builder becoming your primary IDE.

Flex Builder 1.5

Flex Builder 1.5 was released and made available for download on Macromedia's Web site (www.macromedia.com/cfusion/tdrc/index.cfm?product=flex) in late December of 2004. Previously you had to order a trial CD in order to get Flex Builder, so the ability to download Flex Builder is a great way to get started quickly, and the download will allow you to try Flex Builder for 60 days. If you need a permanent license for Flex Builder, the purchase of Macromedia Flex includes five licenses for Flex Builder or you can apply for a noncommercial Flex license at www.macromedia.com/software/flex/productinfo/faq/#ancni, which includes a single license for Flex Builder. Because Flex Builder is really designed specifically for Flex developers and not intended as a more general-purpose IDE, Flex Builder is not offered for sale as a stand-alone product.

If you've tried Flex Builder 1.0 in the past and had some complaints about its performance, you'll definitely want to check out Flex Builder 1.5. In addition to support for Flex 1.5 features (e.g., the new charting and graphing components) you now also have the ability within Flex Builder to turn off the design view to gain maximum performance. (In Flex Builder 1.0 the design view was essentially always working behind the scenes even if you weren't using it.) This is a great addition for code-centric developers, and since this feature is controlled by a simple checkbox, it can easily be turned on or off as desired. Also, if you're developing for both Flex 1.0 and 1.5 you can run Flex Builder 1.0 and 1.5 in parallel with one another. Flex Builder 1.0 serial numbers are valid for Flex Builder 1.5, so you can easily use both versions if necessary.

Working with Flex Builder

When you first open Flex Builder you may think Dreamweaver has just gone haywire and changed its color scheme. There's no problem; the interface for Flex Builder is nearly identical to that of Dreamweaver MX 2004, so if you're a Dreamweaver user you won't miss a beat switching over to Flex Builder. Everything from the file and asset navigation panels, to the site management tools, to the rich CSS functionality that you've come to know and love in Dreamweaver are all present in Flex Builder.

Perhaps the major difference between Dreamweaver and Flex Builder is the ability to build your Flex UIs visually, but there is also added functionality for previewing your Flex applications from within the IDE, an Application and Network Debugger, and more tools specifically designed for Flex developers.

As I mentioned above, you can continue to use other text editors or IDEs to do Flex development, but Flex Builder is the

only IDE that offers a visual development environment for Flex applications. Even for code-centric developers, the ability to quickly drag and drop Flex components to the screen to build UIs can be a very powerful addition to the toolbox, and you'll likely find yourself depending on the other Flex-specific tools as well.

Because Flex Builder is essentially Dreamweaver with additional functionality designed for Flex development, for the remainder of this article I'll focus on the features in Flex Builder that will appeal specifically to those of you doing Flex development. As an aside, I hope once you see how easy Flex development can be with Flex Builder and how it can actually help you learn Flex, you'll be further tempted to try your hand at building Flex applications!

Seeing Is Believing

In addition to the obvious benefits of MXML syntax highlighting and tag completion that Flex Builder offers, the single biggest feature that will probably most appeal to Flex developers is the ability to visually build Flex UIs. The ability to choose between design, code, and split view can really accelerate development, particularly as you begin learning Flex. Many of the Flex containers and components have quite a bit of functionality built into them, and seeing the components and their related code, not to mention being able to change something in the design view and watch the code get updated (or vice versa), can be an extremely powerful learning tool.

As a brief example, let's take a look at how Flex Builder's design view can make our lives a bit easier with the Flex ViewStack container. The ViewStack container is essentially what it sounds like, namely a group of views that are stacked on top of one another, with only one view visible at a time. Each view within the stack can contain other containers and components. In the visual design mode, Flex Builder puts a navigation control above the ViewStack that allows for the addition and removal of views to the ViewStack as well as easy navigation between the views (see Figure1).

As you may imagine, this navigator can come in particularly handy as your UIs become more complex, and it's also a part of the Accordion and TabNavigator components when you're working in the visual design mode. I do the vast majority of my development by handwriting my code, but I have found that these features of the design view are handy for quickly creating the UI for the application. Then with the basic UI framework in place I can focus on writing the remaining code involved.

One of the big new additions to Flex 1.5 is the group of charting and graphing components. You'll be happy to know you can quickly and easily build your charts and graphs visually with Flex Builder (see Figure 2). Simply drag a chart or graph component onto the screen, provide some basic information about your chart, and Flex Builder handles generating the MXML tags for you. Then all that's left to do is hook the chart into dynamic data as needed. This is also a great way to learn the charting and graphing components if you haven't worked with them before.

One other thing to note regarding the design view is that there are two modes: standard and Expanded. Standard

mode represents the UI extremely accurately within Flex Builder, while Expanded mode adds a slight margin around each component (10 pixels to be exact) and adds a dashed line around each container to show the container's boundaries. While this means that the UI isn't rendered quite as precisely in Flex Builder in Expanded mode, you may find that it's easier to visually select containers and precisely place components within containers. Once

you've built much of your UI, you can switch to Standard mode with the click of a button and see a precise visual representation of your Flex application in Flex Builder.

The design view in Flex Builder is extremely powerful and can give a real kick-start to your Flex application development. Even if you're someone who prefers to do all of your coding by hand, laying out the initial UI using the design

view allows you to build the UI faster and leaves you more time for your MXML and ActionScript coding.

What? Bugs in MY Code?

I'm sure the rest of you write completely bug-free code on the first try, but I have a confession to make: occasionally a few bugs creep into my code and when they do, sometimes they can be tricky to track down. The Flex server makes debugging relatively easy because of the extremely helpful error messages it generates. If the Flex error messages don't tell the whole story, Flex Builder's built-in application debugger comes to the rescue. Using Flex Builder's debugger you set breakpoints in your code, step through code line-by-line, and perform interactive debugging within Flex Builder.

A right-click of the mouse in your MXML or ActionScript code brings up a menu from which you can select the option of adding a breakpoint. This tells Flex Builder you want the application execution to pause at that point so you can see what's going on right at that moment. Breakpoints are a great way to check before and after states at various points in your code because in some cases, you may get to the end of a relatively complex operation that isn't doing what you think it's supposed to be doing. Without breakpoints it's sometimes difficult to determine where the missteps in the code are occurring. Breakpoints can be removed as easily as they are added, and there's an option to remove all breakpoints at once after you've hunted down that pesky bug.

Once you set your breakpoints, simply click the Debug button at the top of the Flex Builder interface. If you have your test server configured, this will compile the MXML file, transfer it to your testing server, and launch the application within Flex Builder. If you prefer, you can even debug the application in an external browser. The application will run in the external browser and, as breakpoints are reached, Flex Builder will regain focus so you can review what's going on behind the scenes in your application.

If you debug within Flex Builder, above the Flex application's interface you'll see a set of controls that allow you to play, stop, refresh, and step through each of your breakpoints. Below the application's interface you'll see a set of tabs that allow you
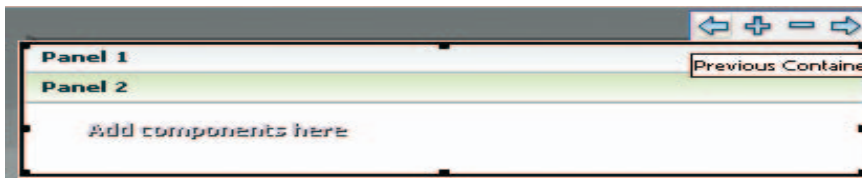
Figure 1:   ViewStack navigation controls
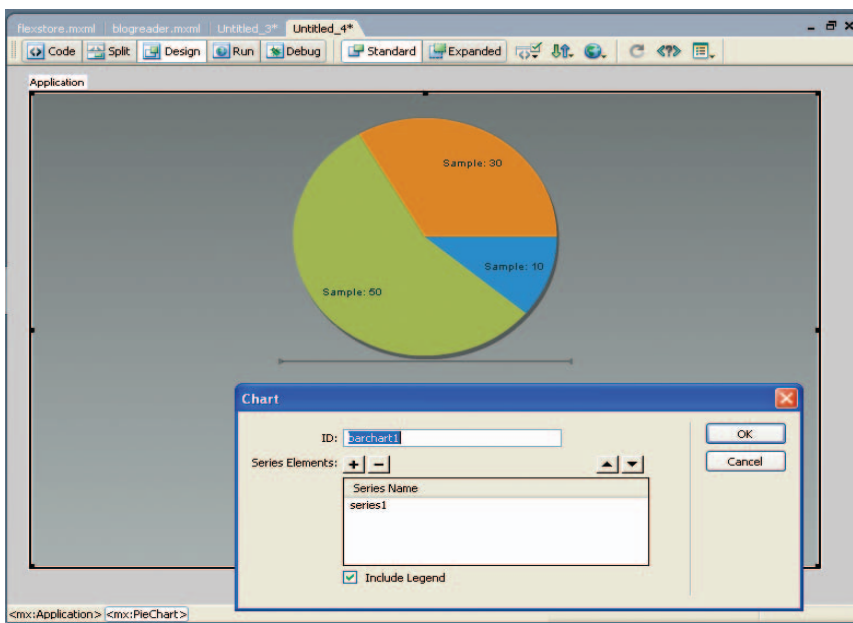
Figure 2:   Visually build charts
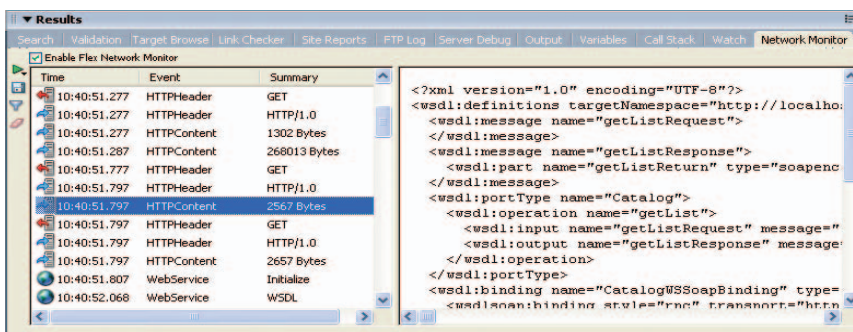
Figure 3: Flex Builder's Network Monitor

to see things such as server debug and call stack information. All in all, Flex Builder offers an extremely rich set of tools that can make your debugging tasks much less difficult.

It's All About Networking

As powerful as Flex is, just as with other Web applications Flex usually relies on external data sources of one form or another from which Flex retrieves and displays dynamic data. Particularly when Web services are involved, if there are network problems or other issues outside of the Flex application itself, sometimes you may experience what I usually refer to as a "silent error." For example, you may not see any data in the DataGrid you're trying to populate, but you don't see any error messages either. For troubleshooting pesky problems such as these, Flex Builder's Network Monitor is indispensable (see Figure 3).

The Network Monitor keeps a running list of all the network activity occurring in your Flex application including get, post,

other HTTP traffic, Web service calls, and SOAP packets. Unless the data returned from a network call is binary, you can even see the actual data being sent or returned in the Network Monitor. This is an extremely handy way to see what your application is sending to a Web service, for example, as well as what you're getting back from the Web service. As I mentioned previously, if you're getting a "silent error" of some sort, without being able to see the details of the network traffic, these types of bugs can be extremely frustrating and time-consuming to diagnose and repair.

Conclusion

This was a very brief introduction to Flex Builder 1.5, but if you like Dreamweaver and you're building Flex applications, you'll love Flex Builder. It can make quick work of visual design, application and network traffic debugging, and can even help you learn Flex with the tag completion and introspection. And best of all for ColdFusion developers, you don't give up any of the

ColdFusion features of Dreamweaver that you've come to rely upon. For those of you doing Flex development in conjunction with ColdFusion, Flex Builder 1.5 may be the ideal tool for you.

Resources

- *Macromedia Flex Developer Center:* www.macromedia.com/devnet/flex/
- *Macromedia Flex Builder Developer Center:* www.macromedia.com/ devnet/flex/flex_builder.html

About the Author

Matt Woodward is a Web application developer for i2 Technologies in Dallas, Texas, and also works as a consultant for his company, Sixth Floor Software. He is a Macromedia Certified ColdFusion Developer, a member of Team Macromedia, and has been using ColdFusion since 1996. In addition to his ColdFusion work, Matt also develops in Java and PHP.

mpwoodward@mac.com

Getting to Know Dreamweaver MX

A powerful tool that keeps getting better

For as long as I can remember, I've been using ColdFusion Studio to develop my applications. Unfortunately, CF Studio appears to be on its last legs. It is still included with Dreamweaver under the name HomeSite+, but has not gone through any major revisions for quite some time.

By Jeffry Houser

The future of Macromedia IDEs lies with Dreamweaver, and if you aren't familiar with it, it's time to start getting up to speed. That is what this article is about. Download an evaluation copy here: www.macromedia.com/cfusion/tdrc/index. cfm?product=dreamweaver.

Setting up the Site

To access many features within Dreamweaver, such as file synchronization, source control integration, and database panel access, you will need to create a site. This is how you do it:

1. **Create the site:** From the site menu, select manage sites. Click the new button, then click site. Or, from the files panel you could select Site --> New Site. The basic wizard. I prefer to use the advanced. Figure 1 shows the advanced tab.
2. **Local info:** On the local info section, you can enter some basic information about the site. You can name the site anything you want. I like to use descriptive names. The local root is the location of the site files on your hard drive. I usually deselect "Refresh Local File Lists Automatically" for performance reasons. You can also specify the default images folder and the HTTP address of the site. The HTTP address is used for the link checker to check links that refer to the site. You can also enable the cache, which speeds up various features.
3. **Remote info:** This section is used to define your remote access to the site. In most cases I have either FTP access or network access. RDS, SourceSafe, and WebDAV are also supported. I will often select the "Upload files to server on save" option. If RDS is an option for you, it will open up some other features in the application panel, such as server bindings and databases.
4. **Testing server:** The testing server section can be used to set the type of server you are testing against (ColdFusion, of course), and how you access it. I'm sure in many cases, the testing server will be your local machine.

5. **Cloaking:** The cloaking section allows you to enable or disable cloaking. Cloaked files or folders will be ignored when doing site operations such as synchronize or a site-wide search.
6. **Design notes:** Design notes are designed to let you share extra information about files with other developers. If you're working in a team these can be useful. Working solo, I will usually disable this feature.
7. **Site map layout:** Allows you to specify how the site map will look. You can view the site map by selecting Map View in the files panel.
8. **File view columns**: This specifies the columns that will be displayed in the expanded version of the files panel. You can expand the files panel by clicking on the expand/collapse button. I usually keep the defaults.
9. **Contribute:** If you want to make your site contribute-friendly, this is the place to do it.

