WHAT'S NEW IN COLDFUSION 4.5

U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

ColdFusionJournal.com

March 2000 Volume: 2 Issue: 3

Announcing... XML DevCon 2000
Coming June 25-28, 2000
JavaCON 2000 September 24-27, 2000

SYS-CON PUBLICATIONS

AbleCommerce

The flexible solution for building electronic shopping sites

AbleCommerce

CLIENT LIST

A few of our e-shopping customers

- 24 Hour Fitness
- Bushnell
- Johns Hopkins University
- BOC Gases
- Key Tronic
- Casio
- Citgo
- GSA
- MADD
- Lucky Brand Jeans
- Compaq
- Mercury Marine
- Dairy Farmers of America
- Mitsubishi
- FAA
- Nike
- Garth Brooks
- Miss Universe/USA/Teen USA
- NorWest Financial
- George Strait
- PGE
- Edwin Watts Golf Shops
- Qwest
- Hasbro
- Siemens
- U.S. Department of Treasury
- International Paper
- United Nations
- Jimmy Buffett
- Jeff Gordon
- Verio
- Swiss Army Knives
- New England Patriots

To see more, visit www.topsites.com

Complete Site Building and Administration from your browser.

Build "your_store.com" with secure Merchant Credit Card Processing. Maintain inventory, add discounts and specials to keep your customers coming back. Increase sales with cross selling and membership pricing. For accurate shipping use AbleShipper™ UPS auto-calculator with the flexible AbleCommerce™ shipping tables. Fulfill document and software purchases through electronic delivery. Check orders online, generate automated email and fax notification.

Enterprise Scalability for stores of any size.

Supports Microsoft Access and SQL. Add store and server license packs to create an unlimited number of stores. Two security levels for system and merchant administration allow ISP's to empower renters with online store design and maintenance abilities.

You can also integrate and customize AbleCommerce™ with source code.

Easy Design Interface for eye catching sites.

Upload graphics directly to your server. Use prebuilt store templates for quick design. Customize your web pages and styles dynamically. Add you own HTML, Java, Flash or any other script to create a unique look. Add comment forms and surveys with automated responses to collect data from your visitors. Track visitor click throughs to determine the strengths of your site design.

> ***AbleCommerce is a perfect example of ColdFusion's strength and flexibility.***
>
> -Ben Forta, Allaire Corp.

> ***The system is easy to use and very flexible from both a customer and administrator standpoint.***
>
> -Rob Derby, Casio, Inc.

TRY IT OUT FREE!

Download the 30-Day Evaluation

www.ablecommerce.com

AbleCommerce, 11700 NE 95th Street, Suite 100, Vancouver, Washington, 98682, (360) 253-4142

AbleCommerce is a trademark of Able Solutions Corporation. ColdFusion is a trademark of Allaire Corp. Company or product names referenced may be the trademark or registered trademark of their respective companies. © Able Solutions Corporation, All Rights Reserved.

Able Solutions

www.ablecommerce.com

Digital Nation

www.dedicatedserver.com

ColdFusion Developer's Journal

EDITORIAL ADVISORY BOARD
STEVEN D. DRUCKER, JIM ESTEN, BEN FORTA,
STEVE NELSON, RICHARD SCHULZE, PAUL UNDERWOOD

EDITOR-IN-CHIEF ROBERT DIAMOND
ART DIRECTOR JIM MORGAN
EXECUTIVE EDITOR M'LOU PINKHAM
SENIOR EDITOR JEREMY GEELAN
PRODUCTION EDITOR CHERYL VAN SISE
ASSOCIATE EDITOR NANCY VALENTINE
PRODUCT REVIEW EDITOR TOM TAULLI
TIPS & TECHNIQUES EDITOR MATT NEWBERRY

WRITERS IN THIS ISSUE
ROBERT DIAMOND, BEN FORTA, HAL HELMS,
EMILY KIM, MARK MURPHY, SHAHRIYAR NEMAN,
MATT ROBERTSON, LEE SURMA, BRUCE VAN HORN

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS,
PLEASE SEND YOUR LETTERS TO
SUBSCRIPTION DEPARTMENT.

SUBSCRIPTION HOTLINE 800 513-7111
COVER PRICE $8.99/ISSUE
DOMESTIC $79/YR. (12 ISSUES)
CANADA/MEXICO $99/YR
OVERSEAS $129/YR
BACK ISSUES $12 EACH

PUBLISHER, PRESIDENT AND CEO FUAT A. KIRCAALI
VICE PRESIDENT, PRODUCTION JIM MORGAN
VICE PRESIDENT, MARKETING CARMEN GONZALEZ
ACCOUNTING MANAGER ELI HOROWITZ
ADVERTISING ACCOUNT MANAGER ROBYN FORMA
ADVERTISING ASSISTANT MEGAN RING
ADVERTISING ASSISTANT CHRISTINE RUSSELL
GRAPHIC DESIGNER ALEX BOTERO
GRAPHIC DESIGNER JASON KREMKAU
GRAPHIC DESIGNER ABRAHAM ABBO
GRAPHIC DESIGN INTERN AARATHI VENKATARAMAN
WEBMASTER BRUNO Y. DECAUDIN
WEB EDITOR BARD DEMA
WEB SERVICES INTERN DIGANT B. DAVE
CUSTOMER SERVICE MANAGER CAROL KILDUFF
JDJ STORE.COM JACLYN REDMOND
ONLINE CUSTOMER SERVICE AMANDA MOSKOWITZ
CUSTOMER SERVICE ANN MARIE MILILLO

EDITORIAL OFFICES
SYS-CON PUBLICATIONS, INC. 39 E. CENTRAL AVE.,
PEARL RIVER, NY 10965
TELEPHONE: 914 735-7300 **FAX:** 914 735-6547

COLDFUSION DEVELOPER'S JOURNAL (ISSN #1523-9101)
IS PUBLISHED MONTHLY (12 TIMES A YEAR)
FOR $79 BY SYS-CON PUBLICATIONS, INC.,
39 E. CENTRAL AVE., PEARL RIVER, NY 10965-2306.

POSTMASTER
SEND ADDRESS CHANGES TO:
COLDFUSION DEVELOPER'S JOURNAL
SYS-CON PUBLICATIONS, INC.
39 E. CENTRAL AVE., PEARL RIVER, NY 10965-2306

© COPYRIGHT
COPYRIGHT © 2000 BY SYS-CON PUBLICATIONS, INC.
ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE
REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS,
ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY OR ANY
INFORMATION STORAGE AND RETRIEVAL SYSTEM,
WITHOUT WRITTEN PERMISSION.

FOR PROMOTIONAL REPRINTS, CONTACT REPRINT COORDINATOR.

SYS-CON PUBLICATIONS, INC., RESERVES THE RIGHT TO REVISE,
REPUBLISH AND AUTHORIZE ITS READERS TO USE
THE ARTICLES SUBMITTED FOR PUBLICATION.

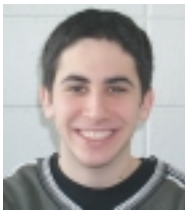
WORLDWIDE DISTRIBUTION
BY **CURTIS CIRCULATION COMPANY** 739 RIVER ROAD,
NEW MILFORD, NJ 07646-3048 PHONE: 201 634-7400

DISTRIBUTED IN USA
BY **INTERNATIONAL PERIODICAL DISTRIBUTORS**
674 VIA DE LA VALLE, SUITE 204, SOLANA BEACH, CA 92075
619 481-5928

ALL BRAND AND PRODUCT NAMES USED ON THESE PAGES
ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS
OF THEIR RESPECTIVE COMPANIES.

SYS-CON PUBLICATIONS

From Disney to Disasters



BY **ROBERT DIAMOND**

It was a bright, sunny day when, after years of construction, Disneyland opened in California in 1955, proclaiming itself the theme park of the future. Unfortunately, as good as it all looked, the park wasn't quite as ready for the hordes of visitors as it thought it was. On opening day almost all of the rides broke down…there were too few trashcans, causing overflows of garbage…the lines were far too long…and not enough water fountains were operating. To compound the situation, thousands of counterfeit invitations had been distributed, so the park was overloaded with many more visitors than it could handle. Because of this, the roads leading to Disneyland were jammed with bumper-to-bumper cars filled with some very cranky passengers. It was by no means a good beginning. Lucky for them, things seem to have turned out quite well today.

What does Disney have to do with ColdFusion? Not a lot really, on the surface – but it's a great analogy if you think of how many Web sites these days have the same kind of openings as Disneyland. The disasters that Web site problems can result in can be much more catastrophic than the teacups not spinning around. With ColdFusion and the wonderful world of the Web, the possibilities for disaster are much worse. Just look at all the news articles these days about teenage hackers breaking into large e-commerce sites and stealing the credit card numbers of thousands of customers.

The security problems that lots of sites are having aren't just little cracks in a big wall; it's a series of wide-open doors that are just inviting people to break in. Remember, for every single problem that's being caught and talked about, there are probably twice as many occurrences that we don't know about yet. The problems that are being reported don't involve any major hacking attempts or schemes. They're tiny – little problems that are relatively easy to find and exploit, and even easier to fix.

The simplest of the problems that are cropping up is that plaintext databases connected right to the Web are easily viewable because the data isn't hidden or encrypted in any way. Another common one is that people are using preset default user ID/password combinations instead of changing them to their own unique ones. Remember, if it's the default password, not only will you know it, but so will anyone else who has ever used the program. I shudder to think of all the bigger problems out there that no one knows about…or, worse than that, about the select bunch of individuals who are exploiting it.

So what's the cause of this problem? More important, what's the solution? In today's superfast Web world, the priority isn't on taking the time to do things right. Instead, it's on getting things done yesterday, which is the only time that's soon enough to keep up with the fast pace. The only goal these days seems to be putting things up online quickly. Today's priorities are all about the bottom line. How much money are you making?…How many people are coming to your site?…And – most of all – what can that do for your stock price?

Quality just isn't the main concern anymore – it's an afterthought. That's extremely evident throughout the industry. Look at how many patches are out for the software you use every day. It's all about getting the product out on time, no matter what, so the balance sheets for the quarter look nice. Why worry about the problem of security now and take the time to do it right, when you can worry about it later?…After something horrible happens, of course.

