SPECIAL INSERT: WEB SERVICES CONFERENCE & EXPO Pg. 41

U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

ColdFusionJournal.com

February 2003 Volume: 5 Issue: 2

RETAILERS PLEASE DISPLAY UNTIL APRIL 30, 2003

$8.99US $9.99CAN

03>

0 71486 02689 1

SYS-CON MEDIA

ColdFusion Developer's Journal

editorial advisory board

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

editorial

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

production

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

contributors to this issue

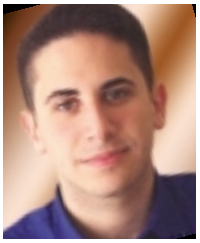
Charlie Arehart, Selene Bainum, Vince Bonfanti, Ben Forta, Hal Helms, Simon Horwith, Kevin Schmidt, Bruce Van Horn, Ron West

editorial offices

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

Product Review Central

You'll notice more product reviews than usual in the next few issues of ***ColdFusion Developer's Journal***. We've got three in this month's issue alone. This is simply the direct result of more CF-related products on the market. It's a great thing for all of us, because many of them can help make our lives easier. There are lots of good add-ons out there, as well as full-featured products running off of ColdFusion. In addition, many of the existing products that we've all used for years have recently been upgraded to take advantage of the new features of ColdFusion MX.



By Robert Diamond

Since the holidays I've also participated in a number of conference calls about product previews. If you ask me, the first half of 2003 is looking very exciting. As developers, none of us like to reinvent the wheel, and there are some good solutions out there to make things easier both for clients/users, and on the server end. All are worth a good look.

This month's issue of *CFDJ* covers some interesting topics that people have been requesting from us – more on Web services, more on optimizing our code, more on best practices, more general programming tips, and advice on dealing with management.

Our **Tales from the List** column by Simon Horwith covers one of the hot topics on the CFDJ-List this month – the use of such "newbie" tags as <CFINSERT>, <CFUPDATE>, and <CFFORM>, etc. Some important pros and cons are discussed.

Later in the issue, Simon talks about a new Macromedia course to help us CFers get a little bit more into the world of Flash. Vince Bonfanti writes about BlueDragon, and where it fits into the CFML world. Ben Forta's column covers what tags and a tool will give you the best performance. (FASTER! MUST GO FASTER!) Sorry, slipped into server admin mode for a moment… Ron West stretches Web services to its limits. Bruce Van Horn revises a previous response, and answers a new question as part of our Q&A column, and Hal Helms tells Part 1 of "A Developer's Story," covering the mystery of successful revisions. Charlie Arehart introduces us to servlet filters in CFMX.

Back to those product reviews I said we'd be having more of… Kevin Schmidt reviews HTML-DOC, from Easy Software Products. Selene Bainum reviews NQcontent V2 from NetQuest, and inFusion Mail Server from On-Line Data Solutions. So if you're interested in the potential of generating PDFs on the fly, a new content management system, or a CF-based mail server, these reviews are for you!

In an attempt to make at least one of my New Year's resolutions come true, I've broken down and registered for a blog. Stay tuned for next month's editorial when, assuming I keep up with it (and, ahem – start writing in it), I'll let you know, and will list the URL.

A request to all of you for the new year – if you haven't yet checked out SYS-CON's new developer portal at http://developer.sys-con.com, do so now. You can pose questions to the expert writers and readers of all of our magazines, as well as participate in some great discussions with others in the know. We've created a 24/7/365 community devoted solely to the developers SYS-CON reaches every month through our *i*-technology publications.

You'll also notice that in this month's issue we've included the full brochure for our Web Services Edge 2003 East – International Web Services Conference & Expo, the world's largest independent Web services, Java, .NET, and XML event, which is being held March 18–20, 2003, at the Hynes Convention Center, Boston, MA. For those of you interested in expanding your knowledge past the world of CF, or those who already have and want to further your education, check it out. We've got lots of great speakers, keynotes, and information-packed tracks. Hope to see you there!

About the Author

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

robert@sys-con.com

SYS-CON MEDIA

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

business development
vp, business development
Grisha Davida grisha@sys-con.com
COO/CFO
Mark Harabedian mark@sys-con.com

advertising
senior vp, sales & marketing
Carmen Gonzalez carmen@sys-con.com
vp, sales & marketing
Miles Silverman miles@sys-con.com
advertising director
Robyn Forma robyn@sys-con.com
advertising account manager
Megan Ring-Mussa megan@sys-con.com
associate sales managers
Carrie Gebert carrieg@sys-con.com
Kristin Kuhnle kristin@sys-con.com
Alisa Catalano alisa@sys-con.com

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

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

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

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

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

Tales from the List

<cfinsert>, <cfupdate>, <cfform> oh my!

One of the most enjoyable aspects of being an active contributor to the CFDJ List is having the opportunity to help developers new to ColdFusion. New developers greatly appreciate the help from "seasoned professionals," and often ask questions that are general enough to ignite lengthy debates over best practices as well as stories about situations and solutions that List members have encountered in the past.

By Simon Horwith

Newcomers to ColdFusion also tend to follow up answers to the questions they post with further questions, which not only leads to more detailed discussion threads but also gives insight into the "mind of the newbie."

Recently, there was a thread on the **CFDJ List** regarding "newbie" tags – specifically, <CFINSERT>, <CFUPDATE>, and <CFFORM>. The thread wasn't actually begun (or participated-in for that matter) by newcomers to ColdFusion, but it did bring up a recurring discussion on the List.

The thread began with a post by Benoit Martin who had discovered that ColdFusion MX now includes a .js file in order to perform form validation when using the <CFFORM> tag. He was curious as to how to configure a ColdFusion Server hosting multiple Web sites so that each site is able to use the <CFFORM> tag without error. He had tried creating ColdFusion mappings that point to the directory without success.

Morgan Kelsey replied to the post reminding Ben that the code generated by the tag relies on a Web site mapping to find the "/CFIDE/scripts/" directory, and that a virtual directory pointing at this folder would need to be created for each site – on the Web server. Ben replied that this would be as much trouble as copying the folder to each site directory and asked if there was any better solution, which led to a posting by ".jeff" stating that Ben should "give up using the lame <CFFORM>, <CFINPUT>, <CFINSERT>, <CFUPDATE> and other newbie CF tags."

Ben defended his use of the tags stating that "the validation scripts written by CF when using those tags always worked perfectly good" for him and that he isn't "trying to look fancy or impress anyone" and that he is just trying "to get my job done as fast and efficiently as possible and those tags help me do so in most cases."

This is a recurring message that more experienced developers send to ColdFusion "newbies" – but is it accurate? The purpose of <CFINSERT> and <CFUPDATE> is to automate the generation of SQL INSERT/UPDATE statements. While these tags may save a little development time, this savings comes at a cost. Before examining the cost of using these tags, bear in mind that in the average application, most INSERT and

—continued on page 30

About the Author

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

shorwith@figleaf.com

PDFs on the Fly

This is great, but can I get it in a PDF?

If you've ever generated any type of custom report using ColdFusion, odds are you've fielded this question at some point in time. And while the task may sound rather difficult, it can be made relatively simple with a particular open-source third-party tool.

By Kevin Schmidt

HTMLDOC

The tool, offered by Easy Software Products (www.easysw.com), is called HTMLDOC. The company has made this tool available through the GNU General Public License, so you can download and use it for free. Among the many things HTMLDOC allows you to do is convert an HTML file to a PDF. HTMLDOC is a normal Windows application, but it has a command-line interface that allows you to access the application with ColdFusion. So, by utilizing CFCs, the <cfexecute> tag, and the <cffile> tag, you can create a simple, easy-to-use component that will take your HTML content and turn it into a PDF.

Installing HTMLDOC

The first step toward this PDF utopia is to download HTMLDOC. To do so, simply visit the Easy Software Products home page. You can install the program anywhere you like, but the default installation is in the program files directory. For the purpose of this article, I installed HTMLDOC into the CFusionMX directory. Once you have the program installed, double-click the HTMLDOC icon. If your installation was successful, you should see a screen that looks like Figure 1.

The GeneratePDF Component

Your first step in terms of ColdFusion coding is to create a component to handle the generation of the PDF. This piece of code will potentially be utilized by several different parts of an application, thus making it a component makes good sense. In order to create this component, you simply use the <cfcomponent> tag. The following code, saved as GeneratePDF.cfc, will create the GeneratePDF component:

```
<cfcomponent>

</cfcomponent>
```

Note, however, that a component without any methods will do you no good. To create a method, you'll need to use the <cffunction> tag. In this case, the method will be called "MakePDF". The following code creates it:

```
<cffunction name="MakePDF" returntype="struct">

</cffunction>
```

The name attribute of the <cffunction> tag gives the method its name, and the returntype specifies that this particular method will return a structure to its caller.

The next step is to set up the arguments for the method. Several pieces of information are needed to generate a PDF. First and foremost is the location of the HTMLDOC .exe file (gHTMLDOC.exe). You also need to know what the HTML content of the PDF is going to be, where to save that content, and what to name the file you've saved it into. In addition, there's a variable that determines whether to delete the HTML file that's been generated. Also needed are the path to where you want to save the new PDF, the name for the PDF file, and any options for formatting the PDF. HTMLDOC allows for a multitude of formatting options such as font size and font face; for a complete breakdown of these options, refer to the HTMLDOC user manual.

By using the <cfargument> tag, you can specify whether or not an argument must be passed in from the calling template. For this particular method, only one argument is required – the HTMLContent argument. All the other arguments may be passed in, but not necessarily; and those that aren't passed in have default values assigned. See Listing 1 for the complete set of <cfargument> tags for this method.

Bring on <cffile>!

Now that all your arguments are either passed in or set to the default values, you're ready to create the HTML document. To do so, you'll utilize the <cffile> tag. For this example, you'll use the <cffile> tag to write a file with the name specified in the HTMLFileName argument to the location specified in the HTMLFilePathArgument containing the content of the HTMLContent argument. Got it? If a file by that name already exists, simply overwrite that file. The code below illustrates what your <cffile> tag should look like:

```
<cffile
action="write"
file="#arguments.HTMLFilePath##arguments.HTMLFileName#" output="#arguments.HTMLContent#"
nameconflict="OVERWRITE"
>
```

Now that your file is written, it's ready to be converted to a PDF. This is where HTMLDOC takes over. Because HTMLDOC is an executable program that's installed on the server, you need to use <cfexecute> to access it. The <cfexecute> tag, by definition, executes a specified process on the server. For this example, the <cfexecute> tag will require three attributes: name, arguments, and timeout. The name attribute specifies what process is to be run –

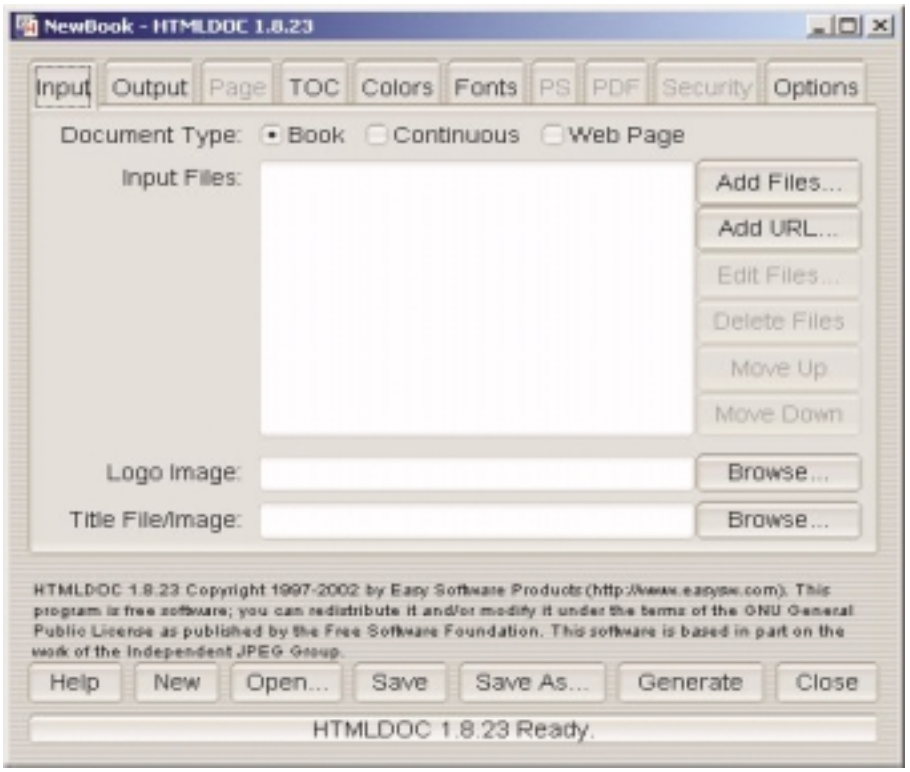


Figure 1: HTMLDOC installation

in this case, the location of the gHTMLDOC.exe file on the server. The arguments attribute holds the command-line variables passed to the application. These will give HTMLDOC any options for the PDF it's creating, tell it where to store the PDF once it's created, and tell it what file to use in creating the PDF. The final attribute, timeout, specifies the amount of time ColdFusion should wait for output from the process. All of the values for these attributes, except two that are required every time you generate a PDF, are present in the arguments scope. Thus, the <cfexecute> tag would look like the following:

```
<cfexecute
name="#arguments.HTMLDOCPath#"
timeout="5"
arguments="#arguments.PDFOptions# --webpage --f
#arguments.PDFFilePath##arguments.PDFFileName#
#arguments.HTMLFilePath##arguments.HTMLFileName#
"
   >
</cfexecute>
```

The two pieces of the arguments attribute, --webpage --f, are the two pieces that are required each time.

Now that you have your PDF, you're ready to do some cleaning up and return information to the calling template. If you didn't pass in the HTMLDelete argument as "no", the default "yes" will be assigned. When the HTMLDelete variable is yes, the following code to delete the HTML file is run:

```
<cfif arguments.HTMLDelete>
<cffile action="delete"
file="#arguments.HTMLFilePath##arguments.HTMLFil
eName#">
</cfif>
```

Now that the original HTML file is gone, it's time to pass back all the pertinent information to the calling template. The information you're going to return is the path to where the HTML file was saved, the name of the HTML file, the path to where the PDF was saved, the name of the PDF, and whether or not the HTML file was deleted. All of these values will be saved in a structure called PDFData. The code to set up this structure is as follows:

```
<cfset PDFData.HTMLFileName =
arguments.HTMLFileName>
<cfset PDFData.HTMLFilePath =
arguments.HTMLFilePath>
<cfset PDFData.PDFFileName =
arguments.PDFFileName>
<cfset PDFData.PDFFilePath =
arguments.PDFFilePath>
<cfif arguments.HTMLDelete>
<cfset PDFData.isHTMLKept = 0>
<cfelse>
   <cfset PDFData.isHTMLKept = 1>
</cfif>
```

Return the Values

Once the structure is set up, the <cfreturn> tag is used to return the structure to the calling template. The <cfreturn> tag can only return one variable – which is precisely why it was necessary to set up the PDFData structure. The code below returns the PDFData structure to the calling template:

```
<cfreturn PDFData>
```

Using the GeneratePDF Component

Now that you have the complete component necessary to turn your HTML files into PDFs (see Listing 1 for the complete component), how do you harness this power? Creating a component makes creating the PDFs as easy as invoking the method. If you're wondering how to get the data for the HTML content variable, there are a couple of ways to do it. One involves working with <cffile> again, and the other involves <cfhttp>.