If you don't want to use a site you don't have to; you can always edit files directly in the file system without problems. I've found that sites are a great way to help organize the files and different projects that I have going on at once.

Dealing with Components

One area where Dreamweaver excels is in its handling of ColdFusion Components (CFCs). If you go to the components tab of the application panel, you should see a list of all components on your current server. If you do not have all the proper settings set up in the site, you'll get a list of what you need to correct before components will be displayed.

The component panel also offers a wizard for creating components. Wizards are often much nicer than typing everything by hand. Use these instructions to use the component wizard:

1. **Start the wizard:** Open up the application panel if it is not already open. You can do so by selecting components from the window menu. Click the plus sign (+) on the component panel to bring up the create component wizard.
2. **Enter component information:** Here is where you will enter the attributes of the component tag, such as the display name, the hint, and which component this component is extending. You can also specify the directory that the component will be created in and its file name.
3. **Enter property information:** The properties section is used to specify the cfproperty tag. When using components only within CF, you do not need to specify properties. However, proper-

ties are used in conjunction with Web services, so if you want to open up your CFC as a Web service, then you'll need to enter data here. Click the plus sign to create a property. Type in its name, display name, hint, and type. You can remove a property by selecting it and clicking on the minus sign (-) button.

4. ***Enter function information:*** The function section allows you to specify functions that will be in your component. You can specify the attributes of the cffunction tag, such as name, hint, access, returntype, roles, output, and display name. Using this wizard, there is no place to put your function code. It will only create the shell of the method for you to use. You can always remove a function by selecting it and clicking the minus sign button.

5. ***Enter argument information:*** The arguments section allows you to specify arguments for each individual method. Select the function from the list of available functions, and then click the plus sign to create an argument. Fill in the argument data, such as the name of the argument, its displayname, the hint, type, default value, and whether it is required or not.

6. ***Complete the wizard:*** Click the okay button and the component will be created and opened in the Dreamweaver edit window.

7. ***Fill in your functions:*** Now you can flesh out your functions as needed and fill in any pseudoconstructor initialization code.

You'll probably notice that the component you just created now shows up in the components panel. If you need to use the component inside a page, you can just click and drag the relevant method onto your page and the cfinvoke tag will be created automatically. Since many CFCs maintain state between individual method calls, you will probably want to modify the generated code to reference an instance of an object instead of the CFC directly. Create the instance of the object with the cfobject tag and then change the component value of cfinvoke to the component reference.

You can use the component panel to access Web services in a very similar manner. In the component panel, select Web services from the drop-down menu. Click the add button and enter the Web Service Definition Language (WSDL) location of the Web service. Dreamweaver will parse the WSDL and give you a list of all the methods and their arguments. Just as with components, you can click and drag the method into your cfm page to generate code for calling that component.

Miscellaneous Features

There are a few other features of Dreamweaver that I find helpful in my development. Here they are in no particular order:

1. ***Tag insight:*** If you've ever written code by hand, I'm sure you'll appreciate how great this is. Start typing the name of the tag, and choose it from the list. Hit the space bar and the tag's attributes are listed for you. For many attributes, the default values are even specified. When you close the tag, the end tag, if applicable, is automatically displayed.

2. ***Hide panels:*** You can hit F4 to hide all panels on a screen. Then hit F4 again to replace them on the screen. This is very helpful when you want more screen estate for the code portion of your editor window. To hide the side panel bar without affecting the panels on top or below the editor, click the arrow button in the bar between the panel window and the edit window. Clicking it again restores the panel.
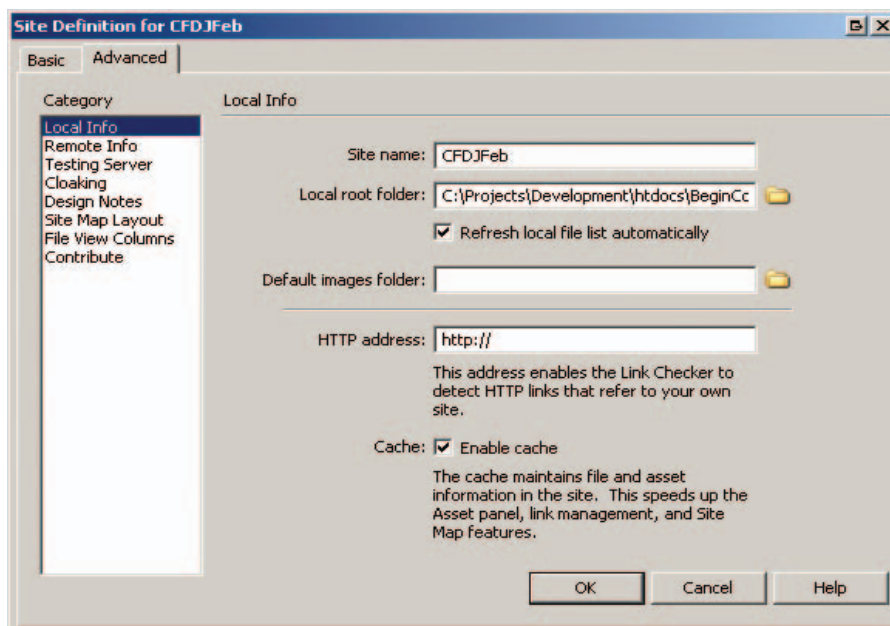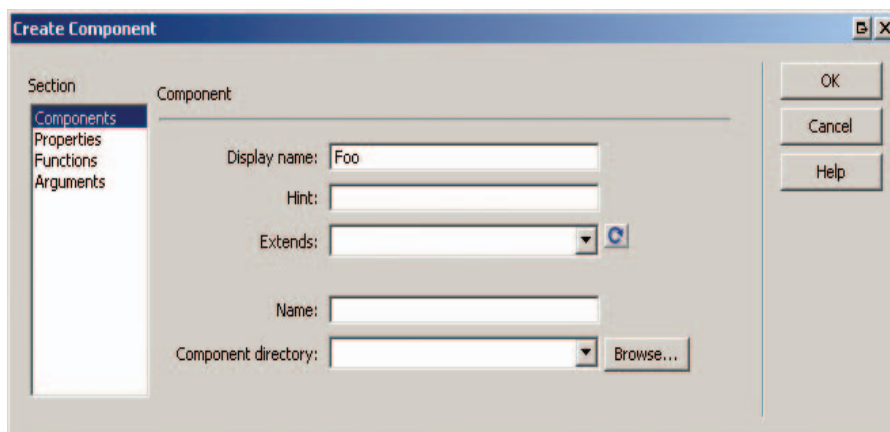

Figure 1: Setting up a site advanced tab

Figure 2: Create component

3. **CFML insert bar:** From the window menu select insert, then the CFML tab. This gives you access to many commonly used CFML language features, including a cfquery wizard, CFML comments, those pesky number signs, cfoutput, cflocation, and cfinclude. Sometimes it's easier to click a button than to type code from scratch. There is plenty of other nonCF-related stuff on the insert bar that can be useful in many situations.

4. **Search and replace:** I wouldn't last a day without using search (and replace) in some manner. Dreamweaver provides many search or replace options. You can search in a current document, within a selected block of text, across all open documents, across all documents in a site, within a folder, or within selected documents of a site. You can also search within source code, text, or within a specific tag. Regular expressions are supported and you can ignore white space. You can also save search information for later use. Control+F is your friend. (Although, I do wish they had a find previous option in addition to find next.)

5. **Synchronize:** In any given week, I'll cycle through multiple projects at more than one client. Since I do all development locally, the first thing I do when I plug into a client's network is synchronize my local site with the most up to date version of their code. The synchronize process deletes any files on my machine that are no longer relevant and it also updates old versions of files. When I'm done with my development for the day, I can always synchronize right back to them.

6. **Snippets:** Snippets are small blocks of code that you can just click and drag onto your own page. You'll find them in the code panel on the snippet tab; you can also create your own. Some of my own custom snippets include a comment header that I place on every document, a dump of application, and session variables.

Conclusions

I'm sure you'll find that Dreamweaver is a very powerful tool that keeps getting better. If you find it lacking in some area, there are many extensions available to help you expand it. A good place to start looking for extensions is in Macromedia's Developer Exchange at www.macromedia.com/cfusion/exchange/index.cfm#loc=en_us&view=sn100&viewName=Macromedia%20Exchange&avm=1.

I hope you've found this article enlightening. Let me know which features in Dreamweaver that I didn't cover here help your development.

About the Author

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

jeff@instantcoldfusion.com

Out-Moding Modular Presentation with xml

Fortunately, with the HTML library's XML syntax, many of these issues are handled automatically. By default, forms are populated with data from the attributes scope. Values from the database can be easily used by setting defaults in the attributes scope with the CFPARAM tag. Thus, much of the work of creating forms is eliminated. Consider Listings 2 and 3, both of which produce the same display. You might notice that, although the second example requires an XML namespace and two new attributes in the select tag, there is no need to replace the HTML input tag with either CFINPUT or a custom tag. This is because the HTML custom tag around the second example uses a series of functions to convert the structured XML document into comparable HTML elements. In the process, it is able to set defaults for commonly forgotten attributes. Or, it may convert deprecated nodes or attributes into a newer format. The CENTER tag, for instance, is converted into a DIV tag with the in-line CSS text-align property set to "center," without any effort on the part of the developer. Because this task is automated by the onTap framework's XML syntax for HTML library presentation, once an application has been converted to use this syntax it becomes very easy to migrate away from deprecated presentation code.

Beyond avoiding deprecated code, this structure and automation also makes presentation for multiple mediums quite simple. For instance, it becomes very easy to send multipart e-mail in both HTML and plaintext formats using the same presentation components. You can use the html.show() function to display an HTML version and the html.showText() function to display the text part of the e-mail.

Although nothing has been implemented at the time of this writing, there are also plans to use the HTML library structure to provide a convenient means of creating G11N globalized applications. In theory, your XHTML templates would be populated with the names of variables in your localized resource bundles and then, at time of display, converted to the appropriate localized values by the html.show() or html.showText() function.

The Last Word

As the author of the onTap framework, I have a bias towards the use of the HTML library. Hopefully this article has helped you understand why I believe so strongly in it. I also hope to have shown you a few of the ways it can help you manage your application's presentation and provide a rich user experience. Feel free to contact me if you have any questions or comment.

About the Author

Isaac Dealey has worked with ColdFusion since 1997 with clients such as MCI and AT&T Wireless. He evangelizes ColdFusion as a volunteer member of Team Macromedia, and is available for speaking engagements.

info@turnkey.to

CFDJ Advertiser Index

Don't Miss CFDJ's Next Issue!

YOUR NEXT ISSUE!

Focus on architecture

Advanced deployment with ant (and dbunit)
Put yourself in control of your development process

Coldfusion everywhere
Part 2 – run any application on the desktop

Linux and bluedragon
For low-cost cfml happiness

Thehub
Application framework and development methodology

Bearing Fruit with Plum

Review of the Plum IDE from Productivity Enhancement, Inc.

P lum stands for Practical Lightweight Universal Methodology, and consists of a comprehensive development environment that includes a Windows-based IDE; code generator; robust application framework; development methodology; simple integrated Content Management System (CMS); and stored procedure, unit test, and component generators. Plum accomplishes so much for a ColdFusion developer that it can literally be overwhelming trying to grok it all at first glance.

By Jeff Fleitz

System Requirements and Installation

CFMX 6.1 is required to run Plum applications as the framework makes extensive use of ColdFusion Components (CFCs). The Plum IDE is a Windows .NET desktop application, so have the Microsoft .NET Framework 1.1 installed on your Windows 2000/XP computer prior to installation. Though the IDE is a Windows application, Plum-generated code can be deployed on any CFMX platform.

Plum was designed to work primarily with MS SQL Server 2000 (the stored procedure generator only supports this option), but will also support Oracle 9i and Microsoft Access 2000. If you are using Oracle you will need to download the Oracle .NET data provider to ensure that the IDE can read the database schema. (A future version of Plum is slated to support popular open source databases such as MySQL and PostgresSQL.)

Creating a Project
The New Project Wizard

Plum includes a new project wizard that steps you through the process of creating a fully functional baseline Web application in about two to five minutes.

The first steps collect and verify information about the operating environment, including application name, directory mappings, and server operating system. Once this information is collected the Plum Framework and starter content are installed.

The next several steps of the wizard verify a connection to your database and ensure that the mandatory Plum tables are present. Plum projects require the presence of five tables that support the Content Management and User Access modules of the application. If the tables are not present, they will be inserted, fully respecting the integrity of the objects already in the database. The remaining pages in the wizard collect and verify mappings for server, custom tag, and Verity collections, and allow you to optionally choose for which tables to generate standard maintenance forms.

At the completion of the project wizard, Plum launches your default browser and runs an initialization routine to verify your environment is set up correctly. At this point you can also optionally create Verity collections for your CMS starter content. When initialization is successfully completed, you are redirected to the home page of your new Web site, complete with starter content! Returning to the Plum IDE reveals a new project tree containing all the major components of your project.

The Generated Application

The application generated by the wizard is a fully functional Web site, with both public and private (administrative) areas. It contains many useful public-facing general purpose forms that handle tasks such as logging in, user self-registration, user profile maintenance, password retrieval, feedback, e-mailing a copy of the currently viewed page to a specific address, and a site search form. Starter content is provided for basic pages common to many Web sites, which can be easily edited through the internal CMS. The administrative modules provide built-in forms for maintaining users/roles, content, and layouts.

XML Project File

The resulting Plum project file is a fully compliant XML file with a .plum extension. Over time I expect Plum adopters will think of many creative ways to extend the use of this file using XLST. A note of caution – you should avoid manually editing the project file or you risk having it become unreadable by the

IDE. You might even want to consider including it in your source control provider, if you are using one.

Working with the IDE

The main purpose of the Plum IDE is to synchronize your project with the schema of your database and rapidly generate code for your application. There is no WYSIWYG design surface or internal code editor; you will still be using your favorite code editor to manage code. The preferred editor is HomeSite+, and custom graphical tag editors are included to work with the generated code. However, it is relatively easy to re-create these tags in a custom tag library in Dreamweaver MX, if that is your preferred editing tool.