More time needs to be spent on doing things well, and that's the only true solution. In the long run your customers will appreciate it and your boss will appreciate it. And if you avoid a disaster, then your company's investors will appreciate it too.

ABOUT THE AUTHOR
Robert Diamond is editor-in-chief of ColdFusion Developer's Journal.

ROBERT@SYS-CON.COM

Structurizing

BY SHAHRIYAR NEMAN & MARK MURPHY

Build a more robust, scalable and potentially faster Web site

Using CF structures to store data allows you to access individual records without querying a database. Until now, successfully implementing a scalable, high-volume site has entailed five basic principles:

1. *Write tight, efficient code.*
2. *Use cached queries.*
3. *Set up a proper network architecture.*
4. *Use top-of-the-line hardware.*
5. *Implement a properly designed database.*

This article suggests a new alternative to cached queries (principle #2): for some sites, using ColdFusion structures to store database data may reduce the bottlenecks caused by excessive database calls, thus improving overall function.

We'll demonstrate how structures, used in conjunction with databases, can let you build a more robust, scalable and potentially faster Web site. We conclude with a rundown of the pros and cons of "structurized" databases so you can judge which solution – Cold Fusion structures, cached queries or a combination of the two – is best for your site.

Transforming Tables into Structures

The first step in structurizing a database is to load all data from tables into structures, which can either be local to the template or persistent in the server's memory. To transform tables into structures, you'll need to use a persistent variable scope so the data can be accessed from any template for the life of the server. To accomplish persistency, you prepend all structures with "server." For example, to load the results of a product category table into memory, you'd use the code in Listing 1.

You may have noticed that the top-level structure is the root product category structure itself. All individual categories are implemented as nested structures, which are keys of the parent structure and the structures themselves. Structure nesting allows you to perform category-wide operations such as structcount() and structkeylist() on different structure levels.

This type of structure methodology can be expanded to include tables with relationships. For example, to implement a one-to-many relationship between the product category table and the product table, you could implement deeper levels of nested structures. Product records are implemented as nested structures of their corresponding product category structures.

Your Database

To demonstrate adding product information to the product category structure, see Listing 2. As shown in the listing, creating structures that imitate one-to-many relationships is straightforward, but the task can become difficult to manage once nesting levels exceed three to four structures. Furthermore, building a structure system that can simulate one-to-one or many-to-many relationships, although not impossible, can be arduous.

Retrieving Information from Structures

Once all the information from a database table has been loaded into a structure, you can access that data directly. For example, to access the category name of product category 1 in Figure 1, see Listing 3.

To return the number of records in a structure, you'd simply use the structcount() function. To return a list of a structure's keys, use the structkeylist() function. To return an array of keys, use the structkeyarray() function.

If you attempt to access a structure key that doesn't exist, ColdFusion will return an error. You can avoid this by prechecking for the existence of that key using the structkeyexists() function.

Multiple records can be retrieved by looping through the structure (see Listing 4). *Note:* Due to the nature of RAM, the structure keys won't necessarily be returned in any particular order. Also, looping through nested structures requires traversing via nested loops. (Structure sorting and traversing is beyond the scope of this article.)

Performing Updates, Deletes and Inserts

Structures aren't meant to replace your database but to augment and enhance its function. It's important to remember that any updates performed on your structures must also be performed on your database.

To insert data or update a structure's data, you need to set that particular structure key(s) to the new desired value(s), the same way you would a simple variable assignment. ColdFusion provides structinsert() and structupdate() functions to accomplish these tasks, but the easiest method is to use absolute variable declarations. For instance, to change the category name key of product category 1, see Listing 5.

Updating the primary key of a structure is a bit trickier because the key itself is the structure. First use the structcopy() function to copy the existing structure into a new one, then delete the original structure. *Note:* Structure functions such as structcopy() and structdelete() affect the structure and all its nested structures (see Listing 6).

Deleting structures can be done easily by using the structdelete() function. For an example of a structure deletion, refer to the call to the structdelete() function in Listing 6.

Finally, structure locking is vital. The importance of properly locking structures can't be overstated, whether they're being read from or written to. If structures aren't locked properly, you could run into synchronization problems where the same structure is accessed or modified at the same time by two different processes,

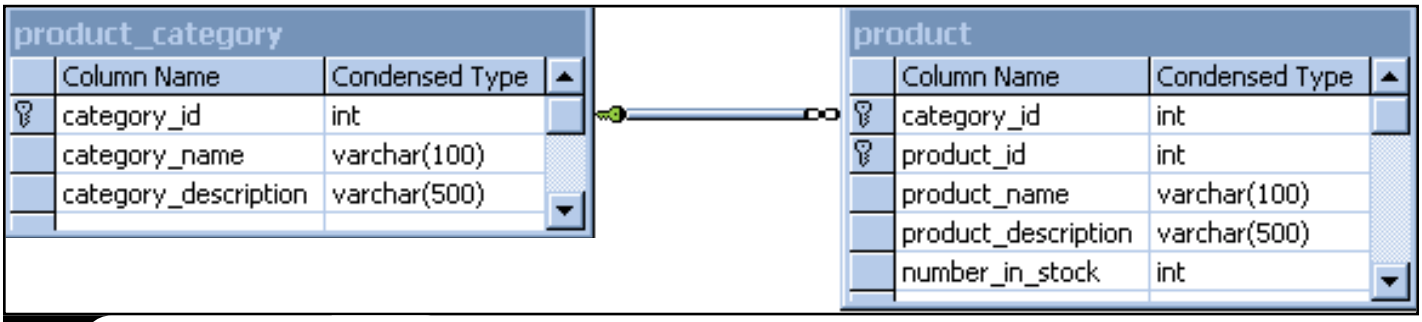


FIGURE 1 An excerpt of a typical e-commerce database schema illustrating a product_category and a product relation along with their attributes

resulting in potential data corruption. Structure locking can be accomplished with the <cflock> tag and isn't illustrated in any of the sample code provided.

Benefits of Using Structures

In theory, using structurized databases improves two important aspects of a Web site: speed and scalability. Storing data in structures can improve data-access speeds, especially in heavy load situations, because the structures reside in RAM. Rather than performing a database call, the Web server retains a copy of the database in memory and accesses it the same way it would a simple variable, significantly reducing data lookup times.

At this point you may be wondering why structures are better than cached queries. The answer is simple: a cached database query still requires calls to a database once data is modified or its cachedwithin attribute has expired. Structures, on the other hand, mirror the data in a database and can be likened to cached queries that never need to be refreshed. This attribute eliminates the need to access a database server when performing lookups. Structures also provide a means of exceeding ColdFusion's limit of 100 simultaneous cached queries. In essence, cached queries provide a static view of a database, while structures themselves are the evolving database.

In terms of scalability, Web servers with structurized databases can theoretically support large amounts of load because they generate far less network traffic and database activity. Moreover, structures can greatly reduce the need for additional database servers since the query load on them is minimal. Simply add more ColdFusion servers to your cluster to handle the extra load!

Drawbacks of Using Structures

Although structures can improve a site's speed and scalability, they have some potentially serious drawbacks that deserve careful consideration. One problem could arise from the type of primary key contained in your database tables. Structures work best with tables that have a single-integer primary key. Using structures with tables that have other datatype or multiple primary keys isn't recommended as the key itself is part of the structure name and must follow ColdFusion variable-naming requirements. Furthermore, the use of integer primary keys will facilitate the enumeration and sorting of your structures.

The size of the database could also create a potential problem with structures. Large amounts of data require Web servers with large amounts of RAM for storing the structures. This can be costly, and there's a limit to the amount of RAM your server can hold. Since RAM isn't permanent memory, all structures that reside in it are lost when your server shuts down. To reinstantiate your structures, a template must be run at boot time to reload all data. Depending on the amount of data stored in the database, reloading structures can significantly increase the server's reboot time.

The final drawback entails replication of structures in Web sites that implement multiple Web servers. When a structure is modified, it's modified only on the server on which the modification occurs. Other servers won't be aware of the structure modification unless a replication system is in place. Developing a replication system can be difficult and adds an additional step to the structure-modification process. With replication in place, structure modification would require:

1. Modification of the structure itself
2. Modification of the database
3. Replication of the updated structure across all Web servers (which could take a long time, given multiple Web servers)

A simple way to develop a replication system could be to design a sender ColdFusion script that loops through a list of servers and posts structure modifications via WDDX packets to a receiver ColdFusion script that processes these modifications. Although this solution may seem straightforward, the scripts must take into account situations in which replication can't take place – for instance, when servers in its list are down or in the process of booting.

Is "Structurizing" Your Database the Right Solution?

Admittedly, structurizing a database is not an easy task. The various benefits and drawbacks of using ColdFusion structures this way can make it difficult to decide whether they're the right storage mechanism for your Web site.

For sites with relatively infrequent database updates, structures can be an excellent solution. For example, an e-commerce site with a relatively static product base could benefit from structures since updates are infrequent and replication wouldn't have to occur very often.

Sites that are highly dynamic, especially sites that frequently update their databases from user input, wouldn't reap the speed or scalability benefits of structurized databases because of the added code complexity and replication overhead. Perhaps the best solution to building more scalable and faster database-driven Web sites lies in a mix of traditional database calls and structures. For those sections of your site that are mainly lookup oriented, the right choice might be structures; for those sections that are update oriented, the right choice may be database queries.

About the Authors

Shahriyar Neman is a senior developer at Convey.com, a Web-based ASP that allows users to create, manage, deliver and commercialize their content. He holds a BA in computer science and biochemistry from New York University.