Working with <cffile>

Just as <cffile> is able to write files, it is also able to read them. When you read a file, you need to specify three things: the action you're performing (in this case the action is "read"); the full path of the file you want to read; and the name of the variable into which you want to store the content of the file. Thus, if you were to use the <cffile> tag to get the data for the HTMLContent variable, it might look something like this:

```
<cffile action="read" file="C:\CFusionMX\www-
root\PDF\testing.cfm" name="HTMLContent">
```

Now you have the variable that holds the data to pass in to the GeneratePDF component.

Working with <cfhttp>

Another way to get the data to pass into the GeneratePDF component is to use <cfhttp>, which will execute an HTTP and get and return the file for you. For example, you could use <cfhttp> to grab a page from your favorite Web site and turn

it into a PDF. For our example, though, the same file, testing.cfm, will be used. When using <cfhttp>, two attributes will be specified: method and URL. The method attribute is either "post" or "get", and in this case you'll specify "get" since you're retrieving a URL. If you prefer, you could leave off the method, as "get" is the default; for my money, though, I prefer to specify it. The URL to be used is http://localhost:8500/testing.cfm. This makes the assumption that you're running ColdFusion MX with the standalone Web server. So, in order for your <cfhttp> tag to work, it needs to look like the following:

```
<cfhttp method="get" url="http://local-
host:8500/PDF/testing.cfm">
```

Invoking the MakePDF Method

Now that you have two ways to generate the data, you're ready to use the MakePDF method of the GeneratePDF component. To do so, employ the <cfinvoke> tag. This tag will both instantiate the GeneratePDF component and make the call to the MakePDF method. For this example, there are three attributes that need to be used with the <cfinvoke> tag: component, method, and returnvariable. The component attribute is the reference to component; the method attribute specifies which method you're calling; and the returnvariable specifies the name to give the variable that is returned. The following code shows how the <cfinvoke> tag should look:

```
<cfinvoke component="PDFGen" method="MakePDF"
returnvariable="PDFData">

</cfinvoke>
```

Passing in Arguments

Obviously you'll need to pass in your arguments – specifically the data that you want turned into a PDF. You can pass arguments to a component in one of three ways. As attributes of the <cfinvoke> tag; using the <cfinvokeargument> tag; or passing in a structure in the argumentCollection attribute of the <cfinvoke> tag. For this example the <cfinvokeargument> tag is used. The <cfinvokeargument> tag takes two attributes, "name" and "value" – "name" being the name of the argument and "value" being its value. Thus, in order to pass in the one required argument of the MakePDF method, you could use the following code:

```
Using <cfhttp>:
<cfinvokeargument name="HTMLContent"
value="#cfhttp.filecontent#">

Using <cffile>:
<cfinvokeargument name="HTMLContent"
value="#HTMLContent#">
```

And that's all there is to it; your component and HTMLDOC will do the rest. Once your component is finished executing, you can output the returned values to see where your HTML and PDF files were written and what they were named, as well as whether the HTML file was kept. The following code will make this happen:

```
<cfoutput>
HTML File Name: #PDFData.HTMLFileName#<br>
HTML File Path: #PDFData.HTMLFilePath#<br>
PDF File Name: #PDFData.PDFFileName#<br>
PDF File Path:
#PDFData.PDFFilePath#<br>
Was the HTML Kept:   <cfif
PDFData.isHTMLKept>Yes<cfelse>No</cfif>
</cfoutput>
```

What you choose to do with your new PDF is up to you. But perhaps you'll want to tell the pointy-headed one that his PDF might take a few days. Then maybe you'll sit back, relax, and catch up on a few of those blogs you've been meaning to get to.

About the Author

Kevin Schmidt is a Macromedia Certified ColdFusion Developer and is part of Team Macromedia. He is the author of Macromedia ColdFusion MX: Training from the Source, *published by Macromedia Press. He is also the manager of the Des Moines Macromedia User Group.*

schmidt@hungrycow.com

Listing 1

```
<cffunction name="MakePDF" returntype="struct">
<cfargument
name="HTMLDOCPath"
required="no"
default="C:\CFusionMX\HTMLDOC\gHTMLDOC.exe"
>
<cfargument
name="HTMLContent"
required="yes"
>
<cfargument
name="HTMLFilePath"
required="no"
default="C:\CFusionMX\wwwroot\PDF\"
>
<cfargument
name="HTMLFileName"
required="no"
default="#DateFormat(Now(), 'ddmmyyyy')#_#Timeformat(Now(),
'hhmmss')#.htm"
>
<cfargument
name="HTMLDelete"
required="no"
default="1"
>
<cfargument
name="PDFFilePath"
required="no"
default="C:\CFusionMX\wwwroot\PDF\"
>
<cfargument
name="PDFFileName"
required="no"
default="#DateFormat(Now(), 'ddmmyyyy')#_#Timeformat(Now(),
'hhmmss')#.pdf"
>
<cfargument
name="PDFOptions"
required="no"
default=""
>
</cffunction>
```

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

NQcontent V2 from NetQuest

A CMS for the developer

Every day it seems as if another content management system (CMS) is popping up on the horizon, many of them built using ColdFusion.

By Selene Bainum

Typically, these CMSs contain the same type of functionality. They:

- Allow users who don't know HTML to add and edit content easily
- Display content using a template system and integrated style sheets
- Create user accounts and assign security permissions allowing certain users to perform only certain tasks
- Allow content to be indexed and subsequently searched by the user
- Modify system settings that control certain functionality of the site
- Perform workflow and/or versioning control over content

Covering the Basics

NQcontent performs all this functionality with aplomb. The Articles Management content is displayed in a hierarchical tree format, with unlimited nodes and levels. Available options are displayed with DHTML – which only works properly in Internet Explorer – making selections quick and easy. The navigation within multi-page forms could be a little clearer, especially when required fields are on another page.

At review time, the only form validation within the administrator uses ColdFusion's built-in server-side validation, but that will change soon. As with most CMSs, content can be edited using the popular Ektron editor that allows a text area field in Internet Explorer to act as an HTML authoring tool.

The security permissions are pretty granular, assigning individual users to one or more groups, and permissions are assigned on a group basis. Groups can be assigned permissions to one of dozens of available actions, allowing for fine-tuned security settings.

The Systems section of the administrator allows for management of application variables, content types, search indexes, and system settings.

Above and Beyond

While NQcontent is similar to other CMSs, it is the additional features that really set it apart. Like ColdFusion, NQcontent runs not only on Windows NT, but also on Unix and Linux. It supports Oracle as well as Microsoft SQL Server, making NQcontent a viable solution for enterprises that have not jumped on the Microsoft bandwagon.

Both the Professional and Enterprise editions of NQcontent will run on the Professional editions of both ColdFusion 5 and ColdFusion MX. If you are using Oracle, however, you will need ColdFusion Enterprise.

NQcontent also comes with several modules that allow developers and administrators alike the ability to manage files, data, and forms via the administrator interface.

Manage Files Remotely

Often, the only access you have to a Web site you are developing is via ColdFusion's Remote Development Services (RDS). Anyone who has ever tried to perform file operations over RDS or tried to copy multiple or large files knows how limiting it can be. NQcontent's built-in File Manager resembles a Web-based version of Windows Explorer. From it you can create, rename, and delete directories and create, rename, edit, delete, and upload files. Since the back end of the manager uses CFDIRECTORY and CFFILE tags, certain operations are not available, such as deleting a directory before deleting the files within it.

This is a simple, yet useful tool that will allow you to more easily manage your Web site. The only directories and files that are accessible by this tool are within the default Web root, however, so virtual directories and file stores outside of your Web root will not be accessible.

Manage Your Data

The backbone of any ColdFusion Web site is certainly its database. Typically, database management is completely separate from your development environment, using tools such as those that come with Microsoft SQL Server and Oracle. If you want to have a graphical interface to the database you don't have many other options. NQcontent's database administration module contains graphical interfaces to database tables and records that are similar to those provided by Quest software in their Oracle database tool TOAD, one of the best on the market.

The database and table data is stored by NQcontent as metadata. You add a table to NQcontent by specifying the data source and table name. NQcontent will then import the column names, and you can modify attributes as well as view, edit, update, and delete rows from tables. You can also specify advanced information such as permissions, triggers, and SQL scripts that should be run at specified times.

The import functionality does not determine table keys and identities, so they must be set separately. One drawback with the tool is that it does not allow the metadata to store more than one primary key column for any table, making workarounds necessary for tables that use combination keys – one or more columns that, combined, create a unique row identifier.

When tables are added to NQcontent, their columns are available to use when creating content such as articles and forms. User-friendly aliases can be assigned to table and column names and these aliases will be displayed throughout the NQcontent administrator. This allows administrators unfamiliar with the database-naming convention to be able to tell what data they are working with.

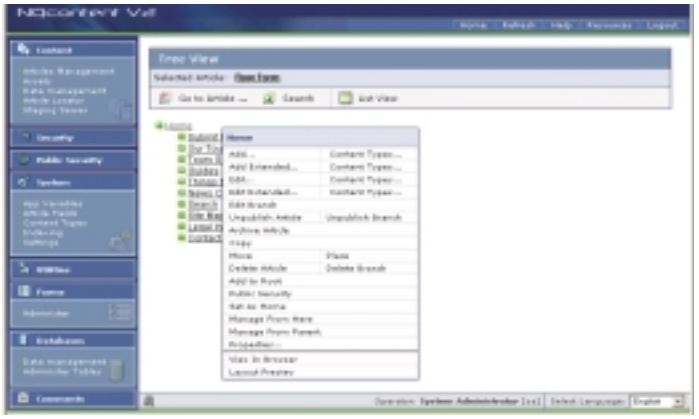


Figure 1: DHTML menu for content options

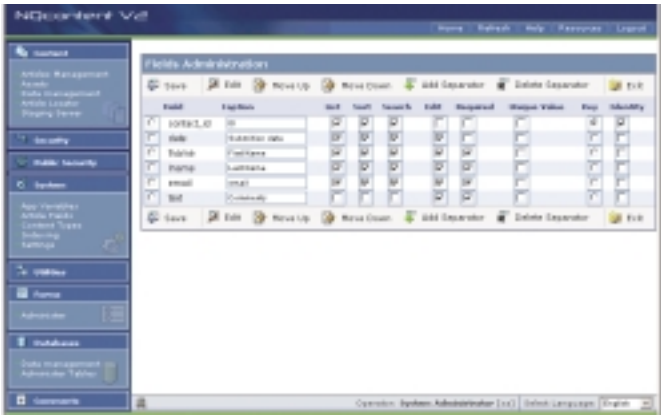


Figure 2: Management of database table fields

Advanced developers, or those with good database skills, might find the interface a bit cumbersome. It is certainly quicker and easier to use a database management application as opposed to using one built within a Web page. If you do not include your Web site's tables in the administrator, those tables and fields will not be available to use in the content administrator. However, if you want nontechnical administrators to access database data, using this tool will certainly save you a lot of time and effort. Most CMS systems do not even offer this flexibility.

Build Forms Effortlessly

Many companies that offer solutions based around CMSs will often charge the client extra to build basic forms, such as Contact Us or Comments. NQcontent comes with a forms administrator module that allows junior ColdFusion developers and even power users to create forms whose submission results can be e-mailed and/or inserted into a database, all without touching any code.

Form fields can be created from scratch or imported from an existing database table defined in the database module. When you add a field to a form, you can specify attributes such as the field type (text box, text area, checkbox, etc.), specify values for multiple options, and set default values. You can even set design options such as where the field labels display and the width of the fields. This functionality will save the time of advanced developers and save money on labor costs.

By using just the GUI forms interface, the form validation is a little limiting. At review time, the only included form validation was performed by ColdFusion's built-in server-side processing, using hidden form fields with suffixes such as _required to ensure that required fields are filled in. Patches have since been applied to include basic JavaScript form-field validation and more advanced JavaScript validation support will be included in version 3.0.

Intermediate and advanced ColdFusion developers can enhance the server-side form validation with their own code using some of the included NQcontent custom tags. This allows more in-depth forms to be created and validated and also allows forms built using the interface to update database records as well as create them.

But Wait, There's More

The Professional edition comes with all of the functionality listed above, as well as some additional administration and management features. The Enterprise edition also contains workflow, content versioning, and a more granular security model that assigns roles as well as groups to specific users. NQcontent also offers add-on products such as Forums, E-commerce extensions, Visitors Guest Book, Forms, and Banner and Polls Managers.

Overall, this is a good CMS that will be used by all levels of users, from nontechnical administrators who are only editing content, to advanced developers managing databases and customized forms. If you are thinking about working with a CMS system, NQcontent is definitely one to take a closer look at.

About the Author

Selene Bainum has been developing with ColdFusion since 1996. She created and maintains www.webtricks.com, a premier ColdFusion tutorial site.

selene@webtricks.com

Vitals: NQcontent V2

NetQuest
8 Duncannon Street, Strand
London WC2N 4JF
Tel: +44 20 7484 5028
Fax: +44 20 7484 5128
info@nqcontent.com
www.nqcontent.com

Test Environment:
HP Pavillon zt1180, 1.2GHz P-III, ColdFusion MX, 512MB RAM, Windows XP Professional, MS SQL Server 2000
Pricing: NQcontent pricing is based on the number of Content Contributors, and starts from $25,000 or $40,000 for the Enterprise Version; both include unlimited database connectors.

Product Snapshot
Target audience: Medium to large organizations looking to create corporate Web sites, portals, content syndication, or e-commerce sites.

Pros:
- Multiple platform support
- Simple to install and deploy
- Easy for nontechnical administrators to modify content
- Advanced features for developer use
- The ability to create forms and manage files and data

Cons:
- Lack of JavaScript or advanced form error validation (at review time)

Client platform:
System operators must use IE 5.x + on PC
Server OS: Windows 2000, Red Hat Linux 7.2, Sun Solaris 8
ColdFusion: Versions 5.0 or MX
Database support: MS SQL Server, Oracle

Ask the Training Staff

I don't know about you, but February is going to be a very busy month for me. Not only is there plenty of CF work to keep me busy, but my wife, my son, and I are eagerly awaiting the birth of baby boy number two!

Nevertheless, I have questions to answer and one revision to make on a previous answer. I hope you find it all worthwhile.

By Bruce Van Horn

First let's deal with the revision of my answer back in December (Vol. 4, issue 12) regarding the use of a placeholder in a query. The question was really centered around writing dynamic queries and how to handle the question of what should go immediately after the WHERE clause. In my example (see Listing 1), I suggested "WHERE Name = 'PlaceHolderText'". I rationalized that nobody's real name is actually "PlaceHolderText" and therefore it would have no impact on the query except to allow for the dynamically generated OR clauses that followed.

Many of you were kind enough to send me notes about a much better way to do the same thing and I agree with all of you. My only excuse was pure laziness in not wanting to address the subject of using constants in SQL since the person who originally asked the question had very little knowledge of SQL. Nevertheless, the subject is worth addressing.

The problem with using my example of creating a placeholder is that it actually does search through the database table to see if someone has the name "PlaceHolderText". This uses unnecessary resources and may actually (though I certainly hope not) return a record if that were indeed somebody's name.

The better way to do this is to use a comparison of two constants. In our example, we want a placeholder that would not return any records (these would be returned by the data in our OR clauses), so it would be best to start with "WHERE 0=1" (see Listing 2). Both 0 and 1 are constants. 0 will never equal 1 and therefore will not return any records. It also creates no interaction with the database, thus saving resources. If we were running a query that needed to initially return all rows and then filter out results through dynamically generated AND clauses (see Listing 3), you would simply use "WHERE 0=0".

This is a very handy trick to keep in your bag of SQL tips and tricks if you didn't already know it. Thanks again to the many of you who caught my laziness and didn't tolerate it! It's good to know somebody out there is reading and keeping me humble!