Round-trip editing is not supported; any manual changes made to the generated code are not subsequently reflected in the Plum project file. If you make custom changes to tag attributes in the code and you want to keep your project file up to date, you will have to manually update the attributes in the IDE. In fact, there comes a point in every Plum project when the generated code is "divorced" from the IDE, after which you might add your own handcrafted code. This is a necessary and intended behavior to allow you to extend the application with your own customizations.

The Workspace

The IDE workspace is split into three frames with the project tree in the left frame, form data entry in the top right frame, and help information for the selected node in the bottom right frame (see Figure 1).

The project tree contains four nodes that focus on the major code generation areas: environment, database, modules, and security. Clicking on a node in the tree activates the associated data input form for that node in the top right frame. The database and modules nodes and their children make liberal use of context menus, so it is a good idea to right-click on these nodes as you get started to view the available command options.

The environment node extends much of the required information gathered by the new project wizard, i.e., application name, directory mappings, data source connection, etc., by allowing you to specify variable settings for the SMTP mail server, Webmaster e-mail address, persistence preferences (session and/or cookie), minimum acceptable browser versions, enabling of error handlers, and whether or not to allow visitors to create their own accounts when visiting your site. All of the settings specified in this node are optionally written to the Environment.cfm file in the root of the application directory during code generation.

The database node expands downward two additional levels to include nodes for each of the tables in the database, and for the column and relationship attributes in each of the tables. Each table node allows you to specify the primary and alternate keys for each table as well as default singular and plural labels for framework references to the table in the application. Column nodes provide a form for specifying column labels, data type, default values, enumerated values, formatting functions, whether nulls are allowed, and finally, whether the value is issued by the database server or supplied by business logic. In the two latter instances, the column values can be displayed on forms, but input controls are not generated to prevent user input. Relations can also be created, but the database synchronization process handles this for you automatically. Relationships are already handled for you in most instances when Plum reads the database schema, but you can add or change relationships manually from the database node context menu. The settings in the database node are referenced by the IDE when building data pages.

The modules node is where you will spend most of your time in the IDE, building the forms for your application. Modules consist of pages (form containers) that are related by functionality, or similar security access restrictions. A page can contain multiple form types if desired.
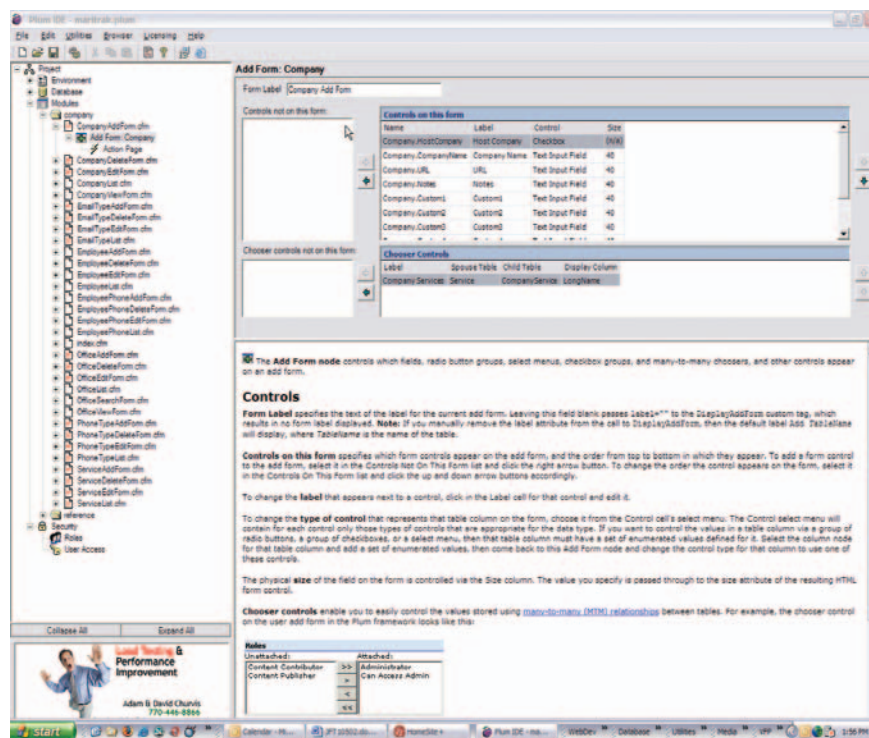
Figure 1: IDE workspace

Figure 2: Tag editor

For instance, you can create a list form that also has an embedded search form on it, or an edit form which has a child list form on it (called a master-detail form), complete with built-in paging controls. Certain combinations are not allowed; you obviously can't have a child list on an add form, for instance. Pages can either be created individually, or you can have Plum generate standard or master/detail form sets for optionally selected tables at one time. It is quite a sight to see Plum generate all of the data maintenance forms for a module in just a few seconds! When you select a form on a page, a builder is presented in the top right frame that lets you choose the columns to be included in the form, as well as the control type and sort order for each column.

Building forms is an iterative process. Once you have a form built the way you think you want it, you generate the code for it and view it in your browser, come back into the IDE and tweak it some more, etc. At some point you may want to start placing your own handcrafted code into the page, or use one of Plum's custom tags that is not available in the IDE. This is the point at which you need to "divorce" the page from the IDE. Individual pages or entire modules can be locked down to avoid accidentally

overwriting customizations.

The security node is where you create application roles and then assign users and modules to the role. Plum then generates the authentication code for each module. Roles are table driven, but not handled for you automatically. You will have to go into the User module on the admin side of your application and add each role code manually to the database.

Working Outside of the IDE

Plum's IDE generates code pages that leverage the extensive custom tag library included with the Plum Framework. Using the provided custom tag editors, it is very easy to create Plum pages and modules manually. The trick is to learn the proper nesting hierarchy of the custom tag calls. Once you master this, you can create pages very rapidly just using your code editor.

This comes in very handy when rapidly prototyping Plum applications that don't yet have a data model defined. The Plum Framework includes a component that creates sample data either programmatically or by retrieving data from text files. You can use this component to generate sample data in two ways: by using a QueryStub, which returns a query object based on passed data type

parameters, or by using the sampleData attribute of the form and list-related tags. Prototyping in this fashion is like wireframing on steroids because all of the prototype code can be leveraged by simply changing the query calls.

Utilities

You can resynchronize your project with the database at any time by selecting an option from the Utilities menu, and the database tree nodes will be refreshed to reflect the changes to the data structure. One, many, or all of the tables can be imported into the project at one time. If an existing table structure has changed, you are notified with a dialog and given the option to accept or decline the changes to your project. Tables, columns, relationships, and their associated attributes can all be manually added to the tree as well.

If you are using MS SQL Server 2000 as your database server, you are in for a treat because you can have the Plum IDE create a complete set of insert, update, delete, list, and get stored procedures for you, one script for each designated table. These scripts are placed in the project's test directory, and can either be used as or as a starting point for further customization. While the stored procedures generated currently support single table operations only, and therefore won't work as is with some of the more complex Plum form types, this is still a very useful tool and an impressive feature for a 1.0 release.

You can further utilize the generated stored procedure scripts by having them converted to CFCs (ColdFusion Components) using the component generator. This tool encapsulates all of the stored procedures as methods calls in a component of the same name as the script, with a .cfc extension.

Unit testing is a popular practice in the OOP world, and involves designing and writing test cases which validate that class methods function properly and return valid outputs. The same principles can be applied to CFCs, although it can be a time-consuming and tedious process to write them yourself. The Plum IDE contains an internal facility that enables you to easily create and execute unit tests against your CFC methods. A series of dialogs allows you to visually

Home Help
[-] App Server
ColdFusion Admin
JRun MC
Restart
System Logs
Heartbeat
Configure

[-] Site Management
Domains
Emails
Horde Web Mail
File Manager
Site Builder
Web Counter
AWStats
Web Logs

[+] Database
[+] Apache
[+] Applications
[+] Advanced
more...

testmycfm DNS,Domain Host, and Email Options:

Domain Host and DNS Configuration   Email Configuration

:: Domain Host and DNS Configuration ::
Please refer to this topic in the Help section before proceeding.
DOMAIN
Select Domain: (1) 1 ▾
Dedicated Address: [testmycfm.webappcabaret.net]   Server Address: [pointer.webappcabaret.net]

Domain Name (mydomain.com) ❓: mycoolcfm.com   [Set Domain]
                                    IP/Host Address: testmycfm.webappcabaret.net

Apache Document Root ❓:   /usr/ngasi/contexts/testmycfm/testmycfm
Apache Directory Index: index.html index.jsp index.do index.html.var index.
AppServer Virtual Path(s) ❓: /*.cfm /CFIDE/
                             [Customize Apache Virtual Host]

DNS & SUB-DOMAINS
DNS Service ❓: false [false ▾] [Set DNS]

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

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

Email Configuration

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

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

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

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

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

NGASI POWERED

build the tests, and the results are graphically displayed in your browser, along with the ability to drill down into the details of the result. This tool is useful if you plan on integrating additional components into the application.

The Framework

The Plum Framework comes with a core set of components, custom tags, and other supporting files that handle most of the heavy lifting required by all ColdFusion applications, including role-based security, data access and validation, structured exception handling, client/session management, browser detection, and layout management. An extensive, unencrypted custom tag library is provided to access the component methods. In fact, all of the code generated by the IDE consists of nothing more than custom tag calls (see Listing 1). There are so many custom tags and associated attributes available that using the provided tag editors is a must until you have spent some serious time using them (see Figure 2).

Forms

Plum's intelligent forms are feature rich and provide you with many options from which to choose. You can build forms that handle single table maintenance, parent-child, and/or many-to-many relationships, all on the same page if you prefer. You can create a stand-alone search form or a combination search/list form on one page. All of the common HTML form control types are supported and some of them have enhanced functionality. One of my favorites is the Filtered Select menu, which can either fully populate a Select menu from a parent lookup, or filter it (if the query results exceed a certain threshold) by collecting criteria from the user in a small text field to the left of the menu and then pressing Tab. This is very useful when dealing with parent lookups with very large tables. Both client-side and server-side form validation are automatically handled for you; the former uses an extended version of the popular qForms JavaScript library.

Content Management System

Plum integrates a rudimentary, but effective CMS into your projects. Most of the starter content generated for a new project is contained within the CMS to provide examples of how it works. As of this writing you could choose between HTMLArea and ActivEdit as the WYSIWYG editor, although efforts are already underway to adopt other commonly used editors that support other popular browsers like Firefox. Multipage content can be easily created with paging controls added to each page. Simple workflow management features are also provided.

Automated Verity Management

Forms are provided for you to automate the entire Verity collection creation, maintenance, and indexing operations. You can create multiple content collections and offer them independently for users to search through.

Clean Separation of Data, Logic, and Presentation

While not strictly adhering to the Model-View-Controller (MVC) design pattern popular in some other ColdFusion frameworks, Plum provides a nice and easily understood separation of the data, logic, and presentation tiers of your application. Plum's straightforward directory layout and file naming conventions make it easy to find and debug code.

Structured Exception Handling

Plum provides very robust structured exception handling with predefined enumerated exceptions that can be turned on or off by checking one box in the IDE (or by changing the associated variable in the Environment.cfm file). When turned on, Plum dumps the exception specifics into an e-mail notification to the Webmaster. You will want to turn the exception handler off when you are debugging pages so that you can view the raw error and avoid having your mailbox overrun with exception reports.

Security Model

You can easily require authentication to the module, page, or page section levels of your application with a single call to the cf_Authorize custom tag. The following tag call placed in the Application.cfm file for a module would authenticate a user against the ADMIN or PUBLISHER roles.

```
<cf_Authorize
  roles="ADMIN,PUBLISHER"
userMustBeAMemberOf="AnyOfTheseRoles">
```

Page sections can be restricted by using the cf_Authorize paired tag:

```
<cf_Authorize
  roles="ADMIN,PUBLISHER"
userMustBeAMemberOf="AnyOfTheseRoles">

You will see me only if you're an ADMIN or a
PUBLISHER.

</cf_Authorize>
```

The Plum Methodology

The Plum methodology takes a slightly different slant on system development that espouses quickness and simplicity. The Plum methodology is tightly coupled with the framework and is meant to help streamline Plum development with what is basically a 10-step process:

1. Chart the business processes your application will support
2. Derive the navigational interface from the business process charts
3. Design the site content search structure
4. Develop the graphical layouts
5. Create the database
6. Generate and tweak the administrative application
7. Create the prototype pages
8. Write the business logic
9. Convert prototypes to the finished application
10. Test and deploy

One of the interesting central themes of the Plum methodology is that documentation should be lightweight and code commenting should be kept to a minimum, arguing that it is more preferable to make the code as self-explanatory as possible and relegate extensive comments to the source control provider. This philosophy is totally counter to methodologies like FLiP, which promote documenting the functionality of the code before it is written. It's obviously up to you whether you adopt the Plum Methodology or not. While it is not imperative that the Plum Methodology be used with Plum, it is worth reading about it before making your decision.

Deployment

In dedicated host situations, the difference between a Plum application in a staging environment versus a production environment is the settings in the Environment.cfm file. So you should be able to upload your directory structure from the staging site to the production site, replacing the staging Environment.cfm file with one that has the correct values for the production server.

However, if you deploy the application to a shared host, there is a potential issue to contend with that involves custom tag mappings. CFMX cannot formally bind an instance of a custom tag to a particular application, so having multiple copies of the same tag name referenced in multiple custom tag paths results in CFMX not knowing which instance belongs with which application. To address this issue, Productivity Enhancement has provided a free utility called CF_2CFModule, which will convert all custom tag calls to CFMODULE="template" syntax, allowing the variable path reference to remain. To use CF_2CFMODULE, you simply drag and drop the root Plum project directory onto the CF_2CFMODULE.EXE icon, (after making a backup of course). You then deploy the converted directory.

Cut to the Chase

Plum could be the Swiss Army Knife of the ColdFusion world. It does just about everything you need, and does it well…for free. That's right! Plum can be used for free if you don't mind looking at one small Productivity Enhancement advertisement in the bottom left corner of the IDE. If you adopt it, you will want to do yourself a favor and pay the $79.95 for a licensed version so you can sleep better at night.

While there are some areas that need improvement, it's an awesome product for a first release. Unfortunately for some, Plum performs most of its magic on Windows. If you develop on Windows, you owe it to yourself to check it out. If you are developing on another platform, you will have to pray for Mono.