Mark Murphy is the director of development at Convey.com. Mark has a degree in information systems from McGill University.

shahriyar.neman@corp.convey.com

mark.murphy@corp.convey.com

Ektron

www.ektron.com

LISTING 1

```
<cfset datasource = "structures">

<!---query the database for the categories--->
<cfquery datasource="#datasource#" name="category_query">
 SELECT *
 FROM product_category
</cfquery>

<!---initialize the top level category structure--->
<cfset server.product_category = structnew()>

<!---loop through all the rows of the query--->
<cfloop query="category_query">
 <cfscript>
  //make a nested structure for the category data
 "server.product_category.record#category_ID#" = structnew();

 //make keys for this category's name and description
 "server.product_category.record#category_ID#.category_name" =
category_name;
   "server.product_category.record#category_ID#.category_description" = cat-
egory_description;
 </cfscript>
</cfloop>
```

LISTING 2

```
<cfquery datasource="#datasource#" name="product_query">
 SELECT *
 FROM Product
</cfquery>

<!---loop through all the products in the query--->
<cfloop query="product_query">
 <cfscript>
  //Nest each product two levels deeper than its product category to enable
easier data retrieval

  //make the product nested structure for this category if it hasn't been cre-
ated yet
  if (not isdefined("server.product_category.record#category_id#.product"))
    "server.product_category.record#category_id#.product" = structnew();

  //make the structure for this specific product
  "server.product_category.record#category_id#.product.record#product_id#"
= structnew();

 //fill this structure with data

"server.product_category.record#category_ID#.product.record#product_id#.pro
duct_name" = product_name;

"server.product_category.record#category_ID#.product.record#product_id#.pro
duct_description" = product_description;

"server.product_category.record#category_ID#.product.record#product_id#.nu
mber_in_stock" = number_in_stock;
 </cfscript>
</cfloop>
```

LISTING 3

```
<!---retrieve one category name from the structure (primary key is soft-coded
for demonstration purposes)--->
<cfset local_variable = 1>
<cfoutput>
 #evaluate("server.product_category.record#local_variable#.category_name")#
</cfoutput>
```

LISTING 4

```
<!---create a product category html table--->
<table border="1">
<tr>
 <td>Product ID</td>
 <td>Category Name</td>
 <td>Category Description</td>
</tr>

<!---loop through the product category structure--->
<cfloop collection="#server.product_category#" item="i">
 <cfoutput>
  <tr>
   <!---parse the word "record" out of the structure key--->
   <td>#right(i, len(i) - 6)#</td>
   <!---evaluate the category name and description--->
   <td>#evaluate("server.product_category.#i#.category_name")#</td>
   <td>#evaluate("server.product_category.#i#.category_description")#</td>
  </tr>
 </cfoutput>
</cfloop>
</table>
```

LISTING 5

```
<cfset local_variable = 1>
<cfscript>
 "server.product_category.record#local_variable#.category_name" = "New
Name";
</cfscript>
```

LISTING 6

```
<!---change product 1 from product category 1 to product category 2--->
<cfset original_product_category_ID = 1>
<cfset new_product_category_ID = 2>
<cfset product_ID = 1>

<cfscript>
 //make sure the new product structure for this product_category exists
 if (not
isdefined("server.product_category.record#new_product_category_ID#.prod-
uct"))
  "server.product_category.record#new_product_category_ID#.product" =
structnew();

 //copy the original structure to the new structure

"server.product_category.record#new_product_category_ID#.product.record#pr
oduct_ID#" =
structcopy(evaluate("server.product_category.record#original_product_catego-
ry_ID#.product.record#product_ID#"));

 //delete the original structure
 structdelete(evaluate("server.product_category.record#original_product_cate-
gory_ID#.product"), "record#product_id#");
</cfscript>
```

CODE LISTING

The code listing for this article can also be located at www.ColdFusionJournal.com

Watchfire

www.watchfire.com

ColdFusion 4.5 – What's New *Part 1*

Should you upgrade?

BY
BRUCE VAN HORN

If you're questioning whether to upgrade to ColdFusion 4.5, let me put that issue to rest for you: Yes! Even though the change from 4.0 to 4.5 is considered only a mid-level upgrade, it's actually a significant improvement.

The enhancements in this new release are too many to cover in just one article. This month I'll review the improvements to ColdFusion Studio; in a forthcoming issue I'll walk you through the changes to the ColdFusion Server and ColdFusion Markup Language (CFML).

Editor Enhancements

At first glance, ColdFusion Studio 4.5 looks much like 4.0 (see Figure 1). It still has the Resource Window on the left, the Editor Window on the right and the Main Toolbar and Quick Bar on the top.

The Editor Window still has the same three modes (selectable by the tabs at the top of the window): Edit, Browse and Design. There are a few noteworthy changes to the Edit mode (see Figure 2). Allaire has added the ability to split an open document into two panes. This feature allows you to view and edit two different sections of the same document at the same time – very handy for those long templates. You can select Split Editor mode from the Options menu or from a button on the Editor Toolbar (located on the left side of the Editor Window).

Another welcome addition to the Editor is the ability to collapse text. This feature lets you hide blocks of code so you can focus on the code you need. Simply highlight a block of text in the Editor and click on one of the little collapse buttons that appear in the editor gutter, or right-click on the selection and choose Collapse Selection from the menu. Your selected block will collapse into an icon on the page that shows just the first few characters of the code. To expand the block again, double-click on the icon.

Also worth mentioning is a new multiple-entry clipboard. Studio now keeps track of the last 36 entries copied to the clipboard (this limit is customizable). In addition to the regular Paste icon, three new icons on the main toolbar allow you to Show Clipboard, Paste All and Clear Clipboard. If you click on the Show Clipboard icon, you're presented with a drop-down window from which you can select the entry to paste into the active document. You can also assign your own keyboard shortcuts to these new options.

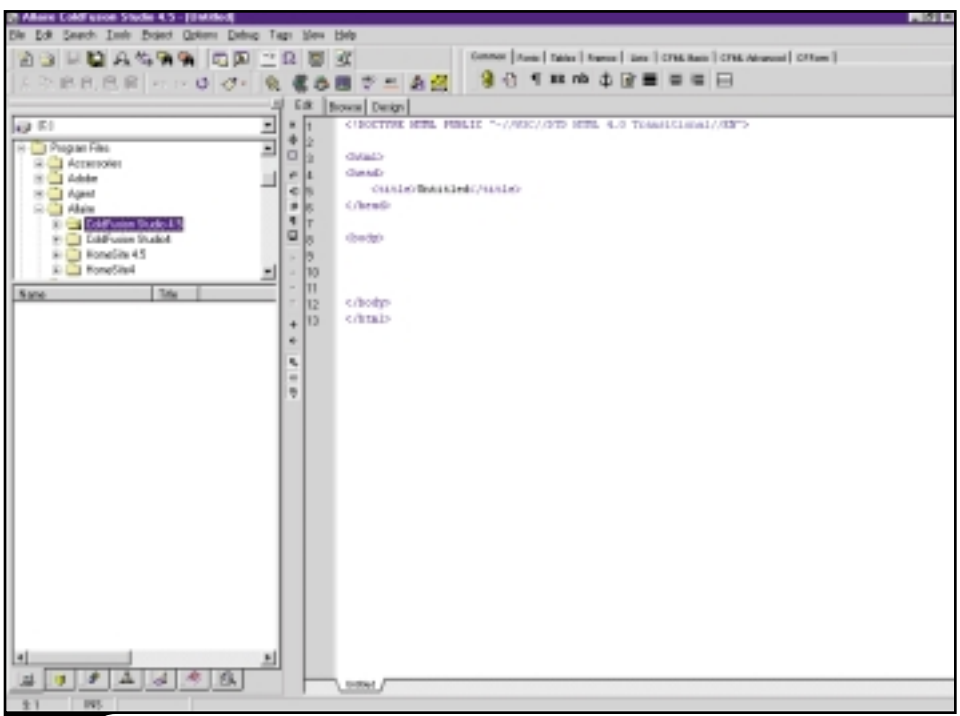


FIGURE 1: ColdFusion Studio 4.5 interface

The Movable Resource Window

In Studio 4.0 the Resource Window was resizable or could be hidden completely, but if visible it was available only on the left side of the screen. In 4.5, however, you can resize it, float it or dock it on the left, right, top or bottom of the screen. Not only can you dock or float the entire Resource Window, but you can dock or float individual sections of it. For example, you can float just the Databases window so you don't have to jump back and forth between it and the Files window.

The Universal File Browser

If you're like me, you'll probably install the upgrade and run Studio without even breaking the shrink-

Macmillan

wrap on the documentation. That's exactly what I did. My first question when Studio 4.5 came up was, "Where did the Remote Files tab go?" I clicked on all the tabs and browsed through all the menus but, alas, never found any reference to Remote Files.

Allaire has merged the Local Files and Remote Files tabs into one tab called "Files." The feature is actually called the Universal File Browser, and once you get used to it it's pretty cool. To see your local files, simply select the drive letter from the drop-down list and navigate to your particular subdirectory as usual. For remote files (on RDS or FTP servers) select the Allaire FTP & RDS option from the drop-down list. If you don't have any remote servers listed, right-click and choose Add RDS Server or Add FTP Server. Once the server is listed, simply double-click it to establish the connection. You can then navigate the directory structure of that server just as you would your local file system.

Another new feature of the Universal File Browser is one you might not notice unless you looked at the documentation. If you open up your Windows Explorer application, you'll now have a new listing with My Computer and all your drives: Allaire FTP & RDS. That's right. They've integrated the Universal File Browser into Windows Explorer so you can browse FTP (File Transfer Protocol) and RDS (Remote Development Service) servers without running Studio. This is a great feature if you're constantly copying files from a remote server to your local machine and vice versa.