Q: I have an application that sends out a lot of e-mail. My problem is I have to check the server a couple of times a day to drag mail from the undeliverable folder back into the spool folder to resend it. When I look at the log file, there are entries that say "No connection to mail server" but there is nothing wrong with the e-mail itself. When I drag them back to the spool folder they usually go out with no problem. Any ideas about this, or is there a way to automatically make CF respool undeliverable mail?

A: I feel your pain! I have had the very same problem with CF for a long time. I'm not sure what causes it to not connect and I'm not aware of any way to get CF to automatically respool the mail, but there is a way to accomplish the same thing. What I've done is this: I created a simple CF page that checks the mail.log file to see if there are any entries that contain the phrase "No connection to mail server". If there are any entries, I locate the file in the "Undelivr" folder and move it back to the "Spool" folder. See Listing 4 for an example of my Respool_Email.cfm page. Then, I added this page to the Scheduler in the CF administrator to execute every 10 minutes. Now if there are any failures to send the mail due to a connection problem with the mail server, those messages are automatically respooled.

Q: I need to retrieve records from our SQL Server database and randomize the order in which the records are displayed. I'm sure I could figure out a way to do this in CF by manipulating the result set, but is there a way to get SQL Server to randomize the results that it gives CF?

A: Yes, there is a very easy method that I have been using that works well for SQL Server queries. All you need to do is add "ORDER BY NewID()" to your query. The NewID() function (this is a SQL Server function, not a CF function!) generates a unique identifier (like CF's CreateUUID() function) for every record in the result set, then sorts by that unique ID. For example, Listing 5 shows a query that would randomly select 10 products from my products table.

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

About the Author

Bruce Van Horn is president of Netsite Dynamics, Inc., a Certified ColdFusion Developer/Instructor, and a member of the CFDJ *International Advisory Board.*

Bruce@NetsiteDynamics.com

Listing 1:

```
<!--- my original suggestion --->
<CFQUERY NAME="qSearch" DATASOURCE="#Request.DSN#">
    SELECT * FROM Table
    WHERE Name = 'PlaceHolderText'
           <CFLOOP LIST="#Form.SearchText#" INDEX="SearchItem">
                    OR Name LIKE '%#Trim(SearchItem)#%'
           </CFLOOP>
</CFQUERY>
```

Listing 2:

```
<!--- a better way to handle dynamic ORs --->
<CFQUERY NAME="qSearch" DATASOURCE="#Request.DSN#">
    SELECT * FROM Table
    WHERE 0 = 1
           <CFLOOP LIST="#Form.SearchText#" INDEX="SearchItem">
                    OR Name LIKE '%#Trim(SearchItem)#%'
           </CFLOOP>
</CFQUERY>
```

Listing 3:

```
<!--- a better way to handle dynamic ANDs --->
<CFQUERY NAME="qSearch" DATASOURCE="#Request.DSN#">
    SELECT * FROM Table
    WHERE 0 = 0
           <CFLOOP LIST="#Form.SearchText#" INDEX="SearchItem">
                    AND Name . . .
           </CFLOOP>
</CFQUERY>
```

Listing 4:

```
<CFIF FileExists("D:\cfusion\log\mail.log")>
<CFFILE ACTION="MOVE"
        SOURCE="d:\cfusion\log\mail.log"
        DESTINATION="d:\inetpub\netsite\htdocs\mail.log">
<CFHTTP URL="http://www.netsitedynamics.com/mail.log"
        METHOD="GET"
        NAME="qMailLog"
        TEXTQUALIFIER="""
        DELIMITER=","
        RESOLVEURL="false"></CFHTTP>
<CFOUTPUT QUERY="qMailLog">
    <CFSET Start = FindNoCase("No connection to mail  server",qMailLog.Message)>
    <CFIF Start gt 0>
           <CFSET FileStart = FindNoCase("d:\CFUSION\Mail\undelivr",
                    qMailLog.Message)>
           <CFSET FileName1 = Mid(qMailLog.Message,FileStart,(Len(qMailLog.
                    Message)-FileStart))>
           <CFIF FileExists(FileName1)>
                    <CFSET FileName2 = ReplaceNoCase(FileName1,"d:\CFU
                      SION\Mail\undelivr","d:\CFUSION\Mail\spool")>
                    <CFFILE ACTION="MOVE"
                   SOURCE="#FileName1#"
                   DESTINATION="#FileName2#">
           </CFIF>
    </CFIF>
</CFOUTPUT>
Yes, moved files
<CFELSE>
No files exist
</CFIF>
```

Listing 5:

```
<CFQUERY NAME="qGet10Prods" DATASOURCE="#Request.DSN#">
    SELECT TOP 10 ProductID, Cat_No, ProductName, Description, List_Price
    FROM  Products
    ORDER BY NewID()
</CFQUERY>
```

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

A Developer's Story Part 1

The mystery of successful revisions

Actually, it's no one's fault." This was the conclusion I had come to after a week of working on a nightmare project I had been called in on. The CIO had asked me for a briefing on what I had found so far.

By Hal Helms

"But how could this be such a mess?" the CIO wanted to know. "Granted, the project involves a sizable change of an existing piece of software, but that was written only three years ago. I've got 1,500 customers waiting to use this new version. It's very high profile – and we can't get it out the door. Costs are crazy. I don't even want to think how far above budget we are. We've missed so many ship dates that the developers won't even guess when it might really be ready."

The client was a large one and had a good-sized development shop dedicated to projects inside the company. I had been asked to come in to triage the situation – to determine what was workable, what wasn't, and to lay out a plan that could be followed going forward. As part of that effort, I needed to understand how the client developed software. So putting on my Sherlock Holmes cap, I began to snoop around.

The Best Intentions

What struck me first was the degree to which everyone involved had tried their best. This was not a case of people who didn't care or internal politics that had sabotaged development efforts. To a remarkable degree, the entire cast of characters had done their best to ensure success.

"We sat in meetings to determine what the users wanted the software to do," one of the lead developers told me. "We worked up prototypes; we sat with the users and watched them work. I honestly don't know what we would do differently if we were to tackle it again today. I think that the users just didn't know how to tell us what they really wanted."

I thanked him and went to see the project lead of the initial project. "It was a big success," she told me. "The CEO and CIO both made a big deal of the project – about different teams working together. For a while, we were the poster kids of how to innovate within a corporate environment. The developers were great. They kept going over things with us until we were sure we had them just right. They offered great suggestions and kept us on track. Really, it was a pleasure to work with them – and that's something different from my normal dealings with developers. I don't know what the problem is now."

She was not the project lead on the revision. "Could it be the new project manager maybe isn't experienced enough?" I asked.

"No," she told me. "Whatever it is – that's not it. I've known and worked with her for almost 10 years and she's top notch. She's rolled out at least seven major new projects in that time and every one of them has been a success. In fact, she helped me on a volunteer basis on the initial version of the software that was such a success. She's really good."

Something in what she said tugged at the back of my mind. What was it, exactly? "How long has ABC been in business?" I asked.

"About 12 years," she said. I started with them when they were about a year and a half old."

I had an idea I wanted to follow up on. "How many new projects would you say you've done in that time?"

"Maybe 10 or 12 – something like that. But the first version of this was the biggest by far."

"And the current project manager has another seven or so? All successful?" I asked.

"Yes, why?"

Gaining Insight

"I'm just trying to get a good view of the situation," I told her, and left. What I saw was a situation in which a company can produce new applications – but can't maintain existing ones.

"Have you found the culprit?" the CIO asked me, running into me in the hall.

"Maybe," I replied mysteriously. "Do you have time to talk on Friday?"

"I never have time," he said, "but I'll make it. My assistant will give you a time."

I spent the rest of the week ruling out other possibilities. The code was of good quality, though documenting the code was often relegated to new developers whose understanding of the entire system (as opposed to an individual code page) was, understandably, superficial. Most of the code comments were fairly obvious: "Loop over result set" – that sort of thing.

I needed to confirm one suspicion and for that I had to wait until Friday morning when the original system's architect came into the office. I took Brian out to lunch. "Say, when you were originally architecting the application, how did you go about it?" I asked him.

"Well, we talked with the clients – the users – of course. We found out what they wanted the system to do and then we designed components around that functionality. We did use cases and then broke the system up into various functional components. I dunno why they're having such problems with it now. I guess we must have missed stuff, but really, I thought we had everything they wanted."

I assured him that it appeared they had done an excellent job of meeting all the user requirements. "Judging by the accolades everyone got – deserved for all your hard work – I think you did great. This problem isn't one of bad apples spoiling the barrel. I suspect it may be just the opposite."

Before we had a chance to talk further, Brian's cellphone rang and he had to cut our lunch short to handle some urgent business. I finished typing up a preliminary report and left to make my 2 p.m. appointment with the CIO.

No One to Blame

"Actually, it's no one's fault," I told the CIO. In response to his question of how they could have such a mess on their hands with no one to blame, I explained what I had observed. "The problem is with the system in place that produces software. And everyone is so close to that system, they don't even see it as a system – as one choice among others. To them, it's just the way to build software."

"And what is that system then?" asked the CIO.

"They're following a process of functional decomposition – of determining the things that need to be done and then designing components – high-quality components, I want to add – to fulfill those functions."

"Sounds right," he agreed.

"But the problem, though, is that the functions that are needed, change. Now, we all know this and we tend to blame our clients for their fickleness: 'If only they would tell us what they want!' But how can they? The clients aren't changing their minds randomly – at least not usually.

"Their changing requirements are based on their responses to changing conditions – market conditions and customers that want more for less, changes in business processes and constraints. Requirements are dynamic; your software design process assumes that things are static."

"So you're saying that functional decomposition is the real culprit?" he asked.

"I'm afraid so. It's not the functional decomposition itself that's bad, but it doesn't match up very well with the hyperdynamic flux that modern business – that you – have to deal with. It's always behind the times – fashioning a solution for a problem that was identified months or years ago."

"That's interesting. I know all my people pretty well and I know you can sometimes be blind, but I've thought all along that I've got pretty good people."

"From what I've seen," I told him, "you've got excellent people. They really want to succeed. But the system won't let them. Or to be more accurate, it will let them succeed initially, but it falls apart when any revisions are needed."

"We just haven't seen this because up until now, we've been mainly developing brand new systems, not maintaining existing ones."

"Right," I agreed.

"But this is going to be an ongoing problem, then," he said. "We've got several large-scale revisions planned and if we don't do better on them, we're all in real trouble."

"I agree."

"So what's the solution? Java? Or do I need to drink the Microsoft Kool-Aid and go with .NET?"

A Solution in Sight?

"Either of those may make sense for you and your environment," I said, "but neither one will solve this problem. The problem isn't which language you use; it's how you architect software."

"But there is a solution?"

"Yes," I said, "but it isn't a silver bullet. It involves a different way of looking at how software is created – one that at once gives users exactly what they need while, in a sense, ignoring what users tell us."

"Ignore our users?"

"Well, that's too strong," I agreed, "but the idea is that users can't ultimately design software. I know some people build methodologies on the idea that users drive the process and developers just implement what they tell us."

"Yeah, I hear that from some of my colleagues. It makes no sense to me. It's like having customers design cars or houses. No thanks."

"Of course, we need user input and feedback but developers have to be more than coding robots."

"Well, what of the idea of developers and users meeting in the middle? Developers move toward domain expertise while users move toward software architecture expertise?"

"That," I said firmly, "seems to me the worst of all possible worlds. Everyone is now, by definition, a novice."

"Well, I admit I'm intrigued. But I want my developers – at least my leads – and my project managers in on a more in-depth discussion. Can you put together something for next week? Do a presentation outlining the problems you see and how you think we can fix things? In the meantime, you don't happen to have any 'magic consultant powder' to fix this project, do you?"

I apologized that I had unfortunately left that particular supply back in Atlanta.

"Ah, we'll struggle through somehow with this particular mess," he said. "I just want to make sure we have a way to stop the pain for future projects. All right, then. Next time, you'll be speaking to all my key people."

I thanked the CIO for his time and left. I felt certain that this firm, with its dedicated developers and such an attuned CIO, had an excellent chance to turn the situation around so that revisions to existing software would be as successful as new projects had been. Now the responsibility was on me to put together a presentation that would help change their minds – even about things they simply assumed to be true and not a matter of opinion.

On the way out, I asked Debbie, the receptionist, for a local yellow page directory. Finding what I wanted, I wrote down the address and left. Forty minutes later, I was at my destination: Frank's House of Models. I got out of the car and when I opened the door, a small bell rang. I could see someone – Frank, I supposed – looking out from behind stacks of models in their shrink-wrapped boxes. "Can I help you?" he asked.

"I hope so," I said. "What would you suggest for a guy who's trying to build a model world?"

Frank smiled at me. "Follow me," he said. "I might have just what you need." (***Note***: to be continued next month)

About the Author

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

hal@fusebox.org

Which Is Faster?

Find out with an invaluable performance comparison tool

Which is faster, <CFQUERY> or <CFLOOP>? Which is faster, CFML or <CFSCRIPT>? Which is faster… ? If you're a ColdFusion developer, chances are that you've asked (or have been asked) these questions and others like them.

By Ben Forta

So, Which Is Faster?

Okay, so which option is actually the faster one? Honestly? I'm not so sure. Once upon a time (pre-CF4) we were told that <CFLOOP> was very slow. Then (when CF4 shipped) we were told that <CFLOOP> had been optimized and outperformed <CFOUTPUT>. And now, well, CFMX changes everything. So, which is faster?

Of course, performance-related questions are legitimate. We all keep looking for ways to tweak just a little more oomph from our code. But the truth is that this is not just a CFML syntax concern; this affects all sorts of code, database calls, design considerations, reuse implementations, and more.

ColdFusion debugging helps us out a little here. Execution time (for a complete page or parts thereof) provides an invaluable peek under the CF covers. But for more granular performance comparisons, for detailed benchmarking, and for ascertaining definitive performance data, debug output falls a bit short.

Enter ColdFusion MX

Ever since ColdFusion MX shipped I've been trying to better understand exactly what the product is doing internally so as to be able to write better code. And like many of you, I've discovered that an interesting side effect of ColdFusion's new processing engine is that many of the tips and tricks I lived by don't really apply anymore; some have negligible effects on my code, and others are just plain wrong. So I created an application, a really simple performance comparison tool, using good old ColdFusion. This tool has proven so invaluable that I feel compelled to share it with you. Read on.

Measuring Execution Time

The key to measuring any execution time in ColdFusion is an intriguing little function named GetTickCount(). GetTickCount() takes no parameters, and returns a number that is, in and of itself, rather useless. So what makes GetTickCount() so intriguing? The number that GetTickCount() returns is a value in milliseconds, and invoking GetTickCount() exactly one second after a prior GetTickCount() invocation returns a value exactly 1000 times greater than the first returned value.

In other words, to determine exactly how long a line of code takes to execute, you could do the following:

```
<CFSET start=GetTickCount()>
... some CFML code ...
<CFSET end=GetTickCount()>
<CFSET executiontime=end-start>
```

Measuring a single execution yields no usable data – a single invocation may run faster or slower at any time based on lots of other factors (ColdFusion factors and otherwise). To really measure execution time, code must be executed repeatedly, and each iteration must be measured individually and stored. Then an average execution time can be determined by processing all of the saved values.

The simplest way to do this in ColdFusion is to loop through the iterations, saving each measurement to an array, perhaps like this:

```
<CFSET start=GetTickCount()>
... some CFML code ...
<CFSET end=GetTickCount()>
<CFSET ArrayAppend(timings, end-start)>
```

Then to obtain the average execution time you could simply use:

```
#ArrayAvg(timings)#
```

Similarly, total execution time, slowest iteration, and fastest iteration could easily be determined using array functions like this:

```
Total: #ArraySum(timings)#
Slowest: #ArrayMax(timings)#
Fastest: #ArrayMin(timings)#
```

To compare the processing of two options you'd simply repeat this process twice (saving the data to two arrays, one for each option), and then compare the results.