For a complete list of Plum offerings, visit www.productivityenhancement.com/plum/WhatPlumCanDo.cfm.

About the Author

Jeff Fleitz has been developing Web applications for over eight years and using ColdFusion since version 2. He is currently the director of Information Solutions for KR Systems, Inc., where he continues to develop applications for the federal government and maritime shipping industries. He has been involved with Plum since the first private beta.

jfleitz@krsystems.com

Listing 1: An Example of Plum's Tag-Based Form Code

```
<cfmodule template="#Request.adminLayout#/Header.cfm" pageType="Database">

<cf_DisplayAddForm
    table="Company"
    label="Company Add Form"
    action="#Application.URLRoot#/admin/company/CompanyAddAction.cfm"
    displayALinkToThisList="Yes"
    focusFirstField="No">

  <cf_PrimaryKey
      primaryKeyColumn="CompanyID"
      primaryKeyType="integer">
```

```
<cf_DisplayTextInput
    column="CompanyName"
    label="Company Name:"
    size="40"
    required="Yes">

<cf_DisplayTextInput
    column="URL"
    label="Website:"
    size="40"
    required="No">

<cf_DisplayTextInput
    column="Notes"
    label="Notes:"
    size="40"
    type="text"
    required="No">

<cf_DisplayChooser
  spouseTable="Service"
    childTable="CompanyService"
    spouseTablePrimaryKey="ServiceID"
    spouseTablePrimaryKeyType="integer"
    spouseTableDisplayColumn="LongName"
    foreignKeyFromChildTableToMainTable="CompanyID"
    foreignKeyFromChildTableToSpouseTable="ServiceID"
    chooserLabel="Company Services">

<cf_DisplaySubmitButton
    buttonText="Save">

</cf_DisplayAddForm>

<cfmodule template="#Request.adminLayout#/Footer.cfm">
```

Search Terms Using a Reporting CFC

What Are They Looking For?

By James Edmunds

I f I ran a candle shop and my visitors began to ask for blue-and-white-striped candles, you can bet I'd move whatever stock of blue-and-white-stripers I had into a prominent position, and I'd be sure that I had a good inventory of blue-and-white-striped candles on hand and on order. I would be avidly reading CSDJ (*Candle Store Developer's Journal*) every month, of course, and keeping abreast of market trends, for who would be fool enough to ignore what the customers themselves were asking for?

We all do the same thing with our Web sites, whether they exist for e-commerce or to provide information about a business or organization and its services and activities. I'm sure most Webmasters spend as much time as I do poring over logs and referrer reports, paying particular attention to which search terms or keywords visitors have used on Google and other search engines to arrive at their sites. The sites that I manage vary in size and in traffic, but are all of a scale such that they happily reside on the hosting company servers. I have about a dozen sites at CrystalTech, one of several hosting companies that pay particular attention to accommodating ColdFusion programmers. Such companies typically also provide a robust statistical reporting package as well as access to raw logs that allow you to do as much rooting around as you wish.

On many of my sites, there are areas with internal search facilities, employing either SQL-match searching, or, more likely, the Verity search features that are offered with ColdFusion MX. For these sites, I have created a ColdFusion Component (CFC) that collects the search terms used by visitors in my Verity search forms and generates a report of which search terms have been used, either upon request or on an automated schedule.

A Word on Verity

You'll note that I've titled the CFC discussed in this article "verityreport.cfc," even though there is nothing about the component that is inherently tied to Verity. Indeed, the component could be used entirely to track SQL searches. What my title betrays, I think,

is how much I value Verity and the extent to which, when I think "search" in a ColdFusion application, I am quite likely thinking of employing Verity. That's because the features of Verity that are included with ColdFusion MX provide a range of powerful search options for CF Web sites. Developers can create index collections defined by URL paths in a domain, or defined by one or more fields in a database; searches can be made against these indices, either in isolation or in combination. The collections contain fields that allow parameters to be passed to the search results. The parameters then allow those results to display meaningful titles or summaries, as well as to include links to detail templates based on a primary key or other identifier. Plus, the search algorithm provides site visitors with an interface that is somewhat similar to what they experience in popular search engines, without requiring the programmer to supply the search logic, as would be the case with a masked SQL search.

That's already pretty inviting, but we're only describing the basics of Verity's powers in the current version of ColdFusion MX (version 6.1). The little that has been publicly discussed about what Verity may add in Blackstone is very intriguing. Macromedia ColdFusion product manager Tim Buntel's Weblog (www.buntel.com/blog/index.cfm) has suggested possibilities such as the ability to have categorized and context-driven searches and results, the ability to search within the context of a prior search, more customization, spelling suggestions for searches, and, generally, more intelligent searching.

This means that, using Blackstone, CF developers will likely be able to offer site-internal searches that are reasonably equivalent to popular search engines, which have a level of sophistication that site visitors have now come to expect. CF's powerful Verity search tool will become even more powerful and useful for site visitors.

First Things First

I mentioned that my Web site workload revolves around smaller-sized sites that reside on hosting company servers. ColdFusion Components are just as valuable for code reuse in that context as they are for developers serving multiple sites that call CFCs stored in a single directory. Write the CFC once, and drop it into the appropriately mapped directory of any hosted site that needs it. (By the way, you can get a good grounding in the principles of CFCs from the excellent series of articles Jeff Houser has recently been writing for *CFDJ*. Highly recommended!)

When I began writing this CFC, which I call "verityreport.cfc" (see Listing 1 – all listings for this article can be downloaded from www.sys.con.com/coldfusion/sourcec.cfm). I wanted a component that would make sense for a variety of my Web sites with search facilities. The basic functions would be (1) to collect information about which search terms site visitors used and (2) to report what those search terms were, for a given period of time. I also wanted to be able to use the CFC without much additional programming, particularly for smaller sites that might or might not have a client's administrative back end or content management system (CMS). I decided to add a presentation method that would generate and e-mail a weekly report to the appropriate parties, as well as provide ready-made reporting when a client's administrative interface does actually exist.

The first step was to create a database table to store the search terms entered by visitors (note that throughout this article, and the accompanying code listings, the database calls are for MySQL;

with little or no modification, they can be made to serve other databases). I created both a field for the search term and a date/time field for the date and time when the search term was entered by a user, so that I could report the terms according to a given set of date parameters. Also, I added an auto-incremented primary key field.

I call the table "tbl_searchterms," and the three fields are "searchterm_id," "thesearchterm," and "thesearchdate."

The first method of the CFC, named "collect SearchTerm," is the one called to write a search term and the time it was called to the database. But, before we look at that, let's take a peek at my typical Application.cfm template, which sets some of the values for arguments that will be called in the CFC. The more I use custom tags and CFCs, the more each of my sites' Application.cfm templates is likely to have parameter-setting lines like this:

```
<CFSET request.dsn = "mydsn">
<CFSET request.dsnusername = "mydsnusername">
<CFSET request.dsnpassword = "mydsnpassword">
<CFSET request.mailto = "someaddress@mysite.com">
<CFSET request.mailfrom = "someaddress@mysite.com">
<CFSET request.thissite= "mysite.com">
<CFSET request.mailserver = "mail.mysite.com">
```

The "collectSearchTerms" method uses those request variables to call the database for the insert of the search term and the time it was called into the database and uses the variable passed in when the method was called to populate the field "thesearchterm." The date, as you can see, is drawn from the component's one line of constructor code (constructor code in a component is the code that precedes any of the individual methods' code, and may be used by one or more of the methods of the component).

This recording method is called with the following code, placed at the point in the site's search facility where the visitor's search term is collected for use against the collection (or database, if it is an SQL search rather than a Verity collection search that is being executed). The following code will typically be found on the action page that lists search results.

```
<CFOBJECT
component="mycfcmapping.verityreport"
name="verityreportcfc">
```
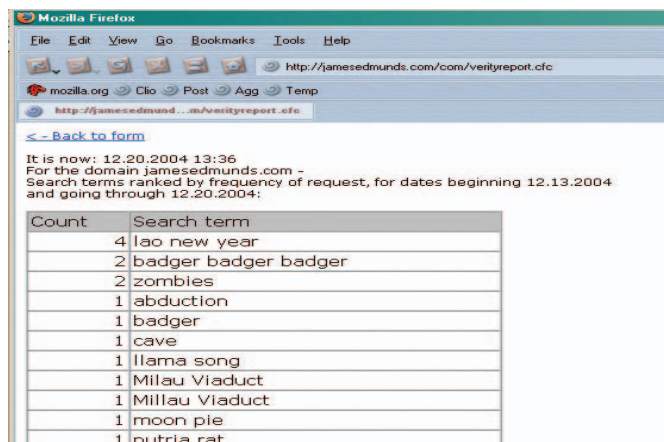

Figure 1: Verity Report CFC screenshot

```
<CFINVOKE
        component="#verityreportcfc#"
        method="collectSearchTerm">
        <CFINVOKEARGUMENT
                name="thesearchterm"
                value="#session.keyword#">
        </CFINVOKE>
```

The variable "thesearchterm" can be derived from a session variable, as in this example, or from a client or other variable, as dictated by the particular methodology you use to produce a page of search results.

This simple function of the component, invoked from whatever points your site visitors are entering their search terms into, creates a database of search terms associated with the times at which they were entered.

Returning the Data

The second method, "getSearchTerms," gathers data according to the two parameters that set the date range we want our report to encompass. This method also supplies default values, drawn from the current time that we set in the constructor code, and, once again, utilizes the datasource request variables we set in the Applicaton.cfm template.

The query that is returned by this method limits the search by start and end dates (defaulting to the most recent week if no arguments are passed in for those values), and orders instances

---

> "We don't have to wonder what visitors to Web sites are looking for: they will gladly tell us. Let's make sure we're paying attention! This CFC helps to keep track of their desires."

---

of "thesearchterm," first by the highest frequency of occurrence and then alphabetically.

It would be a simple matter to call this method, passing in dates if we wished to use a span other than the default and displaying the query results in whatever way we wished. For the manner in which I wanted to use this component, however, I didn't want to have to create any additional code in order for the component to serve its most basic use. Therefore, I created a presentation method that could run without any additional parameters having to be passed in, or could be used in a bit of a fancier way if it were to be included in an administrative back end used by a client.

The third method, "displayMailReport," can be run with this simple invocation (most likely run automatically through CFSCHEDULE or some other server scheduler set to run it once a week):

```
<><CFOBJECT
component="mycfcmapping.verityreport"
name="verityreportcfc">
<CFINVOKE
```

```
component="#verityreportcfc#"
method="displaymailreport">
```

The "displayMailReport" method called in this manner will e-mail the ordered report once a week to the e-mail address specified as "request.mailto" in the Application.cfm template. Or, it may be called with a form that asks for more detail, allowing for a user-specified date range as well as one or more user-specified e-mail recipients. Note one very handy feature of a method that invokes another method within a component: you can pass all the arguments wholesale from the calling method to the called method with this single parameter:

```
argumentcollection="#ARGUMENTS#".
```

Listing 2 provides an example of calling the "displayMailReport" method through a form. A screenshot of the report it can generate – in this case showing some recent search terms used on my blog, poorlcio.com – is shown in Figure 1.

It Could Be So Much Better!

Since I created this CFC, I have found it very useful to receive the weekly generated e-mails with the reports of search terms used by my various sites' visitors. I have also found it handy to use the form version in Listing 2 to generate screen reports or to send e-mail reports to myself or others.

But, like a lot of you, the second I make a mental note that something is done, I immediately start thinking of all the ways I could improve it. A future version of this CFC and its implementation would probably expand the database to include a field identifying which template in a site with multiple search blanks had generated this particular search, and to return that information as part of the reporting. Also, I'm already tinkering with a related CFC that will record and report which database records have been the most frequently called into template pages that are populated from databases – nothing to do with Verity or searches, but certainly information that tells us what our site visitors have been looking for and utilizing.

When we get an even more enriched version of Verity with Blackstone, I'll no doubt be taking another look at this CFC to mine the value of the new Verity features – not only for site visitors when they explore our sites, but also for us developers as we observe their paths through our sites. If this leads me to add a Verity-specific feature or two to my CFC , it would finally live up to its name.

About the Author

James Edmunds is a freelance Internet developer and arts administration consultant living in New Iberia, Louisiana. After a career in journalism that included writing for national publications such as Newsweek *and serving as editor for an alternative weekly newspaper he founded in southern Louisiana, James began to pursue a second career working with arts groups. His interest in harnessing the power of the Internet to serve the interests of the arts led him into Internet development.*

edmundsj@bellsouth.net

web services EDGE conference&expo

Web Services Edge
2005 East

February 15–17, 2005
John B. Hynes Convention Center
Boston, MA

The Largest *i*-Technology Event of the Year!

Who Should Attend

- Software Developers
- Software Engineers
- Application Developers
- Development Managers
- Technical Directors
- Analysts/Programmers
- IT Managers
- CEOs
- CTOs
- CIOs
- Technical Architects
- Team Leaders
- Software Consultants
- IT Directors
- Project Managers

Features & Attractions

- **3 Days Packed with Education and Training**
- **Keynotes & Panel Discussions from Industry Leaders**
- **50+ Hard-hitting and Informative Seminars**
- **FREE Web Services Tutorial Presented by Novell**
- **FREE .NET Tutorial Presented by Microsoft**
- **FREE Web Services Security Tutorial Presented by Critical Sites**
- **Cabana Night Developer Exchange Presented by INETA**
- **Panel Discussions Presented by JCP**
- **Opening Night Welcome Reception**
- **Compelling Case Studies & Best Practices**
- **Hands-On Labs Featuring .NET and Visual Studio Presented by Microsoft Partners**
- **Featured Product Demonstrations**
- **Exhibit Floor featuring hundreds of products**
- **Real-time SYS-CON Radio Interviews**

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

ices Conference & Expo

Keynote Speakers ▶

Tuesday, February 15, 11 a.m.
Matt Ackley
SENIOR DIRECTOR, eBAY
DEVELOPERS PROGRAM

Web Services for eCommerce

eBay, The World's Online Marketplace, has more than 114 million registered users, 10,000 developers, and over 700 live, third-party applications. Four years ago, eBay began allowing third parties to build applications that tap into eBay, and today eBay hosts one of the leading Web services platforms. Through its developer program, eBay enables third parties to create cutting-edge Web services applications that benefit the buyers and sellers on eBay. At present, 40% of eBay's listings come through its API, which handles more than a billion Web services calls a month. Ackley will discuss the rewards and challenges of building and maintaining one of the world's leading Web services platforms, and share insights and practical guidelines for others.