> "Allaire has significantly improved the way Studio handles projects, so it's worth a new look if you've previously given up on it

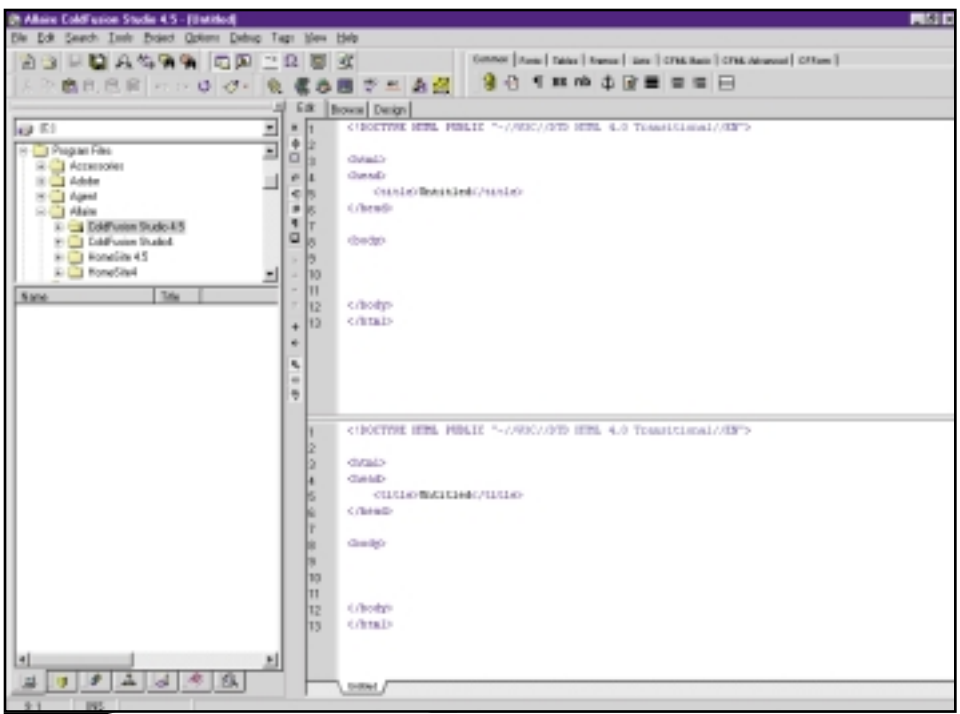


FIGURE 2: Split Editor mode

New Image Map Editor

As an instructor I'm constantly being asked to recommend other software packages to help with Web design where CF Studio falls short, particularly in the creation of Image Maps. Allaire's Fast Track to HTML class covers the use of image maps, but we haven't been able to demonstrate how to create them in Studio because until now it didn't have that capability.

Studio 4.5 now has a built-in image map to create "hot spots" on complex images. This is a full-featured image map editor that allows you to create square, polygon and circle hot spots. You can enlarge the image up to 250% for better detail. Simply draw your hot spot on the image, tell the editor which URL to jump to when that hot spot is clicked, then choose "Save and Exit." All of the necessary HTML is placed in the active document at the current cursor location. It doesn't get much easier than that.

Advanced Project Management

I have to admit that with all of the code I've written and the sites I've built through the years, I've rarely used the Project Management feature of Studio. Many other developers I talk to say the same thing. This may change, however, with the new release. Allaire has significantly improved the way Studio handles projects, so it's worth a new look if you've previously given up on it.

Essentially, you can create your own file system that contains only the files you need for developing your site (or project). There are many benefits to using Studio's project features: easy access to the files you need, streamlined deployment of those files and simplified code maintenance. You can work with or create new projects by clicking on the Projects tab in the Resource Window.

New to the Project Management feature of Studio is the ability to create virtual folders. This means you can emulate the directory structure of your server without having to create the same physical directory structure on your development machine.

RSW Software

www.rswsoftware.com

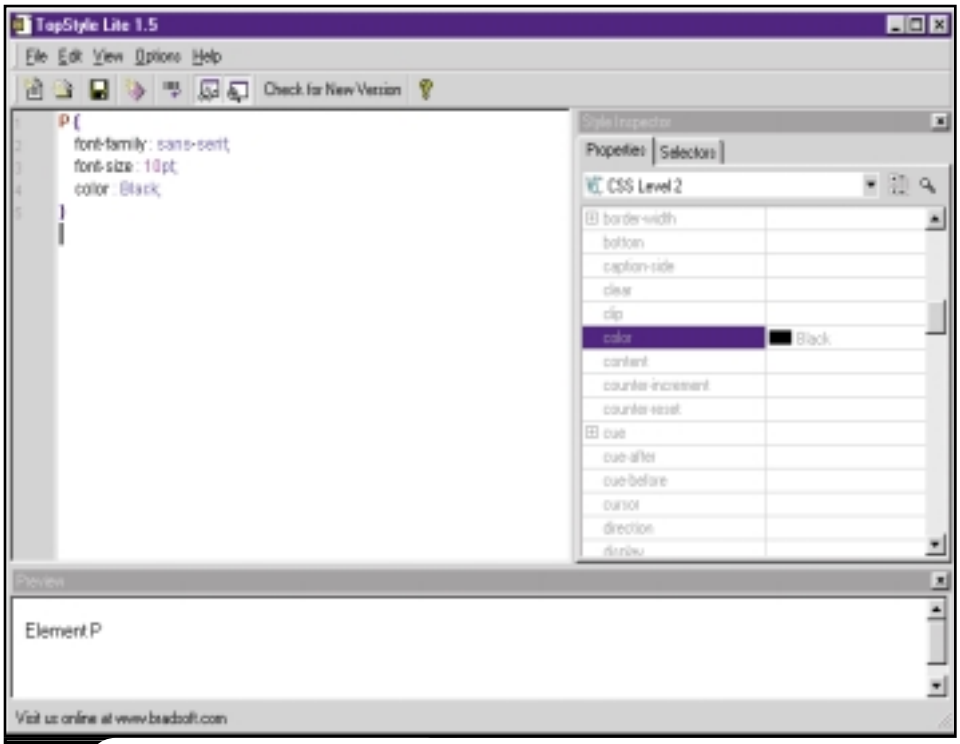


FIGURE 3: TopStyle Lite CSS editor

As for physical folders (those that are directly tied or mapped to a directory on your computer), they can be flagged as *auto-inclusive* or manual-inclusive. Auto-inclusive folders automatically include every file in that directory. *Manual-inclusive* folders require you to tell Studio which files to include in the project.

Just in case you did create projects in Studio 4.0, Studio 4.5 will read and convert those projects to the new format (an XML-defined WDDX format). Studio 4.0, however, won't read the new 4.5 project files. Both versions can coexist because the project file extensions are different. The old files end with ".apf" and the new files end with ".apj".

Scriptable Deployment

I haven't had enough time to really get into this, but scriptable deployment promises to be a great new feature. In the previous version of Studio you could upload an entire project by clicking on the Upload Project button. This feature was limited because it was an all-or-nothing proposition. You had to upload the entire project to the selected server.

With Studio 4.5, however, Allaire has given us much more flexibility by adding scriptable deployment, which gives you the ability to specify exactly how you want a project to be deployed. You can specify which folders will be deployed to which servers or deploy your project to multiple servers in one deployment process (great if you're running server clusters). You can create deployment settings for the entire project or for specific folders. You can set "Relative to the Parent Folder Location" or "Specific Deployment Location" options. You can even mark some folders with the "Do not upload" option if you're not ready for those files to be uploaded.

Studio has a rather sophisticated deployment wizard that'll walk you through the process and create a JavaScript or VBScript file (you decide which one) that you can modify to your liking. When you're ready to deploy, simply run the deployment script file as you would any other Web page.

Function Insight

One of the things that has always made Studio (and HomeSite) stand above other editors is the features that make it easier to remember HTML and CFML tag syntax and attributes. I've always been fond of the Tag Insight feature that pops up a drop-list of appropriate attributes or values for the tag you're currently typing. This release of Studio has added Function Insight. It works just like Tag Insight but recognizes that you're typing a CFML function and automatically displays the appropriate arguments for that function – a welcome feature for those of us who don't have room to keep all those arguments in our heads. Function Insight and Tag Insight can be enabled or disabled from the Settings dialog (F8).

TopStyle CSS Editor

If you haven't yet acquainted yourself with cascading stylesheets (CSS), let me recommend that you do (see Figure 3). CSS is gaining popularity – it's being widely supported by major browsers and strongly encouraged in the HTML 4.0 specification.

Studio 4.0 had a built-in CSS editor, but it was fairly limited. This release includes a new CSS editor called TopStyle Lite, an application written by Nick Bradbury (the original developer of HomeSite) that's a significant improvement over the old editor. One thing I like very much about TopStyle Lite is that it has its own online help system that completely explains cascading stylesheets, from a strong overview of how they work to specific usage. The editor itself is straightforward and easy to use, and has an accurate preview window to let you see how your styles will look.

Summary

Other tweaks and improvements show up throughout the application, though some places could stand improvement. Overall, Studio 4.5 is definitely worth the upgrade and in my opinion is still the best tool for developing ColdFusion applications. It sure beats the text editor and macros I used for way too many years before I started using Studio. Install the new version and see for yourself.

ABOUT THE AUTHOR

Bruce Van Horn is an Allaire - certified instructor and a senior Internet applications consultant for Project Control Solutions, Inc., one of Allaire's Premier Partners.

BVANHORN@PCS-LINK.COM

Eprise

www.eprise.com

When **Not** to Use ColdFusion

BY **BEN FORTA**

Yes, there really are times when ColdFusion should **not** be used

Yes, you read the title correctly, and no, I'm not losing it. I really do believe there are times when you should not use ColdFusion. And before crowds start gathering in the streets to burn my effigy, let me explain.

Understanding ColdFusion

ColdFusion, as we all know, is an application server, and a darn good one at that (whew, saved myself just in time). ColdFusion does lots of wonderful things for us. It reads and writes information in databases, it performs all sorts of conditional processing, it interacts with server file systems, it talks to HTTP, FTP, LDAP and mail servers, and much more.

But the fact that ColdFusion can do all those things for you doesn't mean it should, nor does it mean you should always use ColdFusion to solve all problems.

Let me put it another way. ColdFusion provides all the logic elements needed to write complex business rules, but applications dedicated to processing such rules can probably do the job far more efficiently. Similarly, ColdFusion provides tags and functions that can be used for directory and file manipulation, but compiled code written using low-level IO libraries will typically be far more effective at that same job. Another example: ColdFusion lets you loop through retrieved data to perform aggregate calculations, but any database will dramatically outperform ColdFusion at this same task.

Now before I get into trouble, let me state very clearly that this is *not* a flaw in ColdFusion. It is deliberate and by design. There is no application that does everything better than any other application. Application development always involves tradeoffs – you have to give up some things to gain others. That's true of every application, from operating systems and development tools to the applications that you yourself develop.