Presenting Results

Detailed benchmarking data is useless unless it can be presented in a format that makes it clear and understandable. <CFCHART> (and related tags) makes it easy to plot the data in a highly usable and readable format.

<CFCHART> is usually used to chart database query results, but as this code snippet demonstrates, individual data points can be passed explicitly using the

<CFCHARTDATA> tag. In this example, two arrays (named timings1 and timings2) are being charted in individual data series in a single chart. The inner loops loop from 1–100, and on each iteration, passes the appropriate array value (from both arrays) to <CFCHART>.

```
<!--- Graph it --->
<CFCHART FORMAT="flash"
         SHOWMARKERS="no">
   <CFCHARTSERIES TYPE="line"
                  SERIESLABEL="Option 1 Title"
                  SERIESCOLOR="red">
      <CFLOOP FROM="1" TO="100" INDEX="i">
         <CFCHARTDATA ITEM="#i#"
                      VALUE="#timings1[i]#">
      </CFLOOP>
   </CFCHARTSERIES>
   <CFCHARTSERIES TYPE="line"
                  SERIESLABEL="Option 2 Title"
                  SERIESCOLOR="green">
      <CFLOOP FROM="1" TO="100" INDEX="i">
         <CFCHARTDATA ITEM="#i#"
                      VALUE="#timings2[i]#">
      </CFLOOP>
   </CFCHARTSERIES>
</CFCHART>
```

Putting It All Together

Over the past few months I've found myself using code much like what I described above over and over. So to make testing quicker and cleaner, I created the "Which is Faster?" application. The app is actually incredibly simple; the idea is as follows:

- The two coding options (perhaps <CFLOOP QUERY=""> and <CFOUTPUT QUERY="">) are each placed in their own .cfm files. (I named mine loop.cfm and output.cfm).
- Some tests may require the execution of code that is not part of the actual test (for example, to test <CFLOOP> and <CFOUTPUT> you'd need a query to be executed, but you'd not want it to be executed along with the <CFLOOP> and <CFOUTPUT> tags), so that code is placed in its own .cfm file. (I named mine query.cfm.)
- A form prompts for the names of these files, descriptions, as well as the number of iterations to use when testing.
- The form submits to a processing page that first includes the preprocessor page (if one was specified) and then loops as instructed, including each test file and saving the execution times to arrays, which are then used by <CFCHART> tags to chart the performance.

For my own testing (to test <CFLOOP> versus <CFOUTPUT>) I used three files. Here they are:

QUERY.CFM

```
<CFQUERY DATASOURCE="exampleapps"
         NAME="emps">
SELECT *
FROM tblEmployees
ORDER BY LastName, FirstName
</CFQUERY>
```

OUTPUT.CFM

```
<CFOUTPUT QUERY="emps">
</CFOUTPUT>
```

LOOP.CFM

```
<CFLOOP QUERY="emps">
</CFLOOP>
```

These are simple, crude, and not real-world at all. But they are perfect for this type of testing – you wouldn't want lots of other processing included, as that could skew the results.

FORM.CFM (Listing 1) contains the code for the form, a simple HTML form as described above (see Figure 1). One checkbox worth noting is the "Exclude first iteration" option; as ColdFusion may have to compile the code on the first execution, you may not want to include that in the collected data (as it will skew the results, creating a much slower first value).

PROCESS.CFM (Listing 2) processes the form. The code first uses <CFPARAM> and a series of <CFIF> statements to validate and prepare passed values. If a preprocessor file was specified, it is then executed. Next, if the "Exclude first iteration" option was checked, each included file is executed once (using <CFINCLUDE> tags) so as to force a compile if needed. Then comes the test itself; <CFLOOP> loops from 1 to the specified number of iterations and, within each iteration, both files are executed and the timings for each appended to one of two arrays (one for each test). Finally the results are displayed in five graphs (see Figure 2). The first shows the execution details, and the

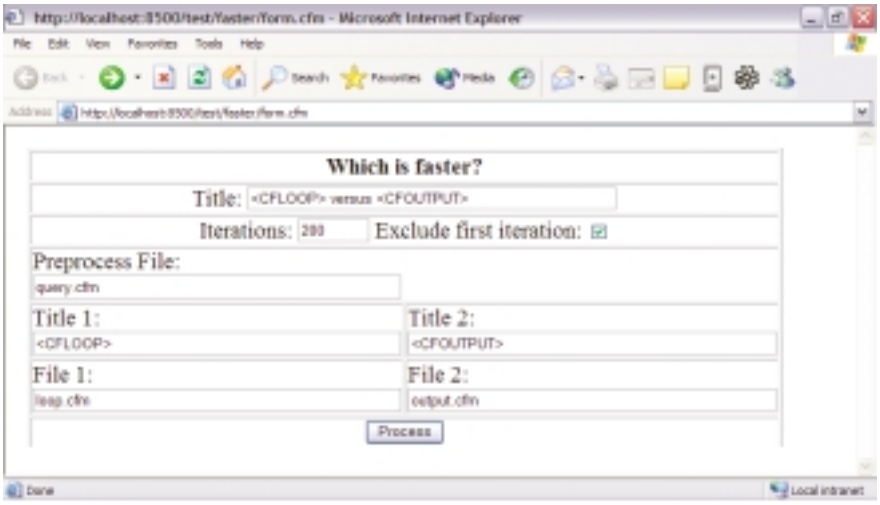


Figure 1: The test is specified in the form

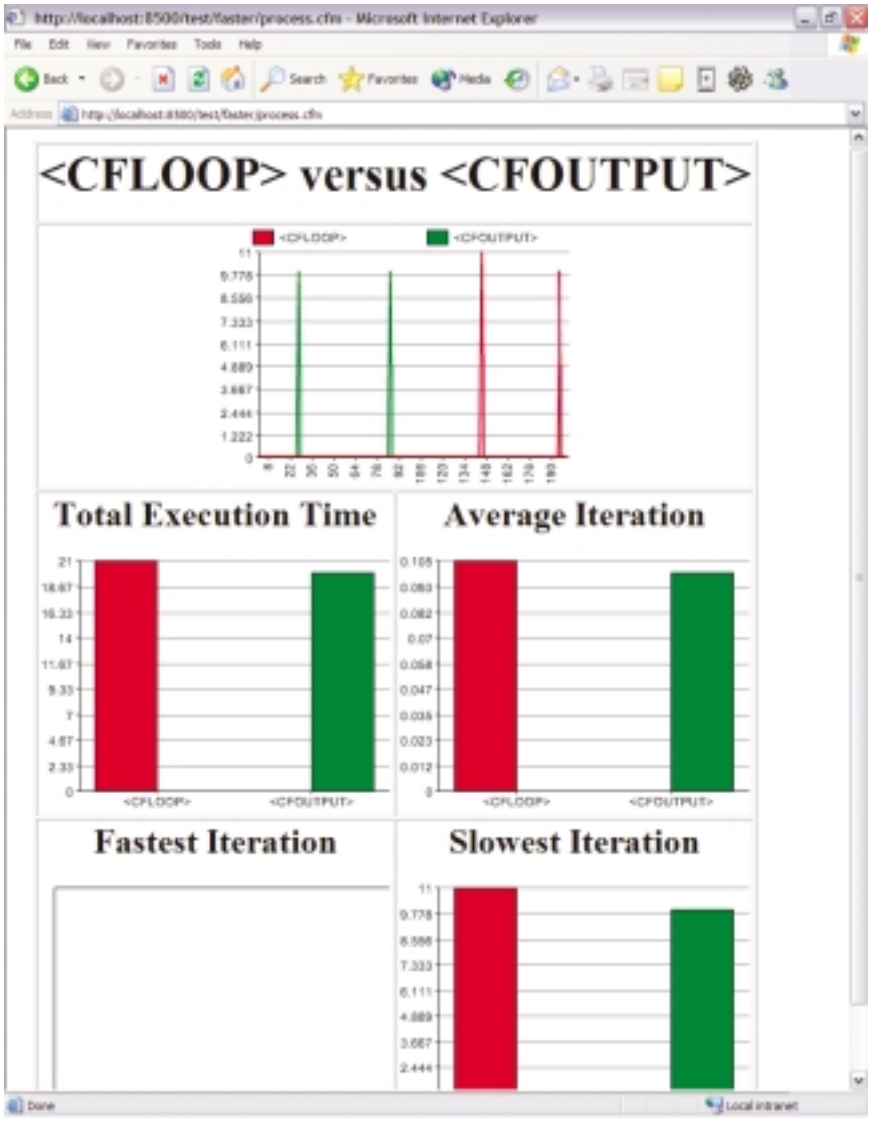


Figure 2: Test results are charted using a series of <CFCHART> tag sets

next four show total processing time, average iteration processing time, as well as fastest and slowest iteration processing time. (In case you were wondering why I used two arrays instead of a single two-dimensional array, these last four graphs are the reason; functions like ArraySum() and ArrayAvg() work on single dimensional arrays only).

Where to Go from Here

Of course, writing all this down has gotten me thinking. It would be nice if instead of just including files with <CFINCLUDE> the application would give you a choice and also allow:

- <CFHTTP> calls to complete pages, local or remote
- <CFINVOKE> calls to Web services or ColdFusion Components
- <CFMODULE> calls to Custom Tags

It's tempting to modify the app right now, but so as to get this column in, I'll leave this enhancement for you to work on. If you find the application useful and enhance it, please let me know.

Conclusion

So, which is faster, <CFLOOP> or <CFOUTPUT>? Answer: <CFLOOP> seems slightly faster, but there is so little difference it's essentially insignificant. Which is faster, CFML or <CFSCRIPT>? Answer: almost no difference (unless I ran ridiculously large tests, as in 10,000 assignments per iteration). Which is faster? Now you can find out yourself – and not just with CFML language elements, but whole blocks of functionality, pages, SQL statements, and more. Enjoy!

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

Listing 1: FORM.CFM

```
<!---
FORM.CFM
"Which is faster" app front end.
Ben Forta - ben@forta.com
--->

<CFFORM ACTION="process.cfm">

<TABLE BORDER="1" ALIGN="center">
<TR>
   <TH COLSPAN="2" ALIGN="center">
      Which is faster?
   </TH>
</TR>
<TR>
   <TD COLSPAN="2" ALIGN="center">
      Title:
      <CFINPUT TYPE="text"
               NAME="title"
               SIZE="50">
   </TD>
</TR>
<TR>
   <TD COLSPAN="2" ALIGN="center">
      Iterations:
      <CFINPUT TYPE="text"
               NAME="iterations"
               VALUE="100"
               SIZE="5"
               VALIDATE="integer"
               REQUIRED="yes"
               MESSAGE="Numeric iterations value is required!">
      Exclude first iteration:
      <INPUT TYPE="checkbox"
             NAME="excludefirst"
             VALUE="1">
   </TD>
</TR>
<TR>
   <TD COLSPAN="2">
      Preprocess File:<BR>
      <INPUT TYPE="text"
             NAME="fileprep"
             SIZE="50">
   </TD>
</TR>
<TR>
   <TD>
      Title 1:<BR>
      <CFINPUT TYPE="text"
               NAME="title1"
               SIZE="50"
               REQUIRED="yes"
               MESSAGE="Title is required">
   </TD>
   <TD>
      Title 2:<BR>
      <CFINPUT TYPE="text"
               NAME="title2"
               SIZE="50"
               REQUIRED="yes"
               MESSAGE="Title is required">
   </TD>
</TR>
```

```
<TR>
   <TD>
      File 1:<BR>
      <CFINPUT TYPE="text"
               NAME="file1"
               SIZE="50"
               REQUIRED="yes"
               MESSAGE="File to process is required">
   </TD>
   <TD>
      File 2:<BR>
      <CFINPUT TYPE="text"
               NAME="file2"
               SIZE="50"
               REQUIRED="yes"
               MESSAGE="File to process is required">
   </TD>
</TR>
<TR>
   <TH COLSPAN="4">
      <INPUT TYPE="submit"
             VALUE="Process">
   </TH>
</TR>
</TABLE>

</CFFORM>
```