Matt Ackley is senior director of the eBay Developers Program. He supports eBay's vision to be the leading platform for global online commerce, and is chartered with creating a thriving ecosystem between eBay, its community of users, and third-party developers and solution providers. Ackley joined eBay in 2003 as part of eBay's acquisition of FairMarket, which provided technology solutions and services to online marketplaces.

Wednesday, February 16, 11 a.m.
Ari Bixhorn
DIRECTOR, WEB SERVICES STRATEGIES,
MICROSOFT CORPORATION

Introducing Indigo: The Unified Programming Model for Building Service-Oriented Applications

Indigo is Microsoft's unified programming model for building service-oriented applications on the Windows platform. It enables developers to build secure, reliable, transacted solutions that integrate across platforms and interoperate with existing investments. Indigo combines and extends the capabilities of existing distributed application technologies, including .NET Enterprise Services, System.Messaging, Remoting, ASMX, and WSE to deliver a unified development experience spanning distance, topologies, hosting models, protocols, and security models. This keynote will provide an inside look at Indigo and show you how Indigo will radically simplify the development of distributed, service-oriented applications.

Ari Bixhorn is the director of Web Services Strategy in the Developer and Platform Division at Microsoft Corp. He is responsible for product planning and technical evangelism for Microsoft's Web services offerings, including "Indigo," the code name for a component of the next version of the Windows operating system, code-named Windows "Longhorn." Bixhorn has spent the past five years at Microsoft, driving product management efforts for the Visual Basic and Visual Studio development systems.

Thursday, February 17, 11 a.m.
Mike Milinkovich
ECLIPSE.ORG

An Open Development Platform for Web Services

Open source technology runs the Internet. Linux, Apache, PHP and Eclipse are highly successful open source communities that provide the backbone for today's Web applications. All indications point to a continued value proposition for organizations for leveraging open source when developing and deploying SOA-based applications. This keynote will examine the benefits of using open source technologies, the decision-making process used when adopting these solutions and the potential for contributing back to the open source community.

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

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

DAY 1 — FEBRUARY 15

		Java	.NET	Web Services	
7:30	Registration				
8:00	**FREE Tutorial** – Ashish Larivee, Novell, **Using a Web Services Framework to Build SOA Applications**				
9:00		(J-1) What's New In JDO 2.0	(.NET-1) Intro to SPOT	(WS-1) Ensuring Web Services Interoperability	
10:00		(J-2) Using Java Messaging in Real-Time Trading Systems	(.NET-2) An Introduction to SQL Server Reporting Services	(WS-2) Web Services Standards: Going Behind the Mask	
11:00	**Opening Keynote** – Matt Ackley, Senior Director, eBay Developer Program, eBay				
12:00	**EXPO OPEN** (12 P.M.–5 P.M.)				
3:00	**Keynote Panel** Presented by JCP – **Web Services and Security** Moderator: Onno Kluyt, Sr Director & Chair, JCP Program, Sun Microsystems			(WS-2B) Solving Complex Business Problems Though SOA	
4:00		(J-3) The ROI of a Java-Rich Client	(.NET-3) Go With The Flow – Human Workflow Services in BizTalk 2004 / (.NET-3B) Techniques with Visual Basic.NET	(WS-3) The XML Data Challenge	
5:00	Opening Night Reception				

DAY 2 — FEBRUARY 16

		Java	.NET	Web Services	
7:30	Registration				
8:00	**FREE Tutorial –** Thom Robbins, Microsoft – **The Next Generation of Visual Studio** (free with VIP preregistration)				
9:00		(J-4) Web Services End-to-End Security on J2EE: Gaps and Proposed Solutions	(.NET-4) The Microsoft Framework: An Agile Software Development Process for Building Web Service Applications	(WS-4) How To Bulletproof Your Web Services	
10:00		(J-5) J2ME and Eclipse	(.NET-5) Web Services Security for Dummies with WSE2	(WS-5) The Role of Policy in Web Services Integration – It's More Than Just Security	
11:00	**Keynote** – Ari Bixhorn, Director, Web Services Stategies, Microsoft Corporation				
12:00	**EXPO OPEN** (12 P.M.–4 P.M.)				
3:00	**Application Server Shootout**				
4:00		(J-6) The Impact of JBoss and Mono on the Application Server Market	(.NET-6) Securing Service-Oriented Architecture with Microsoft's WSE 2.0 / (.NET-6B) J2EE to .NET Interoperability and App. Integration	(WS-6) B2B Policy Enforcement: The Third Rail of SOA Implementation	
5:00		(J-7) Migrating Enterprise Applications Between J2EE Application Servers	(.NET-7) So You THINK You Know What an Object Is… / (.NET-7B) Building and Using Advanced ASP.NET Web Controls	(WS-7) Driving SOA Governance	
6:00	**Cabana Night –** Hosted by INETA				

DAY 3 — FEBRUARY 17

		Java	.NET	Web Services	
7:30	Registration				
8:00	**FREE Tutorial** – Patrick Hynds and Duane Laflotte, Critical Sites – **Security, The New Reality** (free with VIP preregistration)				
9:00		(J-8) Design Patterns and Project Organizational Techniques for "Write Once, Debug Everywhere"	(.NET-8) Migrating ASP to ASP.NET	(WS-8) SOA: From Pattern to Production	
10:00		(J-9) Using Grid Computing with Web Services and J2EE to Create Internet-based SOAs	(.NET-9) Smart Client Development with the Offline Application Block	(WS-9) High Performance Web Services – Tackling Scalability and Speed	(WS-9B) Effective Risk Abatement and Success in a Service-Oriented World
11:00	**Keynote** – Mike Milinkovich, Executive Director, Eclipse Foundation				
12:00	**EXPO OPEN** (12 P.M.–4 P.M.)				
3:00		(J-10) Java Web Services Programming Tips & Tricks	(.NET-10) CLR Internals	(WS-10) So You Want an SOA: Best Practices for Migrating Toward Service Orientation in the Enterprise	
4:00		(J-11) JCP Program: How the Java Technology Binary Software Standard is Managed and Evolves	Visit Web site for update	(WS-11) Four Abilities SOA Will Lack Without a Registry	

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

WS Security / Case Study

WS Security	Case Study
FREE Tutorial –Aaron Williams, JCP, **Developing Web Services Using Java Technology**	
(WSS-1) Identity in SOA	(CS-1) Developing E-Commerce Applications with Web Services
(WSS-2) Securing Web Services with WS-Security	(CS-2) Developing Enterprise Class Web Services
	(CS-3) Service-Oriented Development on NetKernel – Patterns, processes and product to reduce the complexity of IT systems
(WSS-3) Anatomy of a Web Services Attack	

WS Security	Case Study
(WSS-4) Using Mobile Phones as an SSO Authentication Device in SOA Solutions	(CS-4) Orchestrating FORCEnet Engagement Packs with BPEL for Web Services
(WSS-5) Building Intelligent Enterprises with Novell's Identity-Driven Computing	(CS-5) CPI: A Globally Integrated Problem-Tracking and Resolution System Using Java Web Services
(WSS-6) XML Content Attacks	(CS-6) The Transformation of SiteRefresh into a Web Services
(WSS-7) The Interoperability Challenge of Web Services Security Standards	

WS Security	Case Study
on)	
(WSS-8) Transitioning Successfully to SOA and Web Services: Building the Infrastructure for SOA Growth	(CS-8) Using SOA and Web Services to Issue Business Licenses in the District of Columbia
in d	Visit Web site for update	(CS-9) Developing Web Services with Eclipse

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

WEB SERVICES TRACK

WS-1 Ensuring Web Services Interoperability
CHRIS FERRIS, IBM
Tuesday, Febuary 15, 2005 9:00 A.M. – 9:50 A.M.

Despite the open industry standards that underlie Web services, interoperability has been a key challenge for vendors and customers implementing Web services. One reason for this is that the relevant industry standards often permit multiple acceptable implementation alternatives. This presentation will discuss in detail the challenge of Web services interoperability and the role played by the premier industry organization formed to address it, the Web Services Interoperability Organization. In particular, the presentation will cover the critical importance of WS-I profiles to an organization's Web services initiatives, including the manner in which companies can put WS-I profiles immediately to work.

BIO: Chris Ferris is chair of the WS-I Basic Profile Working Group and a senior technical staff member with IBM's Emerging Technology Group. He has been actively engaged in open standards development for XML and Web services since 1999 and is an elected member of the OASIS Technical Advisory Board. Chris is also a coauthor and editor of the WS-Reliable Messaging specification.

WS-2 Web Services Standards: Going Behind the Mask
GLEN DANIELS, SONIC SOFTWARE
Tuesday, Febuary 15, 2005 10:00 A.M. – 10:50 A.M.

Web services and service-oriented architectures (SOAs) are emerging as an integral part of the enterprise IT strategy. According to a recent IDC study, Web services – related revenue is expected to triple from $1.1 billion worldwide in 2003, to $3.4 billion in 2004, and $16.6 billion by 2008. As SOAs proliferate and the number of Web services added to them increases, standards will play an increasingly significant role. This session will look at the state of key Web services standards such as WS-Choreography, WS-Reliability and WS-ReliableMessaging, SOAP/MTOM/XOP, WSDL, XPath, XQuery, and WS-Notification as well as related Java standards and open source efforts. It will also look at the organizational impact of standards adoption in the industry.

BIO: Glen Daniels is manager of standards and consortia at Sonic Software and coauthor of *Building Web Services with Java.* He has been working with Web services technologies since their inception in the late '90s, and in addition to developing products and helping to found Apache's Axis project, he has been an active participant in standards bodies such as the W3C, and a member of the SOAPBuilders interoperability group.

WS-2B Solving Complex Business Problems Though SOA
JOHN DALY, NETNUMINA
Tuesday, Febuary 15, 2005 3:00 P.M. – 3:50 P.M.

BIO: John Daly is a recognized leader in service oriented architecture, enterprise wide integration and business intelligence. As Managing Director of netNumina, a leading firm focused on financial service, pharmaceutical and other fortune 100 companies, he is responsible for helping companies solve complex business problems by delivering sophisticated technology solutions that combine proven technologies with cutting edge thinking. John's ability to deliver technological solutions specifically designed for the user experience separates his approach from those of netNumina's competitors.

WS-3 The XML Data Challenge
JONATHAN BRUCE, DATADIRECT TECHNOLOGIES
Tuesday, Febuary 15, 2005 4:00 P.M. – 4:50 P.M.

Most businesses store and query data with relational databases but need to use Extensible Markup Language (XML) to exchange and display data on the Web and with vendors and partners. As a result, programmers need to deal with both relational and XML data, often at the same time. Emerging standards such as XQuery, XQJ, and SQL/XML, promise to revolutionize data exchange and the ways applications are developed, deployed, and utilized. Learn the key facts about these standards, including what they mean, when

they will be available, and what you, the developer, can do to prepare.

BIO: As the XML Product Manager at DataDirect Technologies, Nancy Vodicka is responsible for DataDirect Connect for SQL/XML, a database-independent SQL/XML implementation that is currently shipping, and DataDirect XQuery, a database-independent XQuery implementation that is currently in development. Nancy has more than 15 years experience in the software industry working with technologies such as XML, Web services, relational databases, and SQL.

WS-4 How To Bulletproof Your Web Services
DAVID MCCAWS, PARASOFT
Wednesday, Febuary 16, 2005 9:00 A.M. – 9:50 A.M.

Web services are gaining industry-wide acceptance and usage and are moving from proof-of-concept deployments to actual usage in mission-critical enterprise applications. Web services range from major services such as storage management and customer relationship management to much more limited services such as furnishing stock quotes or providing weather information. As companies and consumers begin to rely more and more on Web services, the need for developing reliable, high-quality Web services is even stronger. This session will explain issues specific to Web services and will illustrate solid engineering and testing practices required to ensure complete Web service functionality, interoperability, and security. Whether creating Web services from scratch or integrating a legacy back-end server via Web services, the practices and principles outlined in this session will be of great benefit.

BIO: David McCaw has over eight years of experience in helping software development teams improve quality throughout the development process. Over the last three years, he has led the Parasoft Web Services Solutions team, which has developed an industry-leading approach for Web services testing. He has implemented Web service quality solutions for development groups in organizations such as Sabre-Holdings, Yahoo! Overture, and McGraw-Hill. McCaw has an extensive background in the areas of Java and Web service reliability, performance, and security. He is involved with OASIS and WS-I, and is a frequent speaker at industry events.

WS-5 The Role of Policy in Web Services Integration – It's More Than Just Security
TOUFIC BOUBEZ, LAYER 7 TECHNOLOGIES
Wednesday, February 16, 2005 10:00 A.M. – 10:50 A.M.

Too often today the preferences, terms, and conditions describing how a Web service behaves when discovered and invoked is programmed right into the business logic. Hard-coding this behavior logic however introduces cost, complexity, and rigidity into a Web services architecture. A better approach is to abstract a Web services usage "policy" out of code where this metadata can be managed as need be. This session introduces the concept of Web Services Policy and describes how the construct can be used to implement a more customized and versatile Web service infrastructure.

BIO: Toufic Boubez is a well-respected and renowned Web services visionary. Prior to cofounding Layer 7 Technologies, Toufic was the chief Web services architect for IBM's Software Group and drove their early XML and Web services strategies. He is a sought-after presenter and has chaired many XML and Web services conferences. He is an author of many publications and his most recent book is the top-selling *Building Web Services with Java: Making Sense of XML, SOAP, WSDL ,and UDDI.*

WS-6 B2B Policy Enforcement: The Third Rail of SOA Implementation
ALISTAIR FARQUHARSON, DIGITAL EVOLUTION
Wednesday, Febuary 16, 2005 4:00 P.M. – 4:50 P.M.

One of the great benefits of a service-oriented architecture is the ability it gives you to extend programmatic, integration capabilities to business partners. Going beyond simple sharing of data with partners, SOA enables true B2B application integration. At the same time, this capability creates a vexing security policy enforcement dilemma. How can you be sure that a user from a partner organization is actually authorized to integrate with your applications? How can you authenticate that user? Do you even want that headache in the first place? This session will discuss the issues that arise in B2B security policy enforcement and explore several proven approaches

---

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

to solving the problem. In particular, it will focus on the emerging technology of XML Virtual Private Networks (XML-VPNs) and their potential to mitigate security policy enforcement issues in B2B SOA implementations.