ColdFusion is designed to provide you with incredible functionality, and with almost no learning curve. It's a tough balance to maintain, but ColdFusion has proved remarkably capable at playing these dual roles. And in doing so, there is an understanding that there are some things ColdFusion will never do as well as other applications.

Again, this is *not* a flaw. Read on.

When Not to Use ColdFusion

What makes ColdFusion unique is that it was designed with the understanding that there are products and technologies out there that excel at specific tasks. Unlike other application development systems that lock you into a specific set of tools or options, ColdFusion was designed to be completely open and flexible.

Of course, ColdFusion's creators also realized that developers shouldn't have to write low-level C code each time they want to perform directory or file-level operations. Thus the core CFML language was enhanced to include tags that can do those things. But that doesn't mean that CFML is necessarily the best way to solve a problem. Don't confuse the convenient solution with the ideal one.

In many ways ColdFusion is middleware. It is less an application and more a facilitator, glue you use to bind all the bits that make up your application – commercial bits, bits written in ColdFusion and bits written in other languages. These bits are typically called *components*.

Using Components

There's another reason to use other products and technologies to extend ColdFusion, aside from being able to leverage the advantages and strengths of those products and technologies.

Logic implemented in ColdFusion can't be shared easily with other applications. Now, if your

SD 2000

www.sdexpo.com

entire organization ran only on ColdFusion, that wouldn't be a problem, but I've yet to come across any organization that can make that claim. Be it accounting systems, shipping tools, human resource management products, network security systems and who knows what else, your organization is likely made up of lots of different products. By moving logic or business rules, for example, into external components, they may be shared by different applications.

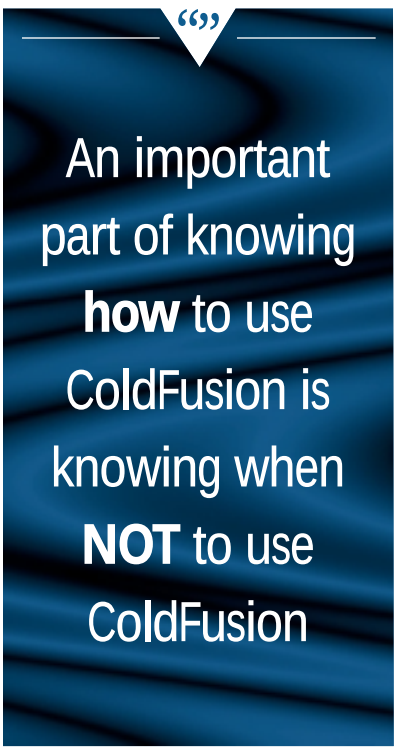
Let me give you an example. You have an e-commerce site (everybody does, right?). When items are purchased you run complex calculations to determine tax and shipping fees. Now you *could* write all that logic in ColdFusion, but if you did you'd end up re-creating that code for every other application that needed access to those calculations. A better solution would be to create components that perform the necessary calculations, and allow those components to be shared by all your applications, including ColdFusion.

Here's another example, perhaps a more tangible one. Your application accepts form fields from a user and then writes them to a database. To ensure database consistency you first massage the data – you convert state abbreviations to uppercase, capitalize country names and other proper nouns, and convert all phone numbers to a standardized format. And you do it all in ColdFusion using simple <CFIF> and <CFSET> statements and a bunch of CFML functions. Of course, data can be inserted into your database from other sources too (other applications, perhaps, including your DBMS's admin tools), so you'd need to find a way to write code to normalize data in every one of those applications too, a task that's not fun and possibly not always even doable. A better solution would be to write those conversion and manipulation rules at the database level (maybe as a trigger, or as a scheduled stored procedure); that way any applications would automatically benefit from those rules, including ColdFusion. In this example that database level trigger or stored procedure is the component, the external block of logic that applications can leverage.

Extending ColdFusion

Allaire has always believed that we developers should be able to pick the best-of-breed products when needed. And if there is no best-of-breed product, well, developers should be able to write their own. With ColdFusion they can. Ever since ColdFusion version 2 there have been ways to extend the product, and with every subsequent release the number of ways to do this has increased.

In ColdFusion 4.5 (the current release) the primary methods include the C/C++ API, COM, CORBA and server-side Java. (I'm not including custom tags here as those are used to write CFML code – they're obviously not used to write code that can't [or should not] be written in CFML.)



C/C++ CFXs

ColdFusion's C/C++ interface has been around since before most developers started using the product (unfortunately, in all that time the interface hasn't really changed much). The C/C++ interface exposes a subset of ColdFusion's functionality to code written in C or C++. Functions are provided to read and write queries as well as to perform basic variable manipulation.

Using this interface it's possible to call any C code from within ColdFusion. Indeed, it's primarily used to write interfaces to other C code libraries (this is the interface I used to create my CFX_Zip tag, for example).

The big advantage of this extension is performance. The big disadvantage is that a good working knowledge of C (or C++) is required, as is the ability to write multiuser safe code. As such, this interface is usually used only for very specific projects.

COM

COM – Component Object Model – is a software specification that's designed to allow applications to dynamically link (include) components at runtime. What this means is that developers can encapsulate functionality into clean published components and then allow other applications to interact with them.

COM objects are usually DLL or EXE files, and they can be written in almost any language imaginable, from C/C++ or Visual Basic to Java or Delphi. Once an object is written, it must be installed and registered on the computer on which it is to be used. Registering the COM object publishes it (and its interfaces) to other applications (like ColdFusion since version 3) so they can use it.

The big advantage of this extension is that COM objects are easy to write (Visual Basic, Visual C++ and other development tools even feature wizards to simplify COM object creation). The big disadvantage is that COM is essentially a Microsoft Windows-only solution. As such, COM is best suited for highly Windows-centric environments.

CORBA

CORBA (Common Object Request Broker Architecture) is an alternative specification for a distributed component object definition and use. The CORBA specification is defined by the Object Management Group (OMG), a group that includes some of the leading industry vendors. In a CORBA environment an object is an encapsulated entity whose services are accessed only through

well-defined interfaces. The location and implementation of each object is hidden from the client requesting the services.

CORBA is a sophisticated and powerful specification. It's also a very capable and very scalable one. And ColdFusion (as of version 4) can call CORBA components.

The big advantage of this extension is that CORBA is powerful, portable and scalable. The big disadvantage is that CORBA isn't cheap (and I'm not just talking about initial cost of software). As such, this ColdFusion extension is best suited for organizations that have an existing investment in CORBA technology. I honestly don't see organizations embracing CORBA just to use it via ColdFusion.

Server-Side Java

By now everyone has heard of Java. What started off as a language promising complete portability (incidentally, C started off with the same promise) has evolved into a powerful platform complete with all sorts of APIs and extensions. At the base of it all there is still a language – Java – and, obviously, doing any work in Java requires a good working knowledge of the language. (For the record, while Java is probably easier to learn than C or C++, it's considerably harder to learn than ColdFusion.)

Server-side Java is just that: Java that executes on the server. Unlike applets, which are Java applications that run on the client (the browser), server-side Java executes on the server, much like ColdFusion. Unlike applets, which are more often than not useless (thank goodness the days of 20-minute page downloads to be able to display a colored rotating bullet are over), server-side Java has quickly proved itself to be capable, scalable and highly portable. And ColdFusion 4.5 supports several different types of Java integration (extending the CFX_J interface made available in ColdFusion 4).

The big advantages of server-side Java are the ones I just listed – scalability and portability, as well performance. The big disadvantage is that you have to learn Java, but newer Java technologies are making that process much simpler. As such, extending ColdFusion with server-side Java is a good choice for just about anyone. There's no real downside to this one.

Summary

An important part of knowing how to use ColdFusion is knowing when *not* to use ColdFusion. There are lots of great products and technologies out there, and you should pick the best components for the job (or write them yourself if needed). The good news is that using the extensibility options listed above, ColdFusion can be made to work with just about anything out there.

As I alluded to before, my current bias is toward extending ColdFusion using server-side Java. One very important reason for this is that Allaire is a big believer in server-side Java (as demonstrated by the acquisitions of Live Software and Valto), and Java is going to play a very important role in the future of ColdFusion. If you use Java to extend your applications, you'll be right on track with what Allaire has in store for you. But I'm out of space, so more on that next month.

ABOUT THE AUTHOR

Ben Forta is Allaire Corporation's product evangelist for the ColdFusion product line. He is the author of the best-selling ColdFusion 4.0 Web Application Construction Kit *and its sequel,* Advanced ColdFusion 4.0 Development *(both published by Que), and he recently released* Sams Teach Yourself SQL in 10 Minutes.

BEN@FORTA.COM

Inteliant

www.inteliant.com

Multipage Dynamic Forms

Creating an unusual form wizard

BY LEE SURMA

Dynamic forms, which are found all over the Internet, are commonly found on e-commerce sites, business-to-business sites and intranets. For the beginners out there, HTML forms are Web pages containing data-entry areas, such as text, checkboxes and radio buttons, that require a submit button to activate an action page to post data, send e-mail, and so on.

A typical example of a dynamic form can be found on eBay. If you type *Lynyrd Skynyrd* in the search area and hit submit, you'll be presented with a list of every piece of Skynyrd information imaginable. The page that displays on eBay and many other sites contains a series of links/anchors containing the ID number of the item in the URL. Clicking on the link will bring you to the detail level for that item. This works well for many sites, but it's somewhat one-dimensional.

Let's Take It Up a Notch

Some sites are more functional when they present the user with a list of items, each with a checkbox, radio button or text area. This allows multiple items to be processed at once. It works well for e-commerce sites that don't have many items and business-to-business and intranet applications where the actions may be repetitive or predictable. Figure 1 shows a list of radio programs that allows our customers to view and edit the programs they air. I'm presenting a pared-down version of our real-world application to keep things simple. The customer is instructed to review the list and select the "edit" radio button for programs that need to be changed.