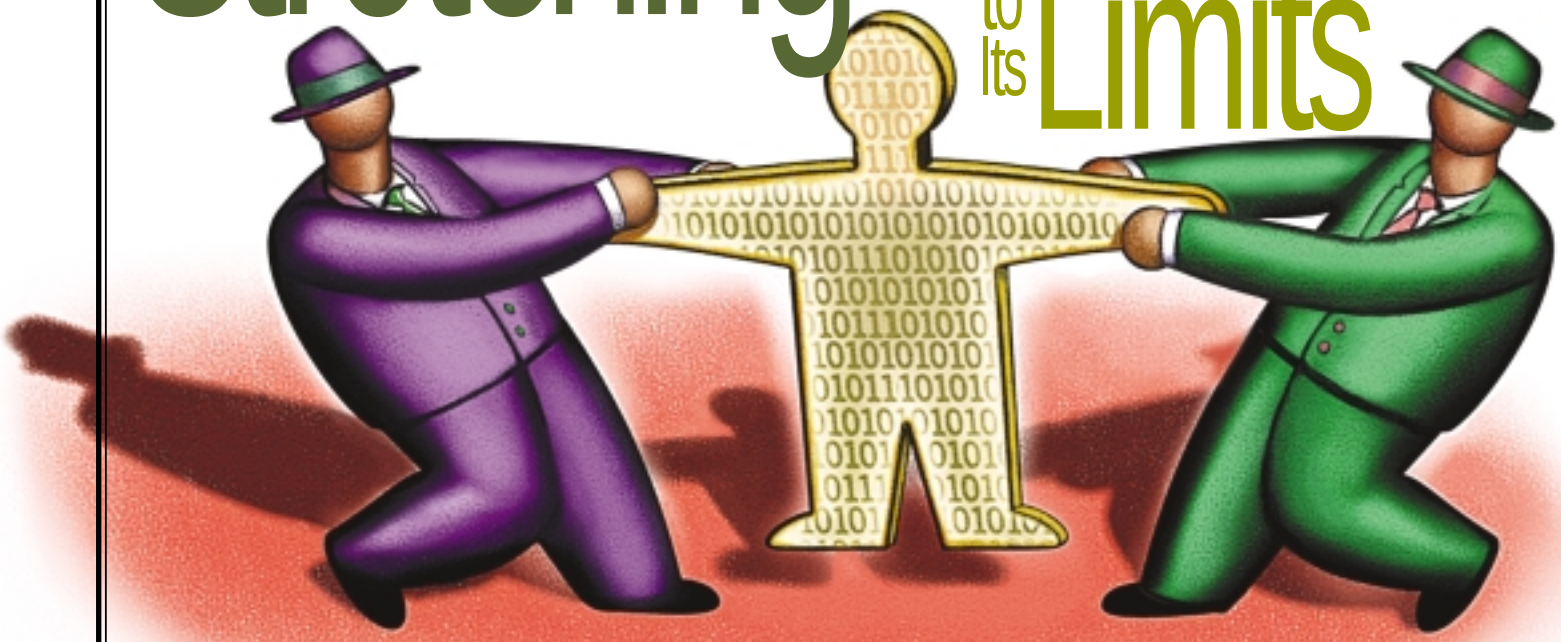
Listing 2: PROCESS.CFM

```
<!---
PROCESS.CFM
"Which is faster" processing.
Ben Forta - ben@forta.com
--->

<!--- Defaults --->
<CFPARAM NAME="FORM.title" DEFAULT="">
<CFPARAM NAME="FORM.iterations" DEFAULT=0 TYPE="numeric">
<CFPARAM NAME="FORM.excludefirst" DEFAULT="0" TYPE="boolean">
<CFPARAM NAME="FORM.fileprep" DEFAULT="">
<CFPARAM NAME="FORM.title1" DEFAULT="">
<CFPARAM NAME="FORM.title2" DEFAULT="">
<CFPARAM NAME="FORM.file1" DEFAULT="">
<CFPARAM NAME="FORM.file2" DEFAULT="">

<!--- Init vars --->
<CFSET chartformat="flash">
<CFSET chartcolor1="red">
<CFSET chartcolor2="green">
<CFSET timings1=ArrayNew(1)>
<CFSET timings2=ArrayNew(1)>

<!--- Check for required --->
<CFIF (Trim(FORM.title1) IS "")
      OR (Trim(FORM.title2) IS "")>
   <CFTHROW MESSAGE="Title are required!">
</CFIF>
<CFIF (Trim(FORM.file1) IS "")
      OR (Trim(FORM.file2) IS "")>
   <CFTHROW MESSAGE="File to execute are required!">
</CFIF>

<!--- Run "silent" --->
<CFSILENT>

<!--- Run preprocess file if specified --->
<CFIF NOT Trim(FORM.fileprep) IS "">
   <CFINCLUDE TEMPLATE="#Trim(FORM.fileprep)#">
</CFIF>

<!--- Process first run if needed --->
<CFIF FORM.excludefirst>
   <!--- Exclude first run, so run each once --->
   <CFINCLUDE TEMPLATE="#Trim(FORM.file1)#">
   <CFINCLUDE TEMPLATE="#Trim(FORM.file2)#">
</CFIF>

<!--- Now loop as needed --->
<CFLOOP FROM="1" TO="#FORM.iterations#" INDEX="i">
   <!--- Get start value ---->
   <CFSET tcstart=GetTickCount()>
   <!--- Run the first file --->
   <CFINCLUDE TEMPLATE="#Trim(FORM.file1)#">
   <!--- Append the timing --->
   <CFSET ArrayAppend(timings1, GetTickCount()-tcstart)>
   <!--- Get start value ---->
   <CFSET tcstart=GetTickCount()>
   <!--- Run the second file --->
   <CFINCLUDE TEMPLATE="#Trim(FORM.file2)#">
   <!--- Get end value --->
   <CFSET tcend=GetTickCount()>
   <!--- Append the timing --->
   <CFSET ArrayAppend(timings2, GetTickCount()-tcstart)>
</CFLOOP>

<!--- No more silent --->
</CFSILENT>

<TABLE ALIGN="center" BORDER="1">
<TR>
<TD COLSPAN="2" ALIGN="center">
   <H1><CFOUTPUT>#HTMLEditFormat(Trim(title))#</CFOUTPUT></H1>
</TD>
</TR>
<TR>
<TD COLSPAN="2" ALIGN="center">
   <!--- Graph it --->
   <CFCHART FORMAT="#chartformat#"
            SHOWMARKERS="no">
      <CFCHARTSERIES TYPE="line"
                     SERIESLABEL="#HTMLEditFormat(Trim(title1))#"
                     SERIESCOLOR="#chartcolor1#">
         <CFLOOP FROM="1"
                 TO="#FORM.iterations#"
                 INDEX="i">
            <CFCHARTDATA ITEM="#i#"
                         VALUE="#timings1[i]#">
         </CFLOOP>
      </CFCHARTSERIES>
      <CFCHARTSERIES TYPE="line"
                     SERIESLABEL="#HTMLEditFormat(Trim(title2))#"
                     SERIESCOLOR="#chartcolor2#">
         <CFLOOP FROM="1" TO="#FORM.iterations#" INDEX="i">
            <CFCHARTDATA ITEM="#i#"
                         VALUE="#timings2[i]#">
         </CFLOOP>
      </CFCHARTSERIES>
   </CFCHART>
</TD>
</TR>
<TR>
<TD ALIGN="center">
   <H2>Total Execution Time</H2>
   <CFCHART FORMAT="#chartformat#"
```

```
            SHOWLEGEND="no">
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor1#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title1))#"
                      VALUE="#ArraySum(timings1)#">
      </CFCHARTSERIES>
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor2#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title2))#"
                      VALUE="#ArraySum(timings2)#">
      </CFCHARTSERIES>
   </CFCHART>
</TD>
<TD ALIGN="center">
   <H2>Average Iteration</H2>
   <CFCHART FORMAT="#chartformat#"
            SHOWLEGEND="no">
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor1#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title1))#"
                      VALUE="#ArrayAvg(timings1)#">
      </CFCHARTSERIES>
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor2#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title2))#"
                      VALUE="#ArrayAvg(timings2)#">
      </CFCHARTSERIES>
   </CFCHART>
</TD>
</TR>
<TR>
<TD ALIGN="center">
   <H2>Fastest Iteration</H2>
   <CFCHART FORMAT="#chartformat#"
            SHOWLEGEND="no">
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor1#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title1))#"
                      VALUE="#ArrayMin(timings1)#">
      </CFCHARTSERIES>
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor2#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title2))#"
                      VALUE="#ArrayMin(timings2)#">
      </CFCHARTSERIES>
   </CFCHART>
</TD>
<TD ALIGN="center">
   <H2>Slowest Iteration</H2>
   <CFCHART FORMAT="#chartformat#"
            SHOWLEGEND="no">
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor1#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title1))#"
                      VALUE="#ArrayMax(timings1)#">
      </CFCHARTSERIES>
      <CFCHARTSERIES TYPE="bar"
                     SERIESCOLOR="#chartcolor2#">
         <CFCHARTDATA ITEM="#HTMLEditFormat(Trim(title2))#"
                      VALUE="#ArrayMax(timings2)#">
      </CFCHARTSERIES>
   </CFCHART>
</TD>
</TR>
</TABLE>
```

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

Stretching Web Services to Its Limits



Dynamic invocation of simple services

XML has already solidified its position in the Web application space as the solution for data extensibility, and we are becoming increasingly aware of Web services and its potential to revolutionize the distributed application architecture. What we have yet to discover is exactly how this is all going to happen.

We know that the Web services architecture allows for easy integration between applications built independently from each other. We also know that sometime in the near future when we want to extend our applications, we will have a system in place that allows us to methodically search and consume a service from a wide selection of services.

With the overwhelming success of WSDL and the proposed UDDI specification, we have begun to work toward the goal of dynamic discovery and invocation of Web services. What we need to do now is to begin to think about what it actually will take to make these services truly dynamic. Regardless of what method we choose for discovery, our applications will need to go beyond their current scopes in order to truly make Web services dynamic.

By Ron West

Most ColdFusion developers know what WSDL is, but may not have spent a great deal of time or resources trying to figure out how it works – and why would you? CFMX makes things so easy; in most cases it would be a waste of time to learn WSDL in depth. Because WSDL is built on XML, and CFMX now has extensive support for XML, we can use things like XPath to help us discover the properties of a given Web service.

This article will give a brief introduction to the wheres and hows of WSDL, and we'll use XPath to search the WSDL file to discover the properties for a given Web service. For demonstration purposes we will apply our logic to a "Simple" Web service, which works with and returns simple data types like strings. We will not be working with complex data types such as arrays and custom data types. Although complex data types are required in order to handle more intricate Web services, we must walk before we can run.

XML and CFMX

Although true support for XML is new to CFMX, XML itself has been around for quite some time. In fact, there are many techniques available for working with XML, one of those being XPath. XPath is a W3C recommendation that allows for broad XML searching. To use XPath with CFMX you use a function called XMLSearch. XMLSearch() takes two arguments: first the XML document and second the XPath expression.

In CFMX there are a few options for creating an XML doc. In our case we will use cfhttp to get the WSDL file. XMLSearch returns an array of nodes that match your criteria. The array that is returned from the XMLSearch() function is basically an array of structures that describes all of the properties of the XML node. The following structure keys are available from a successful XMLSearch():

- ***XMLName:*** The name of the root element found
- ***XMLNsPrefix:*** The namespace prefix for this element
- ***XMLText:*** Any text contained within the open end tag for this element
- ***XMLComment:*** Any XML comments made for this element
- ***XMLAttributes:*** A structure of name-value pairs for each (if any) of the attributes for this element
- ***XMLChildren:*** A structure containing all of these keys for any children elements

Using these structure keys we will have all of the information needed to discover the properties for this service. To gain more insight on how XPath is used, go to www.w3c.org/xpath. I realize that this is a brief description of XMLSearch and XPath, but once we are done here, you should have more than enough code to test this on your own.

To perform a simple search, such as a node existence test, you could perform the following in a script block:

```
MyVar = XMLSearch(myXMLDoc, "//binding");
```

If your XML document contains a node (element) named "binding" anywhere in the document tree, MyVar would be an array with length of at least one. Likewise if we wanted to find all nodes named "operation" that have an attribute named "attr" with value "hello" we could write the following code:

```
MyVar = XMLSearch(myXMLDoc, "//operation[@attr='hello'"");
```

Basically the expressions can be equated to where clauses in a standard SQL statement like: Select * from XML where node = "operation" and attributeName = "attr" and attributeValue = "hello".

WSDL Architecture (the Wheres)

The first task for working with Web services is to discover available services. With UDDI still in the "discovery" stages, we can look to our friends over at XMethods.net. The good folks over there have provided us with numerous Web services. To reduce headaches, we will concentrate only on RPC-style Web services. Let's take a look at the "BorlandBabel" Web service. The service's WSDL file can be found at: http://ww6.borland.com/webservices/BorlandBabel/BorlandBabel.exe/wsdl/IBorlandBabel.

The <description> element is the root element, where you see the namespaces for the collective properties of the WSDL file. We will not cover namespaces in this article (even though they are important, we can proceed without having to go into specifics). Next, the <message> element that defines the structure of the messages sent between the client and the host: we have four <message> elements with the names: "BabelFishRequest", "BabelFishResponse", "SupportedLanguagesRequest", and "SupportedLanguagesResponse".

Notice three of the <message> elements contain <part> elements. The <part> element defines the encoding type for that particular message (it is either a response message or call message). A message without a <part> child element ("SupportedLanguagesRequest") takes no parameters. After that, we have the <portType> element, which defines the parameters for all operations available. The <portType> attribute "name" is used as internal reference, which we will tie together a little later.

Now comes the <binding> element, which is used to provide the operation details. In our example we see two <operation> elements inside the <binding> element, which define the input and output parameters for the services named "BabelFish" and "SupportedLanguages". Finally we have our <service> element, which contains the names of the services available and the location of the service URI.

Now that we have defined the basics of the WSDL file, we can build an organized description of all of the properties of the Web service by mixing in some XMLSearch(). XMLSearch() can be used to search through an XML document using XPath syntax and expressions. Let's get started.

Dissect the Service Properties (the Hows)

There are five basic steps or pieces of information required to consume a Web service:

1. Check style of Web service (make sure that we are working with an RPC-style service)
2. Discover the name of the service
3. Discover what ports (methods) are available for this service
4. Discover what input, output, and/or fault messages are needed for each port (method)
5. Determine the data type for each message

(All references below can be found in Listing 1.)

1. Notice the search string: "//*[contains(name(), 'binding')][@style='rpc']". This means find all nodes that contain the name "binding" and have an attribute named "style" whose value is "rpc". If the length of the array is greater than 0, we have a valid service.

2. In order to enhance our documentation we will now find the name of the service. One feature available in WSDL that can be useful here is the < documentation > element. It is used to return a human-readable description for any element. In this case it would have been nice to have a < documentation> element that would describe the service for us. We could then store that with the service so that users knew what the service was designed for. To find the service name, simply use "//*[contains(name(), 'service')]". We gain access to the service name with XMLAttributes.name. We can use this service name to classify our Web service within our application.

3. Now for the methods. We use "//*[contains(name(), 'binding')]/*[contains(name(), 'operation')][@name]" to return our methods array. It is possible for a service to have multiple methods (and many do). For each method available, there will be one node in our array. To get the names of the methods, we loop through the array and get the value for the attribute "name". From here you would begin to store the data into whatever storage facility was planned. You might store each method in a database and assign it a method ID, which could be used later for the properties of the method.

4. Next we want to discover what input, output, and/or fault messages are required for each method. To do this we access the properties of each child node for the methods defined in step 3. We make another call to XMLSearch() with "//*[contains(name(), 'binding')]/*[contains(name(), 'operation')][@name='#methodName#']/*". Notice we have dynamically supplied the method name in the search string ("@name='#methodName#"). In doing so, we will have direct access to the children elements for that method. The names of the children elements will be input, output, or fault. (You can ignore the "soap:operation" node as this does not reference a message type for this method.)

   Services are not required to define both an input and output and are certainly not bound to using a fault message. It is possible to make a call to a service and not be required to supply an input message and simply receive an output message or vice versa. It is important to remember this key feature when designing dynamic systems, as you would need to abstractly handle a number of possibilities.

5. Finally, we want to discover the data types for each of the messages returned from step 4. This is a little tricky, but if you have hung in so far this should not be difficult to follow. The data types are stored separately from the message ports. We first need to discover the internal reference to the data types for each message by looking into the details of the <portType> element. We make the call "//*[contains (name(), 'portType')]/*[contains(name(), 'operation')] [@name='#methodName#']/*[contains(name(), '#param Name#')]" to gain access to the parameters of the portType for the operation.

   In SQL terms, the line would read something like this: "Select * from XML where name like 'input' and parent like 'BabelFish' "whose" parent like 'portType'. One of the attributes of the returned element will be "Namespace". The "Namespace" attribute defines the internal reference to the message definition, which will have the data types. (It is not necessary to fully understand the use of namespace for the simple service, but in order to work with complex data types this will become an important concept, as it will determine where the data-type definitions are located.) Use the namespace to get at the details for each <part> element of this message. Each <part> element will have two attributes that describe its functionality – the attribute "name" and the attribute "type". The type attribute here represents the encoding type of the message part. It is possible to have multiple parts for any given input message. Each part may have a different data type. There is only one part to an output message. ColdFusion logically maps the encoded data types to ColdFusion data types. The matrix for this mapping is located in the ColdFusion documentation.

What Should We See?

When things run smoothly – and I hope they did for you – you should get the following results for the code provided in Listing 1:

1. A service named "IBorlandBabelservice"
2. Two port (methods) with the following properties
   - a. BabelFish
     - i. Input message with parts:
       1. TranslationMode (of type string)
       2. Sourcedata (of type string)
     - ii. Output message (of type string)
   - b. SupportedLanguages
     - i. Input message with 0 (zero) parts
     - ii. Output message (of type string)

You now have all of the information to dynamically call this service and systematically incorporate it into your application. The metadata for this service can be stored in a database and invoked anywhere in your application. An interesting aspect of this service, and the reason I chose it over the original BabelFish service, is that it uses data returned from one method to instantiate another. The data returned from the "SupportedLanguages" method is a list (whitespace delimited) that contains all languages supported in the call to the "BabelFish" function. To use this service you simply call the SupportedLanguages, and the return variable could be used to populate a selection list for the translation mode in the input message of the BabelFish method. The particular technique here is advantageous over the original BabelFish service because this service provides an automated updating method. If any new languages are supported, they are simply returned in the SupportedLanguages method call. Although we have not outlined a solution to this procedure here, it deserves mention because in the future this will become a common practice.

Incidentally, if you are interested in attempting to use this code with other Simple Web services and are not sure if your Web service fits into this category, you can add the following code to your app: //*[contains(name(), 'complexType')]. If the XMLSearch with this string returns an array with length greater than 0, you are not working with a Simple Web service.

Go Forth from Here

Hopefully this article sets the wheels in motion. Developers and software vendors are adopting Web services at a rapid rate. Once UDDI is solidified and there is a common interface for discovering Web services, we will be left only with dynamic invocation. Now that CFMX natively supports both Web services and XML, we are one step closer to our goal. "Computer, find the best rate for my snowboard weekend. Oh yeah, and make sure the resort has lots of snow." We truly live in a remarkable time.