BIO: Alistair Farquharson is the CTO of Digital Evolution, where he spearheads product development and provides thought leadership to enterprise customers implementing Web services. His skills span many industries and include designing and implementing system architectures, as well as spearheading initiatives such as development/team lead. He is an expert in custom-application development, distributed environments, architecting scalable hardware and software applications and systems, and Web services application development.

WS-7 Driving SOA Governance
BRENT CARLSON, LOGICLIBRARY
Wednesday, Febuary 16, 2005 5:00 P.M. – 5:50 P.M.

In the past year, Web services and service-oriented architectures (SOAs) have become mainstream because of their ability to provide business agility and flexibility through integration, productivity, and reuse. With SOA enablement on the rise, IT groups must address SOA governance as a means of controlling what and how services located within an SOA are deployed. This session will discuss SOA governance, specifically how an organization can manage and control assets and artifacts located within an enterprise, while ensuring that deployed assets meet an organization's business and technical architectural standards. It will also outline governance best practices such as monitoring the UDDI publish process in order to seamlessly tie together the development and operational views of Web services within the enterprise.

BIO: Brent Carlson drives the development and delivery of LogicLibrary's products. He is a 17-year veteran of IBM, where he served as lead architect for the WebSphere Business Components project and held numerous leadership roles on the "IBM San Francisco Project." He is a member of the Eclipse Board of Stewards and a BEA Regional Director.

WS-8 SOA: From Pattern to Production
DAVID CHAPPELL, SONIC SOFTWARE
Thursday, Febuary 17, 2005 9:00 A.M. – 9:50 A.M.

Service-oriented architecture (SOA) represents the opportunity to achieve broad-scale interoperability, while providing the flexibility required to continually adapt technology to business requirements. No small feat, particularly when one considers the extent and complexity of today's IT environments. As both a technology concept and IT discipline, the challenge inherent in SOAs is maintaining the right architectural approach. If all services in an SOA are treated as interdependent point-to-point interfaces, then the complexity of implementing and maintaining them in this spaghetti-like architecture becomes enormous. The enterprise service bus (ESB) has emerged as one of the first true SOA product offerings, bringing SOA from pattern to production. ESBs provide a framework for building and deploying an event-driven, enterprise SOA and accommodates the configuration, hosting, and management of integration components as services across the business.

BIO: VP and chief technology evangelist for Sonic Software, Dave Chappell has over 18 years of experience in the software industry covering a broad range of roles including R&D, code-slinger, sales, support, and marketing. He also has extensive experience in distributed computing, including message-oriented middleware, CORBA, COM, and Web application server infrastructure.

WS-9 High Performance Web Services – Tackling Scalability and Speed
SAMEER TYAGI, SUN MICROSYSTEMS
Thursday, Febuary 17, 2005 10:00 A.M. – 10:50 A.M.

Web services facilitate application-to-application integration and interoperability across different platforms. However, critics usually point to an inefficient processing model and bandwidth requirements for developing Web services. This is often cited as a reason why Web services cannot perform and scale well in production environments. This session takes a detailed look at performance and scalability issues around Web services in the real world, as well as strategies that architects and developers can adopt to mitigate such risks in these applications. Some analytical and modeling strategies that enable acceptable application performance will also be covered.

BIO: Sameer Tyagi works as a senior Java architect with Sun Microsystems. He remains focused on architecture, design, and implementation of large-scale enterprise applications with Java technology. His publications include industry periodicals and books on Java and J2EE technologies including *Java Web Services Architecture.*

WS-9B Effective Risk Abatement and Success in a Service-Oriented World
PAUL LIPTON, COMPUTER ASSOCIATES
Thursday, Febuary 17, 2005 10:00 A.M. – 10:50 A.M.

IT leaders are hoping to leverage the benefits inherent in Web services and Service-Oriented Architectures (SOA) to enable their businesses to be far more competitive and to find new operational efficiencies. But, can we depend on these new technologies and approaches? Management and security are a common concern today and this session provides the necessary background and perspective on both the business and the technical issues. We will examine important principles and recommendations using real-world examples to illustrate key concepts.

BIO: Paul Lipton is a senior architect in the Web services and application management team at Computer Associates (CA) as well as a Strategist in the Office of the CTO. He has been an architect and developer of enterprise systems for over 20 years, and has worked closely with key CA customers to solve important business challenges through the creation of manageable, mission-critical distributed solutions.

WS-10 So You Want an SOA: Best Practices for Migrating Toward Service Orientation in the Enterprise
ERIC NEWCOMER, IONA
Thursday, Febuary 17, 2005 3:00 P.M. – 3:50 P.M.

Replacing complex, monolithic applications with nimble applications built from exposed services promises increased developer productivity, greater flexibility, and ultimately reduced cost. The adoption of Web services and SOA can also remove a significant level of complexity and integration problems from enterprise application development projects. But, as with any large-scale project, IT departments must have the right plan and the right resources in place to ensure a successful transformation of their computing infrastructure. This article will explore what IT organizations need to know to be successful in their attempts to migrate the enterprise to a service-oriented architecture.

BIO: In the role of chief technology officer at IONA, Eric Newcomer is responsible for IONA's technology roadmap and the direction of IONA's e-business platforms as relates to standards adoption, architecture, and product design.

WS-11 Four Abilities SOA Will Lack Without a Registry
LUC CLEMENT, SYSTINET
Thursday, Febuary 17, 2005 4:00 P.M. – 4:50 P.M.

A service-oriented architecture (SOA) is the design blueprint for seamless connectivity between business processes and IT infrastructure, enabling innovation and improving productivity. SOA provides the most efficient, standard way to dynamically interoperate with any customer, supplier, product or employee. SOA makes integration intrinsic. Web services are the foundation building blocks of an SOA, and they are already proliferating inside most enterprises. In an SOA, Web services become business services with the ability to perform a particular function or access data dynamically. This presentation will discuss the four abilities that a registry provides for an SOA.

BIO: Luc Clement is director of product marketing, SOA Registry for Systinet. He is also cochair for the UDDI Specification Technical Committee. Formerly Microsoft UDDI Program Manager, Luc is well known in the UDDI community and has been heavily involved with the UDDI specification for several years.

WEB SERVICES SECURITY TRACK

WSS-1 Identity in SOA

SEKHAR SARUKKAI, OBLIX

Tuesday, Febuary 15, 2005 9:00 A.M. – 9:50 A.M.

The mainstreaming of SOAs requires a more general approach to the notion of identities – beyond simply central management of people identities and into the realm of managing applications, devices, and other identities that represent entities that are first-class participants in this application network while also providing this as a pluggable service into the larger enterprise SOA. Enterprises should view identity as a service that is ubiquitously available and is a shared infrastructure service necessary for application networking, rather than as being managed by a server, such as an Authentication or Access server. While it makes architectural sense to consider an Identity service, there are business and related drivers that may force the need to deploy such an architecture.

BIO: Sekhar Sarukkai is currently a technical architect at Oblix. He was the original founder and CTO of Confluent Software, a leading Web services management company, which was acquired by Oblix in 2004. He holds a PhD in computer science from Indiana University.

Attend a FREE* Tutorial

Thursday, February 17, 2005, 8 – 11 a.m.

Using a Web Services Framework to Build SOA Applications

ASHISH LARIVEE

ASHISH LARIVEE, NOVELL

Service-oriented architecture (SOA) has quickly taken center stage as the primary development style of the next decade and beyond. Businesses of all types are preparing for the SOA revolution that promises consistency of process, reduction in duplicate work, ease of maintenance, service reusability and broad interoperability. The Web Services Framework (WSF) is the foundation that can deliver on the promise of SOA. Come learn about the components of an SOA including the core WSF standards. Attend this free Novell tutorial and learn about the future of SOA-style development, including legacy system enablement, platform interoperability, open source in SOA and building composite applications that leverage SOA services using Novell exteNd. In this session, we will create SOA application logic that orchestrates legacy services, JBoss4 Web services and MS.Net Web services. We will create Web services in Novell exteNd, Eclipse and Visual Studio respectively. We will then orchestrate these Web services and expose a single course, process level interface to public Web service consumers. Finally, we will cre-

Novell.

* Free Tutorials with VIP Preregistration ONLY!

WSS-2 Securing Web Services with WS-Security

DR. JOTHY ROSENBERG, SERVICE INTEGRITY

Tuesday, Febuary 15, 2005 10:00 A.M. – 10:50 A.M.

An up-to-date, comprehensive, and practical discussion of Web services security, and the first to cover the final release of new standards SAML 1.1 and WS-Security. Comprehensive coverage and practical examples of the industry standards XML Signature and XML Encryption will be presented.

BIO: Dr. Jothy Rosenberg is a serial entrepreneur. He is a founder and CTO of Service Integrity, a company that helps Web service operators see, measure, understand and fully leverage operational and business information flowing across their Web service networks. Prior to this venture, Jothy cofounded GeoTrust, the world's second largest certificate authority.

WSS-3 Anatomy of a Web Services Attack

MAMOON YUNUS, FORUM SYSTEMS

Tuesday, Febuary 15, 2005 4:00 P.M. – 4:50 P.M.

A broad range of new security threats is facing enterprises implementing XML Web services, leaving the enterprises open to financial risks, loss of property, and tarnished reputations. The basic rules of security – authentication, authorization, and auditing – no longer provide adequate security in the new world of straight-through processing paths into mission-critical systems. What's worse, WSDL documents provide a guide book to security exposure. Most attacks on traditional Web-based applications exploit weaknesses in HTML-enabled custom, or packaged, applications. However, hackers and other malicious users are quickly uncovering new techniques at the SOAP/XML data level that bypass HTML and target weaknesses in Web services programming, technology, and architecture. This session will outline the innovative techniques that hackers use to map out the vulnerabilities of an organization's network, and how Web server security must now complement Web services security in order to provide an adequate defense.

BIO: Mamoon Yunus, CTO of Forum Systems, was previously a global systems engineer for webMethods, where he developed business integration strategy and architecture for Global 2000 companies. He is an industry-honored CTO in advanced technological solutions for enterprise customers.

WSS-4 Using Mobile Phones as an SSO Authentication Device in SOA Solutions

DR. MICHAEL JUNTAO YUAN, UNIVERSITY OF TEXAS

Wednesday, Febuary 16, 2005 9:00 A.M. – 9:50 A.M.

Federated identity management across multiple single-sign-on domains is a major challenge for SOA-based solutions to fully realize its business potential. The traditional username/password combination is often too weak to protect the extremely sensitive single-sign-on credentials. The new-generation mobile phones could be used to identify and authorize users for SOA services. The device-based authentication scheme depends on not only "what you know" but also "what you own." This session will discuss new advances in Java-based mobile devices to interoperate with Sun's Liberty Alliance Services.

BIO: Dr. Michael Juntao Yuan is an author, developer, and software architect for end-to-end mobile software. He is a contributing editor to *JDJ* and a frequent contributor to many developer forums and publications. He is the author of two books. Michael has a PhD from the University of Texas at Austin and teaches information systems at the college level.

WSS-5 Building Intelligent Enterprises with Novell's Identity-Driven Computing

ASHISH LARIVEE, NOVELL

Wednesday, Febuary 16, 2005 10:00 A.M. – 10:50 A.M.

Companies are now facing complexities dealing with issues such as regulatory compliance and security while still providing for company-wide collaboration between employees, partners, and suppliers. Identity systems are becoming a crucial component of applications, enabling developers to take advantage of a new set of services that know who you are, where you are, what you are trying to do, and can adapt to your changing business needs. Identity-driven computing addresses these problems by applying best practices learned from Novell's leadership in identity management for the management of people to all aspects of an enterprise, including servers, PCs, devices, applications, and even Web services. This presentation will outline identity-driven computing, describe the attributes of an identity-driven application, and discuss steps enterprises can take to make the transition to an identity-driven computing environment.

BIO: With more than nine years of experience in the software industry, Ashish Larivee has designed and developed many enterprise applications across a variety of platforms, including Microsoft, Lotus Notes/Domino, and J2EE. In 1999, Ashish joined SilverStream Software, acquired by Novell in July 2002, and has served in various roles in consulting, development, and technical marketing. In her current role, she helps define the strategy and product direction across Novell's Web Application Development Products.

WSS-6 XML Content Attacks
GIRISH JUNEJA, SARVEGA, INC.

Wednesday, Febuary 16, 2005 4:00 P.M. – 4:50 P.M.

This talk defines a new class of threats, XML Content Attacks, and differentiates these threats from more general Web services attacks and XML security-based attacks. These three related but distinct threat areas are explained. The talk covers XML Content Attacks with regard to tree-based parsing exploits related to coercive parsing, node-depth attacks, and DOM. XML grammar validation exploits such as schema poisoning and lax-content models are discussed, and why traditional schema validation cannot ensure content-model consistency. Web services attacks like WSDL scanning and parameter tampering (SQL Injection, SOAP array attack) are discussed – highlighting common mistakes made when applying message-level security (WS-Security).

BIO: Girish Juneja has more than 15 years' experience in the high technology industry with extensive product management, product strategy, engineering management, and technology marketing expertise. He is the cofounder of Sarvega. Since Sarvega's inception, Girish has led the Sarvega engineering and customer services organizations to develop Sarvega's industry-leading core XESOS technology and XML Networking products.

WSS-7 The Interoperability Challenge of Web Services Security Standards
EVE MALER, SUN MICROSYSTEMS

Wednesday, Febuary 16, 2005 5:00 P.M. – 5:50 P.M.

The Web Services Interoperability Organization chartered its Basic Security Profile Working Group to develop an interoperability profile involving transport layer security, SOAP message layer security, encryption, signatures, and other security considerations. This session will discuss the interoperability challenges presented by current Web services security standards and the work of the WS-I Basic Security Profile. The session will highlight typical Web services security threats and countermeasures and the related design goals, usage conventions, and conformance testing of the soon-to-be-released Basic Security Profile.

BIO: Eve Maler is an XML standards architect at Sun Microsystems, where she coordinates Sun's involvement with Web services security standards such as SAML and the WS-I Basic Security Profile.

WSS-8 Transitioning Successfully to SOA and Web Services: Building the Infrastructure for SOA Growth
DAN FOODY, ACTIONAL

Thursday, Febuary 17, 2005 9:00 A.M. – 9:50 A.M.