The list is dynamically generated from our database, and we can determine who the customer is by asking for a password or checking for a cookie. The dilemma with this kind of form is how to name and track all the different programs that can be called out of the database. In my example each listing has an "accept" or "change" radio box. We need to pass to the action page those programs the user has tagged with a change. Listing 1 displays the table for Figure 1. In most databases the one field that uniquely identifies a record is the primary key or, in my example, the RecordID. It's all pretty standard except for the radio boxes and a "hidden" text area in the code. The following lines of code are the key. Notice that the name for each radio box has an "a" appended to the RecordID. I'll explain this later. Also note the hidden field RecordIDList, which creates a list of all the RecordIDs to be looped through in the action page.

```
<Input Type="Radio" Name="a#RecordID#"
Value="Accept" checked>
<Input Type="Radio" Name="a#RecordID#"
Value="Edit">
<Input Type="Hidden" Name=RecordIDList
Value=#RecordID#>
```

Figure 2 shows the next page, which displays a nice intuitive interface that allows our customers to make changes to their data. Notice the hard-coded area that displays the original record followed by the prefilled text areas. Listing 2 displays the CFLoop necessary to make this happen. The hidden field from the first page provides the list to loop through. The RecordID with the "a" appended to it is evaluated to determine whether the "change" radio box was selected. The "a" is necessary because the evaluate command won't work on a straight numeric value.

A CFIF statement is then used to test for change. If one is indicated, a query on the RecordID is executed to pick up all the values to be displayed. This is somewhat processor intensive, but it beats trying to track and pass dozens of unique variables. In a

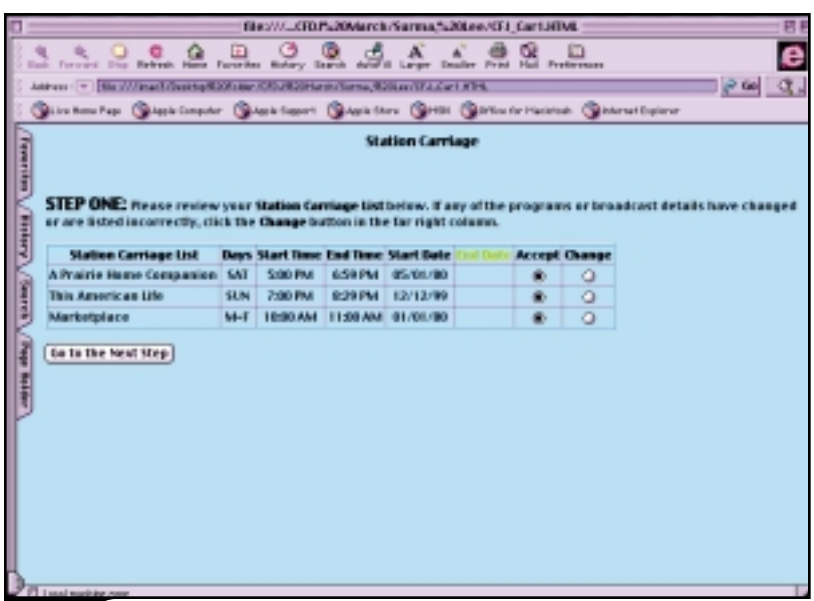


FIGURE 1 Interface allowing customers to view and edit the programs they air

Enteract

www.enteract.com

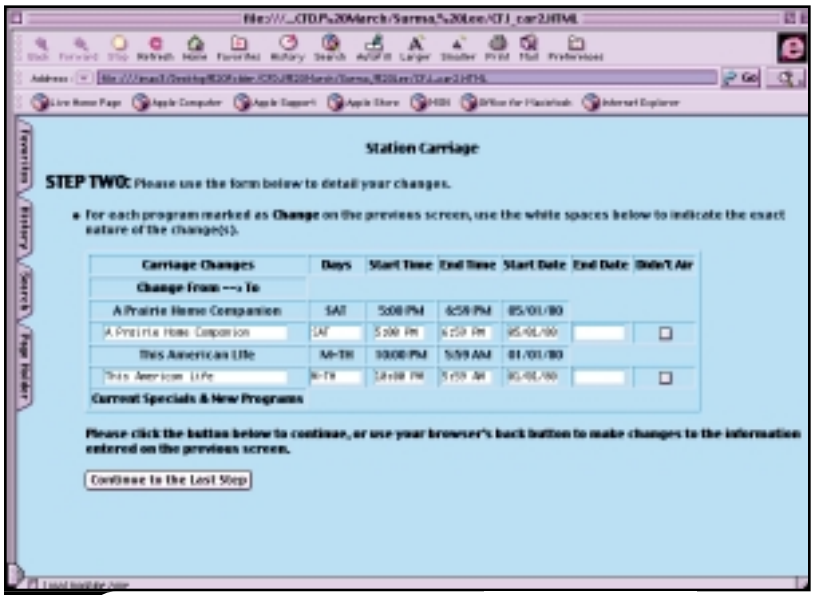


FIGURE 2: Interface allowing customers to amend their data

really high-volume site you may wish to painfully pass all the variables in lists or use some other method. The following code shows the list loop and the evaluate command.

```
<CFLOOP Index="ii" List="#Form.
 RecordIDList#">
<cfset id=#Evaluate("#ii#")#>
<cfif #Evaluate("a"&"#id#")# is "Edit">
</cfif> </cfloop>
```

Each data-entry area in Figure 2 also has the RecordID attached to its name, along with a letter scheme that allows me to identify the various pieces. I again include a hidden form element that puts all of the RecordIDs into a list.The following code snip illustrates this.

```
<Input Type="Text" Name="AA#RecordID#"
Value="#Program#" >
<Input Type="Text"  Name="AB#RecordID#"
Value="#Days#" >
<Input Type="Text"  Name="AC#RecordID#"
Value="#StartTime#" >
<INPUT TYPE="Hidden" VALUE=""
NAME="AG#RecordID#">
```

In this application I look for changes, then send an e-mail to our data-entry people. Creation of a dynamically generated e-mail that's neat and readable is a subject worthy of a future article. In brief, I use a series of <CFSet> statements with the CHR() command to build a variable to be sent out using <CFMail>.

In my last action page I again loop through the RecordIDs and use the following code snip to check for changes in the data and add them to the e-mail text.

```
<cfif #Original.Program# is not #Evaluate("AA"
& RecordID)#>
<cfset EmailText=EmailText&"Program Change:
"
&#Evaluate("AA" & RecordID)#&"     ">
<cfelse></cfif>
```

Conclusion

I hope I've given you enough information so the next time a client asks for an unusual form wizard you can say, "No problem." The key is passing the RecordID from form to form and creatively looping through the results.

ABOUT THE AUTHOR

Lee Surma is a senior manager of Internet technologies at Public Radio International, a Minneapolis-based distributor of public radio programming. He has five years of experience in various Web-database initiatives.

LSURMA@PRI.ORG

LISTING 1

```
 <FORM ACTION="Nextform.cfm" METHOD="Post">
<HTML>
 <BODY BGCOLOR="#CECDB4">
  <CENTER>
   <P><B><FONT SIZE="4">Station Carriage</FONT></B>
  </CENTER>
  <P> 

 <P><B><FONT SIZE="5">STEP ONE: </FONT></B>Please review your <B>Sta-
tion Carriage
 List </B>below. If any of the programs or broadcast details have changed or
are listed incorrectly,
 click the <B>Change</B> button in the far right column.


<P><TABLE BORDER="2" CELLSPACING="0" CELLPADING="3">
<TR>
<TH>Station Carriage List</TH>
<TH>Days</TH>
<TH>Start Time</TH>
<TH>End Time</TH>
<TH>Start Date</TH>
<TH><Font color="red">End Date</Font></TH>
<TH>Accept</TH>
<TH>Change</TH>
<TR><TH COLSPAN="8"></TH></TR>

<TR> <TD ALIGN=Left><font size="3">A Prairie Home Companion</font></TD>
<TD ALIGN=CENTER><font size="3">SAT</font></TD>
<TD ALIGN=CENTER><font size="3">5:00 PM</font></TD>
<TD ALIGN=CENTER><font size="3">6:59 PM</font></TD>
<TD ALIGN=CENTER><font size="3">05/01/80</font></TD>
<TD ALIGN=CENTER><font size="3"></font></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a1855" Value="Accept"
checked></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a1855" Value="Edit"></TD>
<Input Type="Hidden" Name=OngoingIDs Value=1855>
</TR><TR>


<TR> <TD ALIGN=Left><font size="3">This American Life</font></TD>
<TD ALIGN=CENTER><font size="3">SUN</font></TD>
<TD ALIGN=CENTER><font size="3">7:00 PM</font></TD>
<TD ALIGN=CENTER><font size="3">8:29 PM</font></TD>
<TD ALIGN=CENTER><font size="3">12/12/99</font></TD>
<TD ALIGN=CENTER><font size="3"></font></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a9083" Value="Accept"
checked></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a9083" Value="Edit"></TD>
<Input Type="Hidden" Name=OngoingIDs Value=9083>
</TR><TR>

<TR> <TD ALIGN=Left><font size="3">Marketplace</TD>
<TD ALIGN=CENTER><font size="3">M-F</font></TD>
<TD ALIGN=CENTER><font size="3">10:00 AM</font></TD>
<TD ALIGN=CENTER><font size="3">11:00 AM</font></TD>
<TD ALIGN=CENTER><font size="3">01/01/80</font></TD>
```

```
<TD ALIGN=CENTER><font size="3"></font></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a1854" Value="Accept"
checked></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a1854" Value="Edit"></TD>
<Input Type="Hidden" Name=OngoingIDs Value=1854>
</TR>

<TR> <TD ALIGN=Left><font size="3">Beyond Computers</TD>
<TD ALIGN=CENTER><font size="3">M-F</font></TD>
<TD ALIGN=CENTER><font size="3">10:00 PM</font></TD>
<TD ALIGN=CENTER><font size="3">11:00 PM</font></TD>
<TD ALIGN=CENTER><font size="3">01/01/80</font></TD>
<TD ALIGN=CENTER><font size="3"></font></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a1855" Value="Accept"
checked></TD>
<TD ALIGN=CENTER><Input Type="Radio" Name="a1855" Value="Edit"></TD>
<Input Type="Hidden" Name=OngoingIDs Value=1855>
</TR>
</TABLE></CENTER>

<Input Type="Hidden" Name="CL" value="KBPR-FM">
<P><div align="center"><INPUT TYPE="SUBMIT" VALUE="Go to the Next Step"
NAME="SaveNote"></div> </FORM>
</body>
</html>
```