About the Author

Ron West is a senior applications developer with PaperThin, Inc., a privately held Web content management vendor headquartered in Quincy, Massachusetts. Ron has been working with Web applications for seven years. He is one of the directors of the Rhode Island ColdFusion User Group, and is an established writer for several industry publications.

rwest@paperthin.com

Listing 1:

```
<cfscript>
   //serviceURL = attributes.serviceURL;
   serviceURL =
"http://ww6.borland.com/webservices/BorlandBabel/BorlandBabel.exe/wsdl/IBor
landBabel";
   serviceData = structNew();
</cfscript>
<!--- // get WSDL File --->
<cfhttp url="#serviceURL#" method="GET" resolveurl="false">
<!--- // create CFML XML document --->
<cfset wsdl = xmlParse("#cfhttp.fileContent#")>
<cfscript>
   // test for RPC validity
   str = "//*[contains(name(), 'binding')][@style='rpc']";
   rpcArray = XMLSearch(wsdl, str);
   if ( arrayLen(rpcArray) )
   {
      // discover service name
      str = "//*[contains(name(), 'service')]";
      serviceArray = XMLSearch(wsdl, str);
      if( arrayLen(serviceArray) )
         serviceName = serviceArray[1].XMLAttributes.name;

      // discover the methods
      str = "//*[contains(name(), 'binding')]/*[contains(name(), 'opera-
        tion')][@name]";
      methodArray = XMLSearch(wsdl, str);
      // for each method returned get the properties for this method
      for( i=1; i lte arrayLen(methodArray); i=i+1 )
      {
         mStruct = methodArray[i];
         // method name
         methodName = mStruct.XMLAttributes.name;
         // write some code here to store this off into DB or wherever
           [methodId = storeMethod(methodName);]
         // discover the children nodes which will be our input, output
            and fault codes
         str = "//*[contains(name(), 'binding')]/*[contains(name(),
           'operation')][@name='#methodName#']/*";
         mDetailsArray = XMLSearch(wsdl, str);
         for( j=1; j lte arrayLen(mDetailsArray); j=j+1 )
         {
            param = mDetailsArray[j];
            paramName = param.XMLName;
            // store these with the service [messageId =
               storeMessage(methodId, paramName);]
            // discover the internal reference for each message
            str = "//*[contains(name(), 'portType')]/*[contains(name(),
              'operation')][@name='#methodName#']/*[contains(name(),
              '#paramName#')]";
            portData = XMLSearch(wsdl, str);
            // if we have a port definition
            if( arrayLen(portData) gt 0)
            {
               ref = listLast(portData[1].XMLAttributes.message, ":");
               // discover the datatypes with the internal reference to
                  the message element
               str = "//*[contains(name(), 'message')][@name='#ref#']/*";
               partArray = XMLSearch(wsdl, str);
               dump(partArray);
               // for each message discover it's parts (if any)
               for( c=1; c lte arrayLen(partArray); c=c+1 )
               {
                  part = partArray[c];
                  partName = part.XMLAttributes.Name;
                  partType = listLast(part.XMLAttributes.Type, ":");
                  // store each of these with the message data for this
                     method
               }
            }else // we did not have a port definition
               msg = "No port defined";
         }
      }
   }else // we do not have a valid RPC Style service
      msg = "Not a valid RPC Style Web service";
</cfscript>
```

CF Community *—continued from page 7*

UPDATE statements don't take more than 5–10 minutes to code (for a developer who is comfortable with basic SQL) – a <CFINSERT> or <CFUPDATE> tag takes a little less than half that time at best.

On the downside, <CFINSERT> and <CFUPDATE>:

- Can be more difficult to debug
- Encourage developers not to learn SQL
- Are not self-documenting (SQL Queries are self-documenting)
- Carry more performance overhead because ColdFusion has to do more work
- Limit developers by enforcing variable naming conventions and functionality

While the use of these tags is acceptable for beginning developers who do not yet know the basics of SQL and have a tight deadline to meet, their use isn't justifiable for any developer who could find three or four hours to sit down and learn the basics of SQL INSERT and UPDATE syntax.

<CFFORM> is a different story. <CFFORM> can be used (along with nested <CFINPUT> tags) to generate client-side JavaScript validation for form fields. The JavaScript it generates is compatible with the majority of Web browsers on the Web, and it works (you almost never have to debug it). Though it does shield developers from having to learn and/or write their own JavaScript, the <CFFORM> tag is an excellent solution for Rapid Development.

If a form needs more complex validation or validation on form field types not supported by the <CFINPUT> tag, then developers will have to write their own validation or look elsewhere for help. JavaScript manually written by developers can be a bit more streamlined and functional (as mentioned), but if all a developer's needs are more quickly met by using <CFFORM>, then it's hardly a bad practice.

I strongly suggest that all developers who want to do form validation learn JavaScript as soon as they can rather than continue to rely on <CFFORM> and be bound by its limits. Another advantage to learning JavaScript is that you'll find it much easier to learn to build applications in Flash since its language (ActionScript) is another ECMA Script–based language.

The lesson here is that it is not always (but can be sometimes) a terrible practice to let an application automate code generation for you, but it's never an excuse not to learn another technology. Learning not to be dependent on code generators will expand your possibilities, make you more marketable, and result in more robust and better thought-out applications.

Bridging the Gap Between Flash and ColdFusion MX

Macromedia course aids developers interested in building business applications

Everybody knows that Flash MX and ColdFusion MX are able to communicate with each other via Flash Remoting, right? But where can ColdFusion developers go to learn everything they need to know to begin building Flash applications and Flash front ends to their existing ColdFusion MX applications? Where can a developer learn how this Flash Remoting technology works, and how to begin leveraging it in applications? Macromedia Training serves up an answer in the form of "Developing Rich Internet Applications with Flash MX and ColdFusion MX" (a.k.a. "DRIA").

By Simon Horwith

Since the release of the MX suite of products, Macromedia Training has been busy revamping their classes to reflect new product features. In addition to new versions of existing classes, several new classes have been added to the catalog, not the least of which is DRIA. As a Macromedia Certified Instructor, I've seen many new Macromedia course offerings come and go over the years, but never have I been as excited about the release of a new course as I am about DRIA.

Why am I so excited? What exactly is DRIA? In simplest terms, DRIA is a class that teaches Flash to developers. The course material spans three days, and assumes no prior knowledge of Flash. What is assumed is prior programming experience with another language such as CFML. Due to the massive amount of material the course covers in three days, prospective students should already be familiar with programming constructs such as variables, arrays, associative arrays (structures), and conditional logic constructs like if-else statements.

The DRIA syllabus consists of nine units taught over three days. Students learn many essential skills including:

- Setting up a site in Dreamweaver MX
- Introduction to the Flash MX authoring environment interface
- Creating and publishing Flash documents
- Creating, validating, and manipulating with text fields
- Creating variables, arrays, named arrays, objects, and functions with ActionScript
- Controlling application flow with conditional logic and loops in ActionScript
- Fundamental object-oriented concepts and their application in ActionScript
- Debugging Flash applications – including debugging Flash network connections
- Using Flash UI Components such as combo boxes, list boxes, and push buttons
- Creating ColdFusion Components
- Making Flash applications "talk" with ColdFusion via Flash Remoting
- Using data retrieved from ColdFusion to populate Flash MX UI Components
- Working with events in Flash MX applications
- Using the data grid component to build sortable and dynamic data views
- Using data glue to work with record sets
- Persisting data with Flash MX
- Using drag and drop in Flash MX business applications
- Dynamically loading images in Flash MX
- Programmatically working with movie clips and layers in ActionScript

DRIA teaches these and other skills to students as they follow along with the instructor as well as work independently to build Flash MX interfaces for business applications. Students who have already taken either or both of Macromedia's ColdFusion classes will be happy to know that most of the DRIA exercises are performed in the context of the Coffee Valley application (the application used for walkthroughs and labs in the "Fast Track to ColdFusion" and "Advanced ColdFusion for Developers" courses) where many of its existing ColdFusion pages are replaced with Flash functionality. These Flash-enabled pages include a membership directory page, product catalog, and shopping cart – all dynamically driven by ColdFusion.

"Developing Rich Internet Applications with Flash MX and ColdFusion MX" fills a void that has

existed since the release of the MX suite of products (by teaching developers how to use Flash to achieve the same functionality that many currently achieve using HTML forms, JavaScript, and stylesheets). What's more, students can walk into the class with absolutely no knowledge of Flash and, in three days, leave armed with all the tools and knowledge they need to begin building full-blown applications with Flash MX. This is as exciting for me as an instructor as it is for my students.

Although it's important to know what DRIA is, it's also important to know what it is not. DRIA is not a course for aspiring designers. Students do not need to have any artistic ability in order to take DRIA, nor will the course require them to do much more than draw a simple rectangle on the screen (and even that is a trivial task with the Flash MX tools). Basically, if you know how to hold a mouse and move it without its falling off the mouse pad, then you've already got all the artistic ability you need to successfully complete the DRIA class.

Similarly, DRIA does not teach sound or animation. If you are interested in using Flash to create art from scratch or to create animations, look into Macromedia's "Fast Track to Flash MX" and "Flash MX for Designers" courses. DRIA does not teach students ColdFusion, but it does teach students best practices for how to create ColdFusion Components as well as how to make this ColdFusion code accessible from Flash. Students should already be familiar with the basics of the CFML programming language (the "Fast Track to ColdFusion" and "Advanced ColdFusion for Developers" courses are an excellent way to learn these skills). Basically, this is a course for developers interested in building business applications with Flash MX (and ColdFusion MX).

DRIA is a fast-paced course designed to take ColdFusion developers with no prior knowledge of Flash, and turn them into Flash MX developers in three days. While no one can learn everything there is to know about Flash in a single class, students leave with a strong understanding of all of the fundamental skills required to build robust Flash MX business applications that are capable of communicating with ColdFusion (and any other Web service–enabled) applications.

For more information about DRIA, or to register, visit Macromedia's training site at www.macromedia.com/go/dria or Fig Leaf Software's DRIA training page at http://training.figleaf.com/figleaftraining/Courses/DRIA.cfm.

About the Author

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

shorwith@figleaf.com

inFusion Mail Server from On-Line Data Solutions

The extendable mail server

Once they've learned the basics of ColdFusion, one of the first applications many developers will write is an e-mail program. By using CFMAIL and CFPOP, it is easy to create a basic application that will send mail to an SMTP service and retrieve mail from a POP service.

By Selene Bainum

This type of functionality can be limited, however, and the only communications the developer can perform with the mail server is to send and receive e-mail. If the user information is stored in an LDAP directory, it can be accessed programmatically, but that certainly adds another level of complexity. Not to mention all mail server administration, such as creating users and domains, can only be performed by an administrator – typically not a developer – using a tool provided by the maker of the mail server.

Architecture You Can Understand

inFusion Mail Server from On-Line Data Solutions (ODS) changes what you may know, or may not know, about mail servers. Typically, a mail server stores data, such as user accounts, messages, and address books, in a proprietary data store that cannot be accessed by scripting languages, such as ColdFusion. Even when you use CFPOP to read messages, certain information, such as whether or not the message has been read already, is not accessible.

Instead of inaccessible data stores, inFusion stores all information relating to the configuration and administration of the mail server in a database – MS Access, MS SQL Server, Oracle, Sybase, and MySQL are all supported. Even the messages themselves are stored in the database, making the interaction between ColdFusion and the mail server much more useful.

Unlike most mail servers, all of inFusion's functionality is handled with scripting templates, such as ColdFusion, PHP, inline HTML, or Active PERL. That means, in this case, that ColdFusion is required on the mail server to process messages and perform administrative tasks. Included C++ custom tags are used to communicate with the mail server as opposed to using CFMAIL and CFPOP. A GUI tool is also included to set some of the service configuration settings (see Figure 1).

Full Mail Server Functionality

inFusion is a full-fledged e-mail server that allows many different types of functionality such as:

- Assigning users to security groups, allowing them to perform only certain administrative tasks
- Creating several different domains using the same mail server
- Restricting the size of mailboxes within a particular domain
- Restricting the size of messages received within a particular domain
- Requiring optional LOGIN or CRAM-MD5 SMTP authentication
- Restricting Relay and SMTP services to specified IP addresses
- Returning delivery receipts for received messages
- A scheduler tool that allows templates to be run in intervals as short as every few seconds
- Indicating whether or not a message has already been read

It would be impossible to list the complete functionality of inFusion in this review, but suffice it to say it is highly flexible. There are, however, some functions that are not available with inFusion:

- Requesting delivery or read receipts for sent messages
- Returning read receipts for received messages
- Receiving message flags, such as priority, from received messages

Open Source Templates

When you purchase inFusion, you can download FusionMail, a complete e-mail server written using ColdFusion. FusionMail is freeware and the source code is not encrypted, making modifying the existing templates and creating your own quite easy. Since the existing templates are backwards compatible to ColdFusion version 2, you needn't worry about what version you are running.

If you are using ColdFusion as the scripting engine, FusionMail must be installed in order to get the full functionality of your inFusion services. FusionMail comes with several components, including installation, administration, and an e-mail application. The administration interface allows you to modify all of the service system settings that can also be performed with the GUI tool as well as perform tasks such as creating domains and users.

The e-mail application has a familiar interface (see Figure 2); a left-hand frame contains the available folders for the user and a link to the address book, one middle frame contains the contents of the selected folder, and another middle frame displays the current message. Messages that have not been read are bolded, and the current folder view can be filtered to view either read or unread messages or urgent or nonurgent messages. Existing messages can be selected and either marked urgent/nonurgent, moved to another folder, or deleted. The message display frame will display the details of the message, including all attachments associated with the message.

The address book allows individual users to create nicknames and e-mail addresses for their contacts as well as enter additional personal information such as name, address, and phone number. Users can also add additional folders to aid in managing their messages.

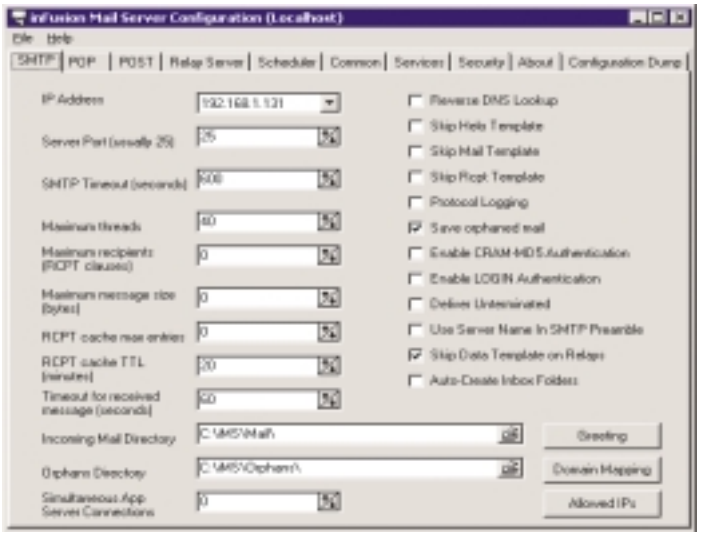


Figure 1: inFusion Mail Server configuration GUI

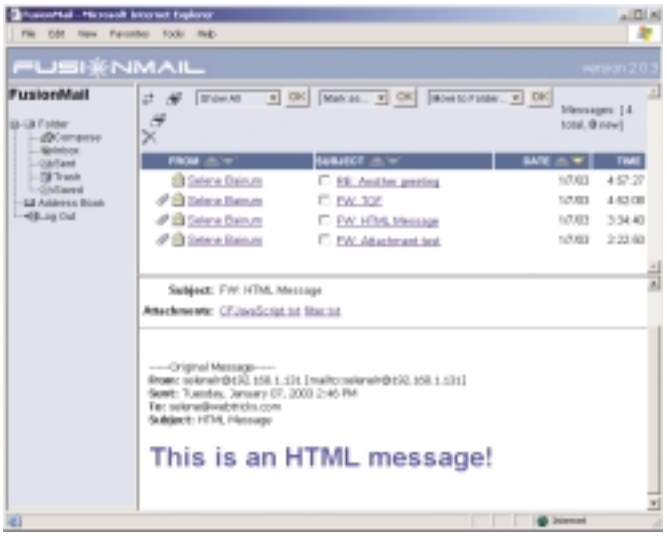


Figure 2: FusionMail's e-mail application

The message compose window contains fields for To, CC, BCC, subject, attachments, and the message body. The address book can be accessed from the compose window, and the selected contacts are copied to the appropriate fields using JavaScript. Multiple attachments can be sent with a single message and it is easy to remove an uploaded attachment before the message is sent. The message body field starts out as a standard textarea but can be changed to a WYSIWYG editor that allows you to create and view the message in HTML format.