This session will address how to approach service-oriented architecture (SOA) management from a project-based level while still allowing room for future expansion and incremental growth to an enterprise-wide SOA. The session will provide valuable insight into how SOA management can help organizations ease the complexity of moving toward a loosely coupled environment.

BIO: As CTO at Actional, Dan Foody leverages his extensive experience in enterprise systems software toward designing robust and manageable service-oriented architectures. He is an active participant in the Web services standards community, including WS-I and OASIS, where he spearheads Actional's contributions on the OASIS Web Services Distributed Management Committee (WSDM).

RESERVE YOUR HOTEL ROOM NOW! AT THE SHERATON BOSTON HOTEL
THE OFFICIAL CONFERENCE HOTEL OF EDGE 2005 EAST

THE SHERATON BOSTON HOTEL
PRUDENTIAL CENTER
39 DALTON STREET
BOSTON, MA 02199
PHONE (617) 236-2000
FAX (617) 236-1702

Hotel Arrangements Are Easier Than Ever! Web Services Edge 2005 East has secured special group rates, well below regular rates, at one of the finest hotels in Copley Square, The Sheraton Boston Hotel. The hotel is directly next door to the Hynes Convention Center, and is connected via The Prudential Mall. To learn more about hotel savings, or to make your individual or group reservations please

contact Meeting & Travel Consultants:
Toll free (866) 549-3985
Direct (732) 549-3985
Fax reservations to (732) 549-3987.
Make online reservations by Jan. 17, 2005
visit: http://www.mtcreservations.com

SPECIAL CONFERENCE RATES

Single	Double
$196	$196

Credit card information is required to guarantee reservations and expedite confirmation. Confirmations will be mailed directly from the hotel, time permitting. All changes and cancellations should be made directly through MTI.

DRIVING DIRECTIONS TO THE SHERATON BOSTON FROM LOGAN INTERNATIONAL AIRPORT

• Follow direction sign from terminals to Sumner Tunnel/Rt. 93 North • Take the Rt. 93 North ramp as you exit the Sumner Tunnel • Exit at Storrow Drive (26 B) • Follow Storrow Drive approximately 2 miles to the Fenway/Kenmore Exit (1st exit after Massachusetts Ave., on left) • Stay left as exiting Storrow Drive going towards Fenway • Continue to 1st set of lights, staying left, going onto Boylston Street • Go through 4 sets of lights on Boylston Street • The Hynes Convention Center's main entrance driveway will be on the right immediately after going through the 4th set of lights.

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

Java and Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. SYS-CON Publications, Inc., is independent of Sun Microsystems, Inc. All brand and product names used on these pages are trade names, service marks or trademarks of their respective companies.

VISIT www.sys-con.com/edge FOR TIMES & SCHEDULES
The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

International Web Services Conference & Expo

For Assistance **Tel:** 201-802-3066 **Fax:** 201-782-9601

CONFERENCE & EXPO: February 15 – 17, 2005

John B. Hynes Veteran Memorial Convention Center • Boston, MA

THREE WAYS TO REGISTER FOR CONFERENCE

1) **On the Web**: Credit Cards or "Bill Me" Please make checks payable to SYS-CON Events
2) **By Fax:** Credit Cards or "Bill Me" 201-782-9601
3) **By Mail:** 135 Chestnut Ridge Road, Montvale, New Jersey 07645, Attention: Registration

Please note: Registrations are not confirmed until payment is received.

Please complete sections 1, 2, 3 and 4

1 YOUR INFORMATION (Please Print)   ☐ Mr.   ☐ Ms.   ☐ Mrs.

First Name _____ Last Name _____
Title _____
Company _____
Street _____
Mail Stop _____
City _____
State_____ Zip _____ Country _____
Phone _____
Fax_____ E-Mail _____

2 PAYMENT METHOD: (Payment in full due with registration)

☐ Check or Money Order Enclosed (Registration confirmed upon receipt of payment)

Check #_____ Amount of Check $_____

Charge my ☐ Visa   ☐ MasterCard   ☐ American Express   ☐ Discover

Name on card_____
Card #_____ Exp. Date_____
Signature_____
Billing Address (if different from mailing address)
_____

3 PLEASE INDICATE YOUR CONFERENCE CHOICE:

Total Registration fee $_____

	Before 12/31/04	Before 1/21/05	Before 2/11/05	On Site
☐ **3D: Three Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions.	$1,195.00	$1,395.00	$1,495.00	$1,695.00
☐ **2D: Two Day Conference** Good for two days of the conference program, including keynotes, all conference sessions.	$995.00	$1,195.00	$1,295.00	$1,395.00
☐ **1D: One Day Conference** Good for one day of the conference program, including keynotes, all conference sessions.	$595.00	$695.00	$695.00	$795.00
☐ **VIP PASS** FREE Tutorials with VIP preregistration	FREE	FREE	FREE	$75.00
☐ **EO: Expo Hall Only** Includes keynotes, expo presentations, and workshops	FREE	FREE	FREE	$50.00

☐ FREE - Novell Tutorial (Feb. 15)
☐ FREE - Developing Web Services Using Java Technology (Feb. 15)
☐ FREE - The Next Generation of Visual Studio (Feb. 16)
☐ FREE - Security, The New Reality (Feb. 17)

SPECIAL GROUP RATES

Registering a group of four or more? Please call Jim Hanchrow at 201-802-3066, and receive an additional discount.

4

A. Your Job Title
☐ Senior Software Developer/Engineer/Architect/Analyst
☐ Director of IT/Development/Engineering
☐ Software Developer/Engineer/Architect
☐ Systems Architect/Engineer/Analyst
☐ Software Consultant/Analyst
☐ Project Manager/Team Leader
☐ Business Development
☐ General Manager/Division or Department Head
☐ CTO,CIO, Chief Architect
☐ CEO, President, Owner, Partner/Principal
☐ VP/Technical Director
☐ Network Administration/Manager/Consultant
☐ Web Developer/Programmer
☐ Database Administrator
☐ VP/Sales & Marketing Director /Product Management
☐ Other (please specify)_____

B. Business/Industry
☐ Agriculture/Mining/Oil/Gas
☐ Computer Hardware and Electronics
☐ Construction/Architecture/Engineering
☐ Consulting (non-computer)
☐ Education
☐ Banking/Insurance/Accounting
☐ Government (including Military)
☐ Healthcare/Pharmaceuticals/Biotech/ Biomedical
☐ Internet/Web/e-Commerce
☐ Manufacturing and Process (other than computer related)
☐ Media/Marketing/Advertising
☐ Non-Profit/Trade Association
☐ Real Estate/Legal
☐ Telecommunications
☐ Transportation/Rail/Sea/ Air/Ground
☐ Travel/Hospitality/Entertainment
☐ Utilities
☐ Wholesale/Trade/Distribution/ Retail
☐ Other (please specify) _____

C. Total Number of Employees at Your Location and Entire Organization (check all users that apply):

	Location	Company
10,000 or more	01 ☐	01 ☐
5,000 - 9,999	02 ☐	02 ☐
1,000 - 4,999	03 ☐	03 ☐
500 - 999	04 ☐	04 ☐
100-499	05 ☐	05 ☐
100 or less	06 ☐	06 ☐

D. Please indicate the value of software products and services that you recommend, buy, specify or approve over the course of one year:

☐ $10 million or more
☐ $1 million - $9.9 million
☐ $500,000 - $999,999
☐ $100,000 - $499,999
☐ $10,000 - $99,999
☐ Less than $10,000
☐ Don't know

☐ $10 billion or more
☐ $1 billion - $9.9 billion
☐ $100 million - $999 million
☐ $10 million - $99.9 million
☐ $1 million - $9.9 million
☐ Less than $1 million
☐ Don't know

E. What is your company's gross annual revenue?
F. Do you recommend, specify, evaluate, approve or purchase mainframe products or services for your organization?
01 ☐ Yes   02 ☐ No

G. Which of the following products, services, and/or technologies do you currently approve, specify or recommend the purchase of?
☐ Application Servers
☐ Web Servers
☐ Server Side Hardware
☐ Client Side Hardware
☐ Wireless Device Hardware
☐ Databases
☐ Java IDEs
☐ Class Libraries
☐ Software Testing Tools
☐ Web Testing Tools
☐ Modeling Tools
☐ Team Development Tools
☐ Installation Tools
☐ Frameworks
☐ Database Access Tools / JDBC Devices
☐ Application Integration Tools
☐ Enterprise Development Tool Suites
☐ Messaging Tools
☐ Reporting Tools
☐ Debugging Tools
☐ Virtual Machines
☐ Wireless Development Tools
☐ XML Tools
☐ Web Services Development Toolkits
☐ Professional Training Services
☐ Other [Please Specify] _____

If you require special assistance covered under the Americans with Disabilities Act, please call 201-802-3066 by February 4, 2005.

CANCELLATIONS, SUBSTITUTIONS, REFUNDS
Fax written request to SYS-CON Registration 201-782-9601. Requests for refunds received prior to January 7, 2005 will be honored, less a 10% handling charge; requests received after January 7, 2005, and before February 4, 2005, will be honored less a 20% handling charge. No requests for refunds will be honored after February 4, 2005. Requests for substitutions must be made in writing prior to February 11, 2005. No one under 18 is permitted to attend. No warranties are made regarding the content of sessions or materials.

Speakers, sessions and schedule are subject to change without prior notice.

No solicitation by anyone other than official exhibitors, sponsors or marketing partners is permitted. Such behavior is cause for expulsion without refund.

CFEclipse: The Developer's IDE

Simple and flexible

Many paths lead developers to Cold-Fusion. In the past ColdFusion was often viewed as an entry-level language enabling designers and other nonprogrammers to build dynamic sites. Since the release of CFMX it has become more recognized among developers familiar with languages such as Java and PHP.

This is partly due to language enhancements such as CFCs, and partly because it now runs as a Web application on top of a J2EE servlet container, thus allowing it to be used to develop the presentation layer for complex J2EE applications. For both types of developers there are requirements that an IDE must support. Many Web development tools are geared toward the first type of users mentioned above. They include features such as WYSIWYG Design Mode and image mapping functionality. For many developers falling into the second category, the WYSIWYG tools are lacking in the features that they are used to in other IDEs, and the non-WYSIWYG tools are either aging or not geared specifically to ColdFusion development. The goal of the CFEclipse project is to provide a tool for this second type of developer.

CFEclipse (CFE) is a plug-in developed to integrate into the open platform created by the Eclipse Foundation. Eclipse is designed specifically to be flexible enough to be used as a base for an IDE, a simple text editor, Web browser, e-mail client, or practically any other user application you care to mention. Eclipse is often perceived as a Java IDE, but in fact the Java development features are provided by nothing more than a plug-in much the same as (albeit much more mature than) CFEclipse. Eclipse itself is written in Java and runs on just about any platform that has a JVM version 1.4 or later.

Although other ColdFusion IDEs are also under development for use within the Eclipse platform, CFE is set apart by one main difference: the CFE project is an open source development project. The benefits of open source development are beyond the scope of this article; however, it is fundamental to the success of this IDE. Anyone can download the source code for CFE, and with that code add any feature he or she desires. Although this task may seem daunting for some beginners, it is also a very liberating experience. In preparation for this article we polled the CFEclipse users' mailing list (http://cfeclipse.tigris.org/servlets/ProjectMailingListList) to learn which features were other users favorites. One of the features mentioned most was the amazing release cycle. Users really appreciate that their suggestions are taken seriously and implemented quickly. This is due to the nature of open source software; anyone can contribute to the code base, which allows for features to be added quickly.

The concept of synergy is an amazing thing. The idea is that the sum of the contributions of the group exceeds those of individual developers. Think of just a few of the people you know who are passionate about technological advances and the things they can accomplish. Now imagine those people working together under a common banner to achieve a common goal. This is what open source development is about. Closed source development is hard to do without strong financial backing. We are all individuals and don't want to be at the mercy of an independent financial backer. There is also a lot of satisfaction in creating the tool you use to do the majority of your day-to-day work.

So now that you are familiar with the "why" and the "how," let's focus on the "what." CFEclipse is built as a developer's CF tool. CFE includes features that every person who writes CF code needs on a daily basis. All of the standard features you would expect are included. CFE supports syntax highlighting of CF code, CF tag, function insight, and project-based development. Here are some features you'll find in CFE that you won't find in contending tools:

- **Bracket matching:** When writing functions or script, it is very easy to get lost in your nested elements. If you place your cursor next to a character that supports matching brackets (( ) { } [ ]), CFE will highlight both the character you are near and its matching character. Never again get lost inside a dateAdd() inside a dateFormat() function!
- **Code folding:** Available in CF Studio and Homesite, this feature has been resurrected in CFE and has had some bells and whistles added to it. When working in large files it is

By Simeon Bateman

By Stephen Milligan

often convenient to be able to highlight a section of code and collapse it. This feature adds an icon to the right margin allowing you to expand and collapse this bit of code as you need it. Not only does CFE allow you to add your own collapsed sections, but it can automatically add them to many block-level CFML constructs. By default, when you open a file it will collapse CFML and HTML comments and cffunction blocks that span more than three lines. It will also add folding markers in the right margin for cfquery, cfscript, cfloop, cfif, cfswitch, and cfcase, and blocks that span more than three lines, but these will not be collapsed automatically. These defaults can be customized in the preferences area. A very handy organization feature!

- *Methods view:* Since the addition of ColdFusion Components in CFMX, many developers have been wooed by the benefits they provide. When working in CFE you can open a CFC, file, and in the Methods View see a summary of the cffunction tags in your document. You can filter your functions based on type of access (private, package, public, remote) and you can sort them alphabetically. Double-clicking on a method name will move the cursor to that method in the document and expand the method if it is currently collapsed.

- *Outline view:* The Methods view is convenient for files such as CFCs, which are mostly made up of cffunction tags. Sometimes it's also nice to have an overview of the structure of normal CFML files. The Outline view is designed to do exactly this. The real benefit is that when working in large files, you can filter your view based on the type of tag. You can filter on the currently selected element, or on a list of predefined tags such as cfmodule, cfset, cfquery, cfinclude, and cfcase. This makes working with long documents much more manageable (see Figure 1).