LISTING 2

```
<FORM NAME="CFForm_1" ACTION="Post.cfm" METHOD=POST >
<HTML>
<BODY BGCOLOR="#CECDB4">
<P> 
<CENTER>
<P><B><FONT SIZE="4">Station Carriage</FONT></B>
</CENTER>

<P><B><FONT SIZE="5">STEP TWO: </FONT></B>Please use the form below
to detail
 your changes.
<ul>
 <li>For each program marked as <B>Change</B> on the previous screen,
  use the white spaces below to indicate the exact nature of the change(s).
 </li>

<P><TABLE BORDER="2" CELLSPACING="2" CELLPADING="2">
<TR>
    <TH>Carriage Changes</TH>
    <TH>Days</TH>
    <TH>Start Time</TH>
    <TH>End Time</TH>
    <TH>Start Date</TH>
    <TH>End Date</TH>
    <TH>Didn't Air</TH></TR><TR>
    <TD ALIGN="CENTER" COLSPAN="7"></TD>
   </TR><TR>
    <TD ALIGN="CENTER"><B>Change From --&gt; To</B></TD>
    <TD ALIGN="CENTER" COLSPAN="6"></TD></TR>
   <TR>
    <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
     <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
   </TR>
<TR>
    <TD ALIGN="CENTER">A Prairie Home Companion</TD>
    <TD ALIGN="CENTER">SAT</TD>
    <TD ALIGN="CENTER">5:00 PM</TD>
    <TD ALIGN="CENTER">6:59 PM</TD>
    <TD ALIGN="CENTER">05/01/80</TD>
    <TD ALIGN="CENTER"></TD>
     <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
   </TR>
         <TR>

    <TD ALIGN="CENTER"><Input Type="Text"  Name="AA1855" Value="A
Prairie Home Companion"  Size=30 Max Length=50></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AB1855"
Value="SAT"  Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AC1855" Value="5:00
PM"  Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AD1855" Value="6:59
PM"  Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AE1855"
Value="05/01/80"  Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AF1855" Value=""
Size=8 Max Length=15></TD>
         <INPUT TYPE="Hidden" VALUE="" NAME="AG1855">
    <TD ALIGN="CENTER"><INPUT TYPE="CHECKBOX" VALUE="X"
NAME="AG1855"></TD></TD>
    <TD ALIGN="CENTER"></TD>
   </TR>


<TR>
    <TD ALIGN="CENTER">This American Life</TD>
    <TD ALIGN="CENTER">Sun</TD>
    <TD ALIGN="CENTER">7:00 PM</TD>
    <TD ALIGN="CENTER">8:00 PM</TD>
    <TD ALIGN="CENTER">01/01/80</TD>
    <TD ALIGN="CENTER"></TD>
     <TD ALIGN="CENTER"></TD>
    <TD ALIGN="CENTER"></TD>
   </TR>
         <TR>

    <TD ALIGN="CENTER"><Input Type="Text"  Name="AA1854" Value="This
American Life"  Size=30 Max Length=50></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AB1854" Value="Sun"
Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AC1854" Value="7:00
PM"  Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AD1854" Value="8:00
PM"  Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AE1854"
Value="01/01/80"  Size=8 Max Length=15></TD>
    <TD ALIGN="CENTER"><Input Type="Text"  Name="AF1854" Value=""
Size=8 Max Length=15></TD>
         <INPUT TYPE="Hidden" VALUE="" NAME="AG1854">
    <TD ALIGN="CENTER"><INPUT TYPE="CHECKBOX" VALUE="X"
NAME="AG1854"></TD></TD>
    <TD ALIGN="CENTER"></TD>
   </TR>

<!--- <TR>
<TD ALIGN="CENTER"><B>Current Specials & New Programs</B></TD>
<TD ALIGN="CENTER" COLSPAN="6"></TD>
</TR> --->
<Input Type="Hidden" Name="Break" value="Break">



</TR></TABLE>
 <P><B>
  <Input Type="Hidden" Name="CL" value="KBPR-FM">
<P>Please click the button below to continue, or use your browser's back but-
ton
   to make changes to the information entered on the previous screen.
 <P><div align="center"><INPUT TYPE="SUBMIT" VALUE="Post Changes"
NAME="SaveNote"></div>


</FORM>
```

CODE LISTING

The code listing for this article can also be located at www.ColdFusionJournal.com

XML DEVCON 2000

www.xmldevcon2000.com

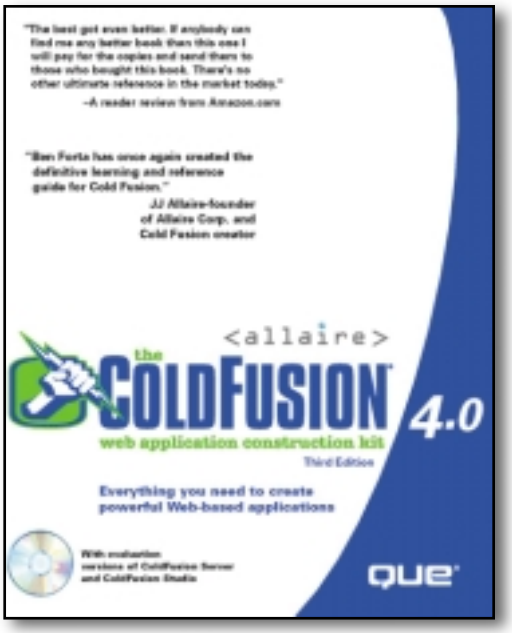
XML DEVCON 2000

www.xmldevcon2000.com

A Beginner's Guide to ColdFusion

ColdFusion Basics

Part 2



A primer in dynamic page development

FROM THE BOOK
BY **BEN FORTA**

Displaying Results in Tables

Most Web browsers support tables. The HTML <TABLE> tag enables you to display data in a two-dimensional grid. Tables are very useful for presenting lists in a clean columnar display.

As HTML tables are used so often to display query results in data-driven pages, and the <TABLE> syntax can be a little confusing at times, the makers of ColdFusion created a ColdFusion tag called <CFTABLE>. The <CFTABLE> tag is designed to conceal the details involved in creating HTML tables. All you have to do is tell ColdFusion what data to put in each column; ColdFusion generates the <TABLE> markup code for you.

The <CFTABLE> tag has another important advantage. It allows you to create tables that can be viewed by all browsers, even those that do not support HTML tables. To do this, ColdFusion renders the output in a nonproportional font and pads fields with spaces so that they line up in columns. While the resulting table might not look as good as a true HTML table, it is functional and is supported by all browsers. (*Note:* While <CFTABLE> can simplify the process of creating dynamic data-driven HTML tables, it does so at a price. <CFTABLE> does not provide much control over table formatting. For more detailed table layout, you need to create the HTML table manually using <CFOUTPUT> and <TABLE>.)

Creating Non-HTML Tables with CFTABLE

For an example of where <CFTABLE> can be used, look at the browser output shown in Figure 9. [*Ed. Note:* Figures 1–11 and Listings 1–10 can be found in Part 1 (***CFDJ***, Vol. 2, issue 2).] Notice how the phone extension is right next to the name, and in a different location on the screen, depending on how long the employee's name is. If the employees were listed in a table, the data could be presented in a cleaner and more organized fashion.

Listing 11 is based on Listing 4, but instead of using an unordered list and presenting each employee as a list item, the list is displayed in a table.

You use the tag <CFTABLE QUERY="Employees" COLHEADERS> to create the table. The <CFTABLE> tag is a special type of <CFOUTPUT>, and therefore requires that you specify a QUERY attribute, just like the one you'd provide to <CFOUTPUT>. <CFTABLE> is only used to display query results, and the QUERY attribute specifies which result set to process.

The COLHEADERS attribute is used to instruct ColdFusion to create optional column headers for each column in the table.

ColdFusion needs to know what columns you want to include in your table. Each column is specified using the <CFCOL> tag. You specify two columns here, one for the employee name and one for the phone extension.

The code for the phone extension column is <CFCOL HEADER="Extension" TEXT="Ext. #PhoneExtension#">. The HEADER attribute specifies the text to use in the column header. This

This article has been adapted from the second part of Chapter 11 of ColdFusion 4 Web Application Construction Kit *by Ben Forta. Published by permission of Macmillan Publishers Ltd. and the author. Part 1 appeared in the February issue of* **ColdFusion Developer's Journal.** *Adaptations of Chapters 12 and 13 will appear in forthcoming issues. The book can be purchased through Amazon.com or by clicking on www.forta.com/books.*

column has a header with the text Extension in it. The TEXT attribute is required; every <CFCOL> tag must have one. It tells ColdFusion what you want to display in this column. The TEXT attribute here contains the expression "Ext. #PhoneExtension#". As ColdFusion processes each row, it replaces the #PhoneExtension# field with the value of the retrieved PhoneExtension column.

The employee name column may look more complicated, but it really isn't at all. This is the source for the column:

```
<CFCOL HEADER="Employee" TEXT="<A
HREF=""empdtl1.cfm?EmployeeID=
 #EmployeeID#"">
[ic:ccc]#LastName#, #FirstName#</A>">.
```

Again, you first specify the text for the optional header in the HEADER attribute. The TEXT attribute contains the text to display, and because the name has to be a hyperlink, you must also specify the A HREF link tag.

In fact, the contents of the TEXT attribute are almost the same as the hyperlink tag used in Listing 7, with one notable exception. You'll notice that the link tag has double quotation marks around the URL, instead of the usual single set. You need the double quotation marks to tell ColdFusion to treat this as a quote, and not as the end of the TEXT attribute. If you had entered a single quotation mark, ColdFusion would have thought that the TEXT attribute ended right after the HREF=. Because it would not know what to do with the text after the quotation mark, ColdFusion would have reported a syntax error.

Now that you understand the code listing, run the template by typing http://yourserver.com/a2z/11/employ4.cfm.