The look and feel of FusionMail's applications are not very sophisticated. Many developers will want to modify the templates and their functionality, but the fact that you can do that far outweighs the limitations of the existing templates. The current e-mail application framework is not compatible with Netscape 4.7 and the WYSIWYG editor is compatible only with Internet Explorer.

Expanding the System

The true value of inFusion may not be what comes with it, but what you can add to it. Sometimes it is worth more to be able to create your own functionality than to purchase similar functionality that you cannot change.

For example, if you normally want to add the functionality of a list-serv to your application, you must purchase the server software at an additional cost. However, the cheaper the system, the less sophisticated the administration interface. inFusion's database comes equipped with the tables necessary to build your own list-serv functionality.

But why stop there? You can certainly add your own tables to the database and integrate your own applications such as scheduling. In fact, inFusion can easily be the backbone of an intranet or extranet application.

Flexible Pricing

Since your company's mail server needs may differ from the next company's, ODS offers à la carte pricing options so you don't have to pay for more functionality than you need. Pricing is based on the maximum number of simultaneous connections needed. The Small Office Home Office (SOHO) bundle can be purchased for $500 and supports up to 10 simultaneous connections. Larger companies can purchase the Enterprise edition that supports up to 1,024 simultaneous connections for $4,950. There is a free single-connection developer edition that makes development even easier. If you don't need the full bundle of services (SMTP, POST, POP, Relay), you can purchase just the components you need.

A Good Way to Go

When it comes to flexibility and expandability inFusion Mail Server is a hands-down winner. The ability to actually create mail server functionality using ColdFusion, as opposed to just a pretty way to read and send messages, is a life saver. inFusion comes with ample documentation to help walk you through the mail server itself and the open nature of FusionMail makes it easy to extend the included functionality. Combining the extendability with the flexible pricing, ODS ensures that inFusion is a great solution for almost anyone who is using ColdFusion and looking for a fully integrated mail server system.

About the Author

Selene Bainum has been developing with ColdFusion since 1996. She created and maintains www.webtricks.com, a premier ColdFusion tutorial site.

selene@webtricks.com

Vitals: inFusion Mail Server

On-Line Data Solutions, Inc.
1919 Middle Country Road • Suite 204
Centereach, NY 11720 • Phone: 1-631-737-4668
Fax: 1-631-737-9539 • E-mail: info@coolfusion.com
Web: www.coolfusion.com

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

Pricing: inFusion pricing ranges from $250–$4,950 based on the components purchased and number of simultaneous connections required. Free developer editions are available.

PRODUCT SNAPSHOT

Target audience: Organizations that are looking for a configurable mail server that can be enhanced and administered using ColdFusion templates

Pros:
- Full mail server functionality
- Flexible pricing
- Extendable functionality
- Multi-database support
- Nonproprietary data stores

Cons:
- Can only be administrated using a Web interface
- Lacks a few common mail server functions

Server OS: Windows platforms
ColdFusion: Version 2.0 and above, including MX
Database Support: MS SQL Server, Oracle, Sybase, MySQL, MS Access

CFML Forever!

BlueDragon expands your range of choices

Since New Atlanta first announced BlueDragon about a year ago, we've been asked two common questions: "What is BlueDragon?" and "Why did you create it?" I thought a good starting point and introduction for my first column (which, hopefully, will become a regular ***CFDJ*** feature) would be to answer these questions, particularly the latter.

By Vince Bonfanti

Briefly, BlueDragon is a CFML scripting engine and runtime module that is implemented as a standard Java servlet. BlueDragon is packaged as a standalone server based on New Atlanta's ServletExec servlet/JSP engine, complete with its own built-in Web server and adapters for the most popular Web servers – Microsoft IIS, Apache, Netscape Enterprise Server, and iPlanet Web Server. BlueDragon is also packaged in a version for J2EE (Java 2, Enterprise Edition) that allows you to deploy CFML applications onto any standard J2EE server – including BEA WebLogic, IBM WebSphere, JBoss, Macromedia JRun, New Atlanta ServletExec, Apache Tomcat, and others.

Is BlueDragon a competitor to Macromedia's ColdFusion MX? Well, some people choose to view it that way, and maybe that's the easiest way to understand BlueDragon. But that's not the way we view it. If BlueDragon was merely a feature-for-feature clone of CFMX, there really wouldn't be much value to it. Instead, BlueDragon offers features, packaging, configuration, and deployment options that aren't supported by CFMX, thereby expanding your range of choices as a CFML developer (to be fair, there are also features, packaging options, etc., offered by CFMX that aren't supported by BlueDragon).

Building Dynamic Web Sites

Our reasons for creating BlueDragon can best be summed up by the phrase: CFML forever! New Atlanta and its founders have long histories building dynamic Web sites, dating back to early 1995 before the creation of Java, CFML, or Web application servers. We've developed Web sites and products using every major server-side scripting technology: CGI, Perl, ISAPI, NSAPI, JSP, ASP, PHP, CFML, etc.

The leading product for publishing FileMaker Pro databases on the Web – Lasso – was created by the founders of New Atlanta; the core technology of Lasso was incorporated into FileMaker Pro itself in 1997, and both Lasso and FileMaker Pro are still widely used today (primarily on Mac OS, but also on Windows and Linux). New Atlanta's ServletExec was the second commercial implementation of the Java Servlet API, shipping a few months after JRun in 1997. New Atlanta has been, and still is, represented on every Expert Group for every version of the Java Servlet and JavaServer Pages (JSP) ever published by Sun Microsystems.

The point is: we know server-side scripting technology, and we like CFML! Every CFML developer who is half awake knows that during the past two years or so the alphabet soup of server-side scripting technologies has been reduced to three market leaders: JSP, ASP, and PHP. These are all good technologies, and all have particular strengths to recommend them, but none offer the same ease-of-use and immediate productivity of CFML, particularly for Web site designers who do not have a strong background in "traditional" programming languages such as C/C++ or Java. CFML is absolutely unique and is a technology that deserves to survive and thrive as a legitimate alternative and complement to the "big three."

New Atlanta, through the BlueDragon product family, is commited to helping ensure that your investment in CFML over the years – in terms of the code modules, custom tags, and solutions frameworks that you've created, and in terms of the education, training, and techniques you've developed through hard experience – does not become marginalized or devalued in the face of the market dominance of JSP, ASP, and PHP.

A Standalone Server

As I mentioned above, BlueDragon is offered as a standalone server, with its own built-in Web server (for development) and adapters for the most popular Web servers (for deployment). We think this is an important product configuration and that there are many, many cases where you'll want to use CFML as a "standalone" technology. However, one of the key advantages that JSP and ASP have over CFML (and PHP) is that they're built on "platform" technologies – J2EE in the case of JSP, and .NET for ASP. These platform technologies provide access to a rich set of functionality, as well as robustness and scalability, which simply can't be matched by a standalone CFML server.

BlueDragon aims to provide a high level of native integration with the J2EE and .NET platforms, to the extent that you can use CFML as a complete alternative to JSP or ASP for developing native J2EE or .NET Web applications. Of course, you'll also be able to deploy CFML side-by-side with JSP and ASP as fully integrated native components of J2EE or .NET Web applications. Think about this for a moment, because it's a subtle but important point. In the past your CFML could only be deployed on proprietary ColdFusion servers from Allaire or

Macromedia; that is, when you developed using CFML you were a ColdFusion developer, period. With BlueDragon, when you develop using CFML you can be a J2EE developer, or a .NET developer, or a ColdFusion developer, or all three at the same time!

Our goal is this: if you're a CFML consultant or solutions provider and one day a client asks you (or already has): "Can you develop a J2EE application for me?" we want you to be able to answer, "Yes!" If you're an in-house developer and your CIO says one day, "We've decided to standardize on .NET, what are we going to do with all those ColdFusion applications?" we want your answer to be, "No problem." The key point is, and this bears repeating, that our goal is to make your CFML pages work as fully integrated native components of J2EE or .NET Web applications, not as a separate component or add-on extension.

The version of BlueDragon for J2EE Application Servers that is currently shipping is implemented as a standard Java servlet that allows you to deploy CFML pages within a standard J2EE Web application or Web Archive (.war) file. These Web applications can be deployed to any J2EE server that supports recent versions of the Java Servlet API and JavaServer Pages (JSP) specifications. There are not separate versions for J2EE servers from different vendors; there doesn't need to be because BlueDragon is a standards-based implementation.

A number of lead customers have successfully ported their CFML applications to BlueDragon/J2EE – these applications are being deployed on BEA WebLogic, Borland Enterprise Server, and New Atlanta ServletExec, and have been tested using Apache Tomcat. Two of these applications consist of over 2,500 CFML pages each, and one is based on Fusebox. I'll provide details of their experiences and the BlueDragon/J2EE architecture in future columns.

BlueDragon for .NET

BlueDragon for .NET is currently under development with early alpha versions running in our lab. This version of BlueDragon is fully implemented as managed code that executes within the the .NET Common Language Runtime (CLR). We're very excited about BlueDragon for .NET with its promise of high-performance native integration with ADO.NET and COM. Look for a formal announcement of BlueDragon for .NET either before or shortly after this column is published.

Finally, let's not forget about PHP and the open-source platform in general. New Atlanta is fully committed to supporting open-source technologies such as Linux, the Apache Web server, JBoss, and Tomcat. We've investigated creating a version of BlueDragon that is integrated with the PHP runtime, but at this point we're not sure it really makes sense. PHP is not built on a platform technology such as J2EE or .NET, but instead runs on top of the "bare" operating system and Web server. Therefore, while integrating BlueDragon with the PHP runtime might be an interesting technical exercise, it doesn't seem that it would result in the same benefits as BlueDragon for J2EE or BlueDragon for .NET.

A Free Version

However, we recognize that among the key benefits of PHP, and open source generally, is the fact that it's free (that is, it's available at no cost). This can be a very powerful factor for some people when choosing a technology, overwhelming all other decision-making criteria. In order to remove this "disadvantage" from CFML, New Atlanta will soon announce (or has already announced, depending on the timing of this column's publication) a free version of BlueDragon. This will be a standalone server, with adapters for popular Web servers, which will be completely free for development and deployment. There will be some advanced features not supported by this free version, but it will be much more feature-rich than the old ColdFusion Express that you may remember. Our intention is to make this a truly useful product, not just a "teaser," and we hope you'll give it serious consideration for your CFML deployments.

We at New Atlanta hope that you'll view BlueDragon as a new tool to help you remain a productive, relevant, successful CFML developer for many years to come. CFML forever!

About the Author

Vince Bonfanti is president and cofounder of New Atlanta Communications, developers of Java- and CFML-based server products. A charter member of Sun's JavaT Servlet API and JavaServer PagesT Expert Groups, Vince has been a JavaOne speaker and a contributor to Java trade magazines and online publications. He has also been a featured speaker at Toronto's CFNorth and Washington's CFUN conferences as well as at local ColdFusion User Groups around the country.

vince@newatlanta.com

Fun with Filters in CFMX

An introduction to servlet filters

There are a lot of powerful new integration points between CFMX and Java, but one that might benefit a lot of people – even those who don't know how to program in Java – is the ability to leverage servlet filters in CFMX.

By Charlie Arehart

A filter is a Java program that can be used to execute before and/or after your CF template. More than just running some code in advance, a filter can also manipulate the request (the HTTP stream) that is calling your template, and can change how, or even if, your template is run, or cause some other template or code to run first. It can also manipulate the output (the HTTP response) that your template generates.

ColdFusion MX can run servlet filters. You don't need to write them because some have already been written for servlet and JSP developers. Because ColdFusion MX runs atop a J2EE server, we can leverage these or even write our own filters if we want to.

In this article, I'll show you where to find some example filters and how to implement them in CFMX. You don't need to understand Java to use them. By the way, they work in both the Enterprise and Pro (as well as developer and trial) editions of CFMX, and also with the built-in Web server in CFMX and external Web servers like IIS and Apache.

How Do Filters Compare to Application.cfm?

Filters may sound very similar to the way Application.cfm (and OnRequestEnd.cfm) work. But a filter has many added benefits and is really quite different from those two traditional mechanisms in CF. Before getting into the details of filters, it may help to first contrast them to those more traditional approaches.

Most will know that Application.cfm and OnRequestEnd.cfm are templates that run before (and after, in the case of OnRequestEnd.cfm) a ColdFusion template is executed. They provide a means to effect some kind of pre-or post-processing in order to augment your template.

Some use them to add a test (such as security) before a template runs, or to create some data structure (perhaps persisted in the application scope) to be shared by many templates. Some even use them for creating headers (and footers), though that may get complicated if you have a template that should run without the headers or footers.

Some Challenges with Application.cfm/OnRequestEnd.cfm

One problem with the traditional CF approach is that there's no way to stop these two files from running for any one template, as might be desired when trying to use them for headers and footers. (Of course, using them for headers and footers may be an arguable approach, but it does point up a limitation with them.) Any time you run a CF template, CF looks to run any existing Application.cfm in the current or a parent directory, whichever it finds first. And if there's an OnRequestEnd.cfm in the same directory as the Application.cfm that's run, that will be executed at the conclusion of the named template.

You can't tell CF to *not* run these files if they exist. Some of the other challenges are:

- You can't apply them selectively to only some templates in a directory
- You can't run more than one. As soon as CF finds the closest one (in the current or an ancestor directory), it runs that one and looks for no more
- You can't have just an OnRequestEnd.cfm. It only runs if there is a corresponding Application.cfm in the same directory
- If you do a CFABORT, any OnRequestEnd.cfm is not executed
- They apply only to CF templates, not to HTML pages or indeed to JSPs or servlets running alongside your CF templates (which is now possible in CFMX, though JSPs can run in CFMX Enterprise only)

Now, many developers haven't worried too much about these limitations. Heck, some never even knew about OnRequestEnd.cfm. It came out in 4.01 with little fanfare. See my February 2000 article, "Hidden Gems in 4.0.1 – What You Might Have Missed," at www.sys-con.com/coldfusion/article.cfm?id=81 for more such surprises.

So why should they get excited about servlet filters? What do they really add?

Where Filters Augment Traditional CF Approaches

Filters work differently from the traditional approach of automatically running Application.cfm and OnRequestEnd.cfm. For one thing, you have much greater control of when a filter is run. Through some configuration files built into CFMX, which we will see later, you map filters to a given URL pattern, so that they can be made to apply to:

- A single template
- All templates in a given directory and subdirectories
- All templates of a given file type
- All templates on the entire server

Try *that* with Application.cfm! They can also apply not just to requests for CF templates but also even HTML or image files, for instance.