- *Versioning integration:* Although not a ColdFusion-specific feature, the integration with Versioning Systems via other Eclipse plug-ins is incredible. Beyond being able to utilize all the features of CVS and SVN repositories, Eclipse also has a local history. The local history allows you to compare against the versions of the file that Eclipse has kept locally on your computer. So where normally you would need to go to the versioning server to compare previous versions, Eclipse allows you to do this very easily with every saved version of the file. By default Eclipse maintains seven days worth of changes locally.

- *Integrated browser:* Many Web-centric IDEs will allow you to launch the page you are currently working on into an external browser. CFEclipse allows you to open a browser window
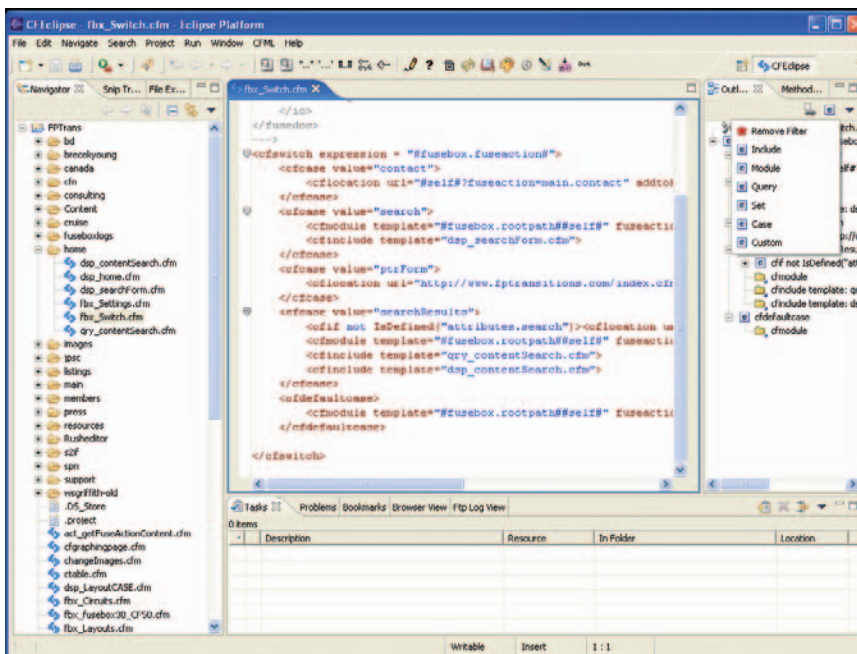

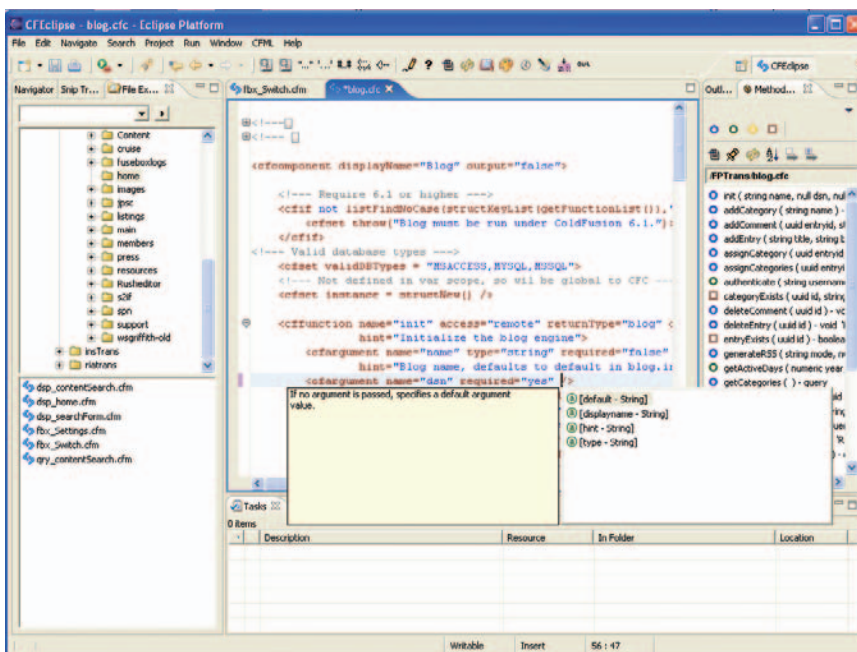**Figure 1: Code Folding Tabs, Project Navigation, and Outline View**

Figure 2: File Explorer, Code Completion Hint for cfargument, and CFC Methods View

inside the IDE itself. This is very similar to what you get in both Dream-Weaver and Homesite. The difference in CFE is that you can specify a single root URL for a project and always have that URL loaded when you fire up the browser. CFE also allows you to have two tabbed browsers active. This allows you to review the page you are working on, as well as keep the Macromedia Livedocs or any other Web site at your fingertips. And all inside the IDE!

- **Scribble pad:** At one point or another pretty much every Web developer has created a tmp.cfm file, or something similar, to test some small snippet of code outside the existing project. CFEclipse allows you to designate a file within any project that will act as a "scribble" file. While working in CFE you can activate the scribble pad via a keyboard shortcut. This will open the scribble file and load it into the integrated browser, allowing you to quickly test short snippets of code without having to hunt around in the file system for the last file you used. Using the same keyboard shortcut again will save the scribble file and refresh the integrated browser.
- **Snippets:** Reminiscent of the snippets of Homesite , snippets allow you to create blocks of code that can be entered into your document by double-clicking the snippet from the snippet view. A text trigger can also be assigned to a snippet, allowing you to type the text trigger followed by a keyboard shortcut to add the snippet to your document. This approach literally gives you an unlimited number of snippets that can be inserted using only triggers and the keyboard shortcut. The snippets implementation also allows you to use Homesite-style snippet variables of the form $${variable:default} within the snippets. When you insert a snippet of this type you are prompted for the values you would like to use. This is very convenient for creating snippets that contain skeleton code for constructs like CFC getter and setter function pairs.
- **Right-click to open/create file from cfinclude:** cfinclude offers many benefits to developers. CFE allows you to right-click on the template attribute of a cfinclude tag and choose the file to include. If the file does not exist, CFEclipse will attempt to create the file in the location you specify.
- **File Explorer:** One of the most frequent complaints I have heard of Eclipse-based plug-ins is that there is no way to edit a file that is not part of a project. The File Explorer solves this problem by allowing you to explore your computer and open any file. Another benefit of the File Explorer is that it uses the same split-view approach that Homesite uses, with folders on top and files below. The File Explorer view also allows you to browse FTP sites and edit the files as if they were local, again in the same manner as Homesite (see Figure 2).
- **TODO tasks:** Another popular feature is the ability of the IDE to track comments containing TODO items. Any comment that contains the string "TODO:" is tracked globally and can be viewed in the tasks view. When looking at the tasks view you can double-click on a task and the relevant file will be opened with the cursor on the line where the TODO: marker is located. The overview ruler in the right margin also allows you to quickly scan all the TODO tasks for the current document and jump directly to any one of them.
- **Framework support:** Currently under development are many enhancements to the CFEclipse IDE that allow specialized functionality based on frameworks such as Mach-ii, Fusebox 3, and Fusebox 4.
- **Problems view:** CFEclipse also includes a parser that checks for syntax errors in your ColdFusion document. When problems are found, they are compiled into a list that is tracked globally in the same way as the TODO tasks. In the latest development versions of CFEclipse, parser problems are also underlined in the text of the document. This makes it much easier to see exactly where the parser found a problem and it is usually relatively easy to spot the cause, such as an extra </cfif> tag in a nested set of if statements.

Installation

Installation of CFE requires that you first download some version of Eclipse. The Eclipse download page can be a confusing place, so we'll try to demystify it a little.

Releases and Builds

A release is a version of Eclipse that the Eclipse Foundation deems stable and suitable for production use. A build can come in a variety of flavors such as stream, integration, or nightly. Details on what each of those mean can be found at http://download.eclipse.org/downloads/build_types.html. If you are unsure about which version to download, download the latest release.

Once you've decided which version of Eclipse you want, you may be presented with a rather long list of things that you can download. Depending on which download mirror you are using, this list will vary in length, but you should always have an option to download at least the Eclipse SDK. CFE will work with either the Eclipse SDK or the Platform Runtime Binary. The SDK includes support for Java projects and a number of other things, while the Platform Runtime Binary provides a minimal set of base functionality. If you don't intend to do any Java development, the Platform Runtime Binary is probably the best option as you can download the rest of the tools that come with the SDK download separately. At the time of this writing, the latest release is 3.01 so the rest of this article will be working on the assumption that you have downloaded the Platform Runtime Binary for that version. You can download the necessary files at http://eclipse.org/downloads/index.php.

Once you have downloaded the appropriate version of Eclipse, you can extract it to any location on your hard drive. Once it is extracted you can run Eclipse by navigating to the folder you extracted it to and double-clicking on the Eclipse executable. The file extension of the executable depends on which operating system you are using, but it should be the only file called eclipse in that directory.

When Eclipse starts up for the first time, you will be asked to specify a location for the workspace. The workspace is what Eclipse uses in the same way as a lot of Windows applications use the user profile directory. That is, it is where a lot

of the configuration and metadata about your plug-ins will be stored. It may also be where you keep your project files, but that is certainly not a requirement and probably not recommended. If in doubt, accept the default location for the workspace. You will probably also want to tick the check box telling Eclipse to always use that location. If you do, you can change that setting at any time by going to: Window > Preferences > Workbench > Startup and Shutdown.

Once you have specified the workspace location you should be presented with the welcome screen. It's probably a good idea to explore the information provided in this page as once you start using Eclipse, you will probably forget about learning the bigger picture and want to answer specific questions. It will help you enormously in the long run if you invest the time up front as many of the concepts in Eclipse are quite different from other IDEs and editors.

Once you're satisfied that you know enough to begin, you'll probably want to download CFEclipse. Eclipse has a very convenient mechanism for doing this called update sites. Plug-in authors can create a site that has a particular structure and set of files that Eclipse can use to determine if it has the most up-to-date version of that plug-in. You can configure Eclipse to download and install a plug-in from one of these sites by going to: Help > Software Updates > Find and Install.

On the dialog box that appears, check the radio button labeled "Search for new features to install," then click Next.

On the next screen, click the button labeled "New Remote Site". In the dialog box that appears, type CFEclipse or something similar for the name, and http://cfeclipse.tigris.org/ as the URL, then click OK to close the dialog.

Check the check box beside the CFEclipse update site you just added and click Next. You should be presented with

a list of CFEclipse releases. You only need to check the check box beside the most recent release and click Next.

You should be presented with the CFEclipse license agreement, which you'll probably agree to, and then click Next.

Eclipse now has enough information to install the CFEclipse plug-in, so click the finish button.

During the install you may be presented with a warning that you are installing an unsigned Feature. This is normal, and as long as you trust us, you can continue and install the feature anyhow.

Once the installation completes you should be prompted to restart Eclipse. We recommend that you do restart because otherwise some features such as preference pages may not appear correctly.

If you managed to follow all the steps above and have restarted Eclipse, you should be ready to start using CFEclipse.

Alternative Ways to Get CFEclipse
As a Zip File

You can download CFEclipse as a zip file from the main project site: http://cfeclipse.org:
1. Close Eclipse.
2. Download the latest release and unzip it so that the com.rohanclan.cfml.xxx directory inside the zip ends up inside the Plug-ins directory for your Eclipse install. (The Plug-ins directory is a subdirectory of the directory containing the Eclipse executable.)
3. Start Eclipse again.

The CFEclipse Nightly Builds

From time to time we make a snapshot of the CFEclipse code base and release it to the public for general testing and evaluation. The purpose of these "nightly" builds is to make sure that the features we are adding are working

the way we expect, and that we haven't broken anything in the process. You can find out what the latest nightly build is by going to www.spike.org.uk/blog/index.cfm and taking a look in the right margin.

The instructions for installing the nightly build are exactly the same as for installing from the zip file on the main CFEclipse site.

Time and space constraints limit the amount of information we have been able to put into this article, but if you would like to learn more there are a number of excellent resources listed below.

Mailing Lists
- *Includes a user and dev mailing list: http://*cfeclipse.tigris.org/servlets/ProjectMailingListList

Blogs and Wikis
- *Home of the nightly builds:* www.spike.org.uk/blog/index.cfm
- *The CFEclipse blog/wiki site:* www.spike.org.uk/cfeclipse:
- *Mark Drew has some interviews and musings:* http://cybersonic.blogspot.com/

Resource Sites
- *The main project site:* http://cfeclipse.org
- Nathan Strutz's CFEclipse resource page: www.dopefly.com/pages/cfeclipse.cfm
- Videos and demos: www.rohanclan.com/projects/cfeclipse/index.cfm

---

About the Authors

Simeon Bateman is director of development for Business Transitions LLC of Portland, OR. Simeon is a Certified Macromedia ColdFusion developer and has been utilizing Macromedia ColdFusion daily for the last 5 years.

simeon@simb.net

Spike has been working with ColdFusion since 1998, and has worked on some of the largest ColdFusion projects in Europe and Australasia. He is a Macromedia Certified Instructor and member of Team Macromedia for ColdFusion. He is currently the lead developer for the CFEclipse project.

spike@spike.org.uk

> ## "The concept of synergy is an amazing thing. The idea is that the sum of the contributions of the group exceeds those of individual developers"

FLASHFORWARD
& FLASH™ FILM FESTIVAL

**SAN FRANCISCO 2005, APRIL 6-8
HERBST THEATRE**

THE WORLD'S PREMIER FLASH EVENT

The 13th Flashforward conference and Flash™ Film Festival, the largest and longest running Flash user conference in the world, shares the latest in design, development, education and inspiration.

FLASHFORWARD 2005 FEATURES:
- **4 In-Depth Flash Workshops**
- **25 One-Hour Seminars**
- **20 Q&A Sessions**
- **17 "Ask the Experts" Sessions**
- **11 Technology Showcase Seminars**
- **Flash™ Film Festival**
- **Macromedia Keynote**
- **Exhibition Area**
- **Meet the Speakers and Flash Authors Reception**
- **Networking Receptions**
- **Exclusive Conference Workbook with Speaker Notes**

REGISTRATION:
**$599 through March 4
$699 through April 5
Additional $50 Macromedia User Group discount
$799 at the door**

ADDITIONAL INFORMATION:
- **Call toll free 1.877.4.FLASH.4**
- **www.flashforwardconference.com**
- **1.805.640.6679 outside the US and Canada**