As you can see in Figure 12, the employee names and phone extensions are now displayed in clearly labeled columns.

How is this table created without using the HTML <TABLE> tag? Look at the source code generated by ColdFusion to find out.

Select the View Source option in your browser. (In Netscape, select Page Source from the View menu; in Microsoft Internet Explorer, select Source from the View menu.)

As you can see in Figure 13, ColdFusion used the HTML <PRE> tag, which displays text exactly as it appears in the source code. Web browsers usually ignore whitespace characters such as spaces and line feeds. The <PRE> tag instructs the browser to maintain all spacing and line feeds, allowing ColdFusion to lay out the data exactly as it wants the browser to display it.

Creating HTML Tables with CFTABLE

Tables created with the HTML <TABLE> tag, of course, look much better, so ColdFusion also supports HTML tables. As you can see in Listing 12, to create HTML tables all you need to do is specify the HTMLTABLE attribute in the <CFTABLE> tag.

Figure 14 shows the same employee list screen rendered in an HTML table. Note that standard fonts are used when displaying data in an HTML table; the fixed font used when the <PRE> tag is specified is not used. Therefore, it is safe to use any other HTML formatting options in the CFCOL TEXT attribute if required. Therefore, if you wanted the name in bold, for example, you could have specified this:

```
TEXT="<A
HREF=""empdtl1.cfm?EmployeeID=#Employ-
eeID#"">
[ic:ccc]<B>#LastName#,
#FirstName#</B></A>"
```

ColdFusion would still have been able to display the table correctly. The <B> and </B> tags are HTML tags, not CFML tags; therefore, ColdFusion passes them through to the Web server to be sent to your Web browser.

To create this table, ColdFusion generated HTML table code. This source code, as displayed by the browser's view source function, is shown in Figure 15.

TIP

Viewing the source code generated by ColdFusion is very useful when debugging template problems. When you view the source, you are looking at the complete output as it was sent to your browser. If you ever need to ascertain why a Web page does not look like what you intended it to be, a good place to start is comparing your template with the source code it generated.

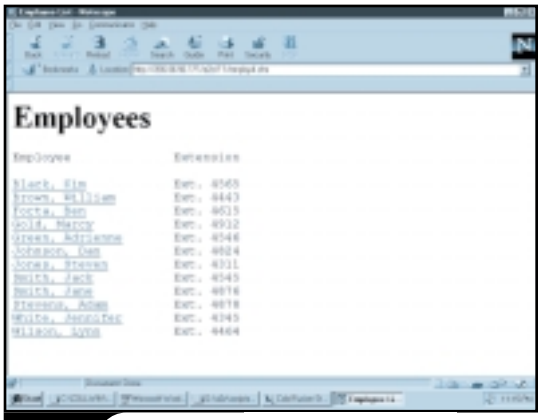


FIGURE 12: ColdFusion can generate non-HTML tables using the <CFTABLE> tag.

FIGURE 13: To see how ColdFusion interprets your template, view the generated markup language code with your browser's view source option.

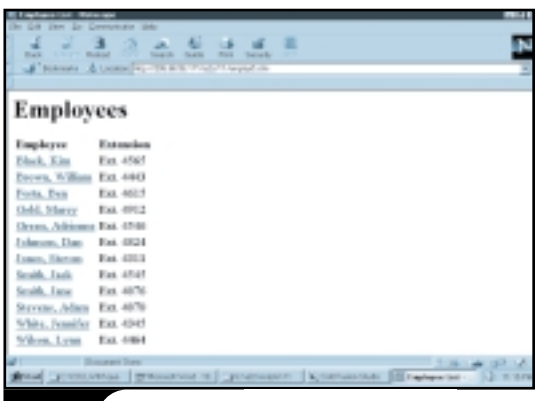


FIGURE 14: You can use the <CFTABLE> tag to create HTML tables.

Creating HTML Tables Manually

As good as the ColdFusion <CFTABLE> tag is, it is very limited. HTML tables support many advanced features, including table headers, cells that span multiple rows or columns, borders and border colors, background colors and images, and more. If you really want to control how your tables are displayed, you must resort to creating your tables manually.

Listing 13 demonstrates how to

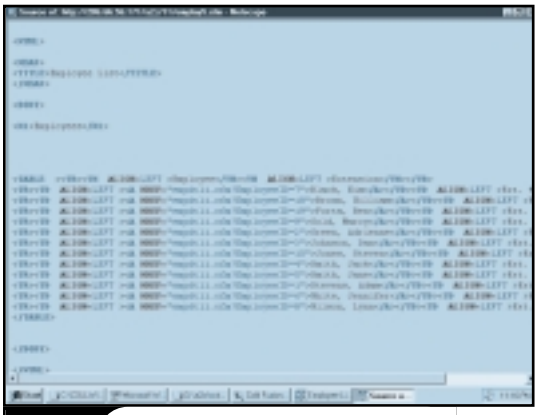


FIGURE 15: ColdFusion can generate all the required code to create HTML tables.

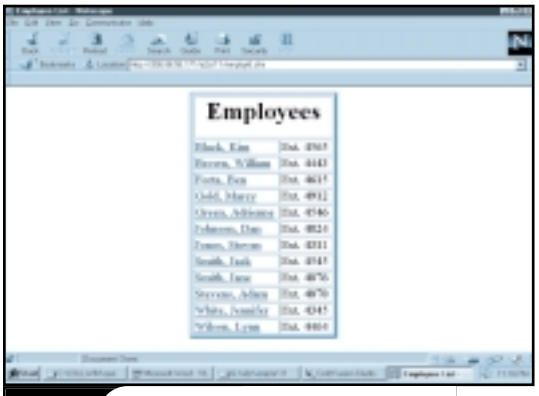


FIGURE 16: Creating tables manually allows a greater degree of control over table appearance.

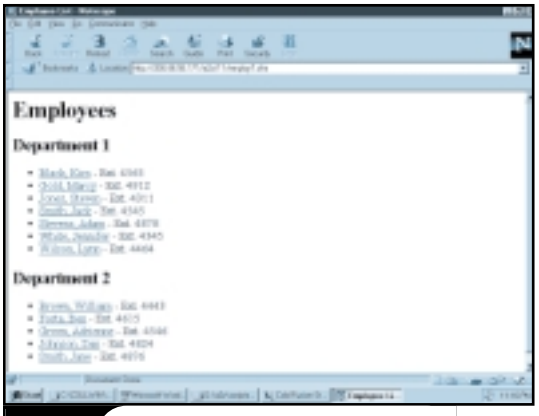


FIGURE 17: You can use the <CFOUTPUT> tag to group query results and display them accordingly.

manually create a bordered table for our employee list, and Figure 16 shows what the output looks like.

Look at the code in Listing 13. First we created the table with the <TABLE> tag and specified an optional border. HTML tables can have borders of varying thicknesses, and the BORDER attribute specifies the border to use; then we created a table title and placed it in a header cell (specified with the <TH> tag) that spans two columns.

Next comes <CFOUTOUT>. A new table row is created as each query row is output. This is why there is a complete table row (<TR>) and cells (<TD>) within the <CFOUTPUT> code block.

Finally, you close the table with </TABLE>.

As you can see, manually creating tables requires a little more effort and a better understanding of HTML tables, but the rewards are well worth it. (*Note:* HTML tables are a very useful way to format data, but there is a cost associated with using tables. In order for a browser to correctly display a table, it cannot display any part of that table until it has received the entire table from the Web server. This is because any row, even one near the end of the table, could affect the width of columns and how the table will be formatted. Therefore, if you display data in a table, the user will see no data at all until all the data is present. If you were to use another type of display, a list for example, the data would be displayed as it was received. The reality of it is that the page will likely take as long to fully load with or without tables. The downside of using tables is that it takes longer for any data to appear. This does not apply to tables created without the <TABLE> tag.)

Grouping Query Results

Before a new level of complexity is introduced, review how ColdFusion processes queries.

In ColdFusion data queries are created using the <CFQUERY> tag. <CFQUERY> performs a SQL operation and retrieves results if there are any. Results are stored temporarily by ColdFusion and remain around only for the duration of the processing of the template that contained the query.

The <CFOUTPUT> tag is used to output query results. <CFOUTPUT> takes a query name as an attribute and then loops through all the rows that were retrieved by the query. The code block between the <CFOUTPUT> and the </CFOUTPUT> is repeated once for every row retrieved.

All the examples created until now displayed results in a single list or a single table.

What would you do if you wanted to process the results in subsets? For example, suppose you wanted to list the employees by department. You could change the SQL statement in the <CFQUERY> to set the sort order to be department, and then perhaps by name within each department.

This would retrieve the data in the correct order, but how would you display it? If you used <CFOUTPUT> as we have until now, every row created by the <CFOUTPUT> block would have to be the same. If one had a department name, all would have to, because every row that is processed is processed with the same block of code.

How would you create the output shown in Figure 17?

The solution is to group the data results. Grouping allows you to have more than one <CFOUTPUT> loop. To understand how grouping works, look at the template in Listing 14.

The first change you made was adding the DepartmentID column to our SQL SELECT statement, and modifying the sort sequence with ORDER BY DepartmentID, LastName, FirstName. To group results by a column, that column must be the first in the sort sequence. Because we want to sort by DepartmentID, that column is now the first in the ORDER BY list.

The big change, however, is the <CFOUTPUT> block. There are now two of them, one nested inside the other. The outer <CFOUTPUT> tag has a new attribute: GROUP="DepartmentID".

Don't panic – this is explained right now.

A <CFOUTPUT> tag creates a loop that executes once for each row retrieved by a query. When you add the GROUP attribute, you instruct ColdFusion to execute the <CFOUPUT> block only when the group value (the field specified in the GROUP attribute) changes.

If you had seven employees all with the same DepartmentID, the GROUP <CFOUTPUT> block would be executed just once. In our list we have 12

TIP

Pay close attention to what code you place within and without the <CFOUTPUT> block. Misplacing a <TR> or </TD> could result in a badly formatted HTML table, and some browsers may opt to not even display that table.

LISTING 11: EMPLOY4.CFM – Using <CFTABLE> to Create Non-