More important, because these are added declaratively through configuration files (rather than controlled by code in the template directories) you can also:

- Designate multiple URL patterns for a given filter
- Designate multiple filters to a given pattern
- Add/remove/modify filters without the templates knowing about it
- Apply changes to a filter across several applications

Remember that filters can either process the incoming request or the outgoing response (HTML result, typically) of a CF template (or servlet or JSP if running under CFMX). What kinds of things could a request (input) filter do? To name a few:

- Log and audit visits
- Block execution of a page for a given time period (perhaps during a maintenance period)
- Block visitors from given domains
- Trigger events when a resource is accessed
- Cache/reuse a page's results
- Modify the input request before processing by CF
- Test for/handle cross-browser scripting attempts
- Modify request character encoding

It's true that CF developers can do some of these things themselves in Application.cfm processing, but remember that they may already be written as servlet filters, thus saving you the development effort. Also, remember the greater flexibility in mapping filters to URLs.

What could a response (output) filter do? Again, this is just a short list of creative solutions that do exist or could be created to process the output of a template before returning it to the user:

- Log page execution time
- Localization (targeting output to a locale)
- Image conversion
- XML transformations of XML output
- Encryption
- Data compression (gzip)

I mentioned that filters have been written and used by JSP and servlet developers for some time. There are public repositories of them that address all sorts of useful programming solutions, some of which are just as valuable to CF developers.

Where Can I Find Them?

As for where you can find these already written, there are a few places including jsptags.com and servletsuite.com. There are surely more, but the latter has quite a few that might interest CFers, including:

Cache filter
www.servletsuite.com/servlets/cacheflt.htm
Billing filter
www.servletsuite.com/servlets/billflt.htm
Profiler filter (track execution times)
www.servletsuite.com/servlets/proffilt.htm
IP filter (access restriction, load balancing)
www.servletsuite.com/servlets/ipflt.htm
Zip/Compression filter
www.servletsuite.com/servlets/gzipflt.htm

The Power of a Compression Filter

That last item in the list, a compression filter, may be the most compelling one for many and, I'll give a specific example of one as a demonstration of how all this works. A compression filter, also often called a Zip or gzip filter, is a tool that compresses the output of a Web page so that it generally takes up far less space when being sent from the server to the browser. The browser can

then decompress the page and render it as usual to the end user.

CF pages are notorious for having excessive white space, so a compression filter can be a big win for us. The good news is that most browsers, even relatively old ones, can support decompression. A compression filter will determine if a browser can support compression by checking the HTTP header called accept-encoding (or in CF, you could test it with cgi.http_accept_encoding) before trying to compress the page. A smart one might also balance the size of the page being processed before trying to compress it, since compression does cost some CPU cycles on both the server (to compress) and the client (to decompress) page.

Still, for all but the most trivial page, compression is generally a good thing, and it's rather easily tested to confirm the savings. In some testing I did, it resulted in a 4:1 reduction in bandwidth (size of pages downloaded to the browser). For sites that pay for their bandwidth utilization (someone is always paying for it), or just for the speed improvement of sending "lighter" pages to the browser, compression is valuable.

(It's worth noting that both Apache and IIS offer their own forms of compression that, if implemented properly, will generally suffice to provide the benefits that compression can bring. But if they're not set up, or not set up properly, then a filter in CFMX may be a good way to go. In fact, I wonder if in the future Macromedia may build one in and make it more simply enabled via an Administrator setting. Until then, it's a good example of a filter.)

How to Implement a Filter in Three Easy Steps

It's incredibly easy to implement a filter. If you visit any of the pages mentioned here, you'll find explanations of the filters, the downloadable files, how to configure them, and maybe an example that might be written as a JSP page. Most don't even show that, because there's nothing necessarily specific about using a filter with either CF or a JSP page, for instance. You just associate the filter with a given URL, and the filter doesn't care what kind of page it's processing before or after.

The only clue that the filter is written in Java is that the downloadable files will likely refer to a JAR (Java Archive). But all you need to know is where to put that file and how to set up CFMX to map the filter to a URL. That's really all there is to it. Indeed, the explanations on the page will often be very similar if not identical to what you need in CFMX. There are just three simple steps. I'll walk you through implementing that gzip/compression filter.

- ***Step 1:*** **Download the filter (JAR file):** Upon visiting the page that holds the filter, such as the last one for the gzip filter mentioned above at www.servletsuite.com/servlets/ipflt.htm, you'll find a link to the JAR file that needs to be downloaded and placed into the cfusionmx\wwwroot\WEB-INF\lib directory. It doesn't matter if you use another Web server or store your CF templates somewhere other than the cfusionmx\wwwroot. That directory will still exist, and under it you will find the WEB-INF\lib subdirectory. Place the file there. This is the location in which J2EE servers (like that underlying CFMX) expect to find Java programs like filters, when they are packaged as JAR files. (If you happened to get a filter or servlet that was just a "class" file, rather than a JAR, it would be placed in the cfusionmx\wwwroot\WEB-INF\classes directory.)

- ***Step 2:*** **Implement the filter mapping (XML configuration):** As the page showing the downloadable filter may also indicate, the next step is to modify the web.xml file to define a pointer and map to the filter. Again, this file does exist in CFMX, specifically in the cfusionmx\wwwroot\WEB-INF. The directions will likely indicate just the XML that you need to add to the web.xml file. In the case of the compression filter, it would be:

```
<filter>
   <filter-name>GZIPFilter</filter-name>
   <filter-
class>com.cj.gzipflt.GzipFilter</filter-class>
</filter>

<filter-mapping>
   <filter-name>GZIPFilter</filter-name>
   <url-pattern>*.cfm</url-pattern>
</filter-mapping>
```

Note that in the next to last line, we're indicating that this filter should apply against all files of type "cfm", and since we don't say otherwise, it means all such files in all directories that are run under CFMX. You could create additional filter-mapping entries to list other URL mappings. To just control files in a "test" directory under your webroot, the pattern could be "/test/*.cfm" or simply "/test/*".

Of course, as when modifying any of the XML files in the CFMX configuration, be very careful. It may even be wise to save a backup of the file before editing it so that you can recover in case the server fails to start, etc.

- ***Step 3:*** **Restart the CFMX server:** The last step is to restart the server. CF will not detect the filter unless the server is restarted. It may not seem obvious that this should be so, but it actually looks for and preprocesses any filters at startup.

Conclusion

This has been a pretty quick introduction to servlet filters. You'll notice we didn't look at a single line of Java code. We're not interested here in creating filters, just in using them. There are plenty of resources for learning more about creating them – though there's not too much more you need to know about using them. We've run out of room to cover any further details, but if you're interested in more on this subject, including a few other tips that may help you as you explore them, take a look at the slides from my one-hour user group presentation, "Augmenting Application.cfm with Filters," available at my site at www.systemanage.com/presentations.

About the Author

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

carehart@systemanage.com

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

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

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

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

3rd Annual EAST

Web Services Edge Conference & Exposition

FINAL PROGRAM

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

March 18–20, 2003
Hynes Convention Center
Boston

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

Event Sponsors

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

Media Sponsors

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

OWNED BY SYS-CON MEDIA

PRODUCED BY SYS-CON EVENTS

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

web services EDGE conference&expo

JDJ EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

THE LARGEST WEB SERVICES CONFERENCE & EXPO

FEATURES & ATTRACTIONS

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

WHO SHOULD ATTEND

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

Keynotes & Highlighted Speakers

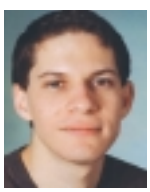
KEYNOTE SPEAKERS

John Magee
Vice President, Oracle9i

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

Mark Herring
Director Java, Web Services & Tools Business

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



Miguel de Icaza
Cofounder and CTO

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

Mark Hapner
Distinguished Engineer, Sun Microsystems

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

Simon Phipps
Chief Technology Evangelist, Sun Microsystems

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

Dave Chappell
VP, Chief Technology Evangelist, Sonic Software

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

Eric Newcomer
Chief Technology Officer, IONA

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

Dean Guida
CEO and President, Infragistics

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

Conference Advisory Board

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

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

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

Bob Familiar
.NET Architect,
Microsoft
New England

Thomas Robbins
Senior Technology Specialist,
Microsoft
New England

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

J.P. Morgenthal
Chief Services Architect, Software AG

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Conference at-a-Glance

THE LARGEST AND THE MOST

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

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

JAVA UNIVERSITY℠ PROGRAM

9:00AM -5:00 PM

Web Services Programming Using Java™ Technology and XML

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

9:00AM -5:00 PM

Java 2 Platform Programmer Certification Fast Path

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

9:00AM -5:00 PM

Java 2 Platform Architect Certification Fast Path

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

FAST TRACKS & TUTORIALS

9:00AM -5:00 PM

XML Certified Developer Fast Path

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

9:00AM -5:00 PM

Russ Tool Shed

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

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

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

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

9:00AM -5:00 PM

Mobile .NET

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

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

SUN MICROSYSTEMS Java University℠ Program

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

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

Java University℠ Program Take-Aways:

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

Tuesday, March 18, 2003

Web Services Using Java™ Technology and XML

SANG SHIN, SUN MICROSYSTEMS, INC.

Who Should Attend

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

Prerequisites

Experience using the Java programming language and basic knowledge of XML

Overview

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

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

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

Benefits

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

Wednesday, March 19, 2003

Java™ 2 Platform: Programmer Certification Fast Path

PHILIP HELLER, PRESIDENT, HELLER ASSOCIATES

Who Should Attend

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

Prerequisites

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

Overview

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

Benefits

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

Thursday, March 20, 2003

Java™ 2 Platform: Architect Certification Fast Path

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

Who Should Attend

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

Prerequisites

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

Overview

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

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

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

Benefits

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

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

2003 East Gold Sponsors

COMPUWARE

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

Microsoft

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

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

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

ORACLE

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

2003 East Silver Sponsor

hp invent

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

2003 East Featured Sponsor

Sun microsystems

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

2003 East Corporate Sponsors

IONA

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

Rational the e-development company™

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

SAMS

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

SYBASE

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

DIGITAL EVOLUTION
The Business Value of Web Services

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

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

REGISTER ONLINE TODAY @ SYS-CON.COM

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

Don't Miss the Developer Event of 2003

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

Conference
March 18–20, 2003

Exposition
March 19–20, 2003

Hynes Convention Center, Boston

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

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

Ben Forta Defends His Position

("But It's Free!" [*CFDJ*, Vol. 4, issue 12])

Dear Ben:

In all my time as a CF developer, I never thought I'd see the day when undisputed CF Guru Ben Forta would speak about something he wasn't qualified to speak about. I read your article, aptly named, "But It's Free!" in *ColdFusion Developer's Journal* (Vol. 4, issue 12), and shook my head in disbelief. The first thing that came to mind as I began reading was, "Isn't Ben an employee of Macromedia?"

I've seen this hundreds of times in CF discussion areas where CF enthusiasts on MM's payroll come out and thunder their way through this political minefield. Aside from that, it's unfortunate that you missed the point behind why people are saying that PHP, Perl, and mistakenly ASP and Java are free, and therefore a better option. I agree with your deduction that ASP and Java are free as in beer. Therefore there isn't any advantage other than price, as your article accurately shows.

Ignoring ASP and Java for now, the reason that PHP, Perl, and others like them are free is because they are free as in speech, not free as in beer. This means that people can make changes to the source, customize it to their needs, and fix it if necessary because the source code is freely available to whoever wants to browse it.

Instead, CF is closed source and completely controlled by one company, so we find ourselves with such discussion topics on the Web as "Cold Fusion MX Using Hundred of Processes and Killing Server (II)" because the company in control did not respond to it as a problem until it became an epidemic. ("Cold Fusion MX Using Hundreds of Processes and Killing Server (II)" is the second edition of a very long topic on the developers' forum on MM's Web site. You'll have to do a search for the first edition, which was so long that you couldn't download it anymore, even using broadband. http://webforums.macromedia.com/coldfusion/messageview.cfm?catid=10&threadid=489312.)

This is the reason PHP and Perl are free. This is why we need a "Free ColdFusion" movement. This is why you get bombarded by e-mails, albeit some misguided, for free is better.

Finally, in case you were going to throw some anti-"free as beer" stuff at me, which is not the topic of my e-mail: there are companies that make money selling their open-source products – companies like Red Hat, IBM, and Sun to name a few. Rather than rant and rave about open source, I would like to recommend a good book: *Just for Fun: The Story of an Accidental Revolutionary*, by Linus Torvalds and David Diamond.

Spurred on by your article, I'm going to start a Web site (www.cfopensource.org) to talk and to discuss the differences between free as in beer and free as in speech, and how it will affect CF developers. I encourage you, and others with like minds, to join me in this cause.

–Paul Woods
ColdFusion Open Source Advocate
paul@paulwoods.ca

Ben Forta responds:

Paul,

Thank you for your feedback. You are correct, there is a distinction to be made between options like ASP and options like PHP. And my comments in the column definitely are more applicable to ASP-type options (which is exactly why the majority of the column did indeed use ASP as an example).

You are also correct that there are different types of "free." However, the point of my column was that initial cost of goods is not the entire cost – ongoing cost of development and cost of included services must be taken into account. That is true of both the ASPs and PHPs. PHP in particular has very rich function libraries (and even has functions that CFML does not), but what CFML does do, it does far easier and simpler, there is no disputing that. And as such, the primary message of the column is very applicable even to PHP.

As to "people can make changes to the source, and therefore customize it to their needs and fix it if necessary because the source code is freely available to whoever wants to browse it," that is true. Although most people don't – most don't ever even look at the source code – the appeal is more that the source code is available to those who want it, allowing for a more diverse group of contributors to the product. Open source, its merits, and the future of ColdFusion in an open-source market is an important discussion to have, but it deserves (and requires) a forum unto itself, and is definitely outside the scope of that column.

–Ben

Macromedia Ships JRun 4 for MAC OS X

(*San Francisco*) – Macromedia, Inc., has announced the availability of Macromedia JRun 4 for Mac OS X. JRun 4 is the first commercial J2EE-compatible application server to support Mac OS X. Currently used at more than 10,000 companies worldwide, JRun 4 is the fast, affordable, and reliable solution for delivering Java applications. JRun 4 for Mac OS X is immediately available for purchase from the Macromedia Online Store at www.macromedia.com/go/buyjrun4.

"With the industry's most extensive support of standards, Mac OS X is the best platform for Java development and deployment," said Ron Okamoto, Apple's vice president of worldwide developer relations.

Christian Cantrell Named Macromedia Server Community Manager

(*San Francisco*) – Christian Cantrell has been named Macromedia Server Community Manager. He has been developing large-scale, Web-based applications in ColdFusion, Java, JSP, and Macromedia Flash for the last five years. He is the author of numerous tutorials and white papers, and is coauthor of *Flash Enabled: Flash Design & Development for Devices*. You can keep up with Christian by reading his blog at http://markme.com/cantrell/weblog.

Macromedia Flash Communication Server MX, by Kevin Towes, Now Available

Macromedia Flash Communication Server MX is a complete guide to integrating rich media and real-time collaboration into applications with Flash Communication Server MX. It covers ActionScripting, Flash Remoting, and other key implementation issues from a server-side perspective. It provides excellent tutorials for creating usable, media-rich applications that allow real-time collaboration.

Kevin Towes is cofounder and CTO of Pangaea NewMedia in Toronto, Canada. He is a certified Macromedia ColdFusion Developer and has led his team at Pangaea to achieve much success with Macromedia products for many years. Kevin is an industry leader in the fields of Macromedia Flash, ColdFusion, and database integration. Watch for his article in an upcoming issue of *CFDJ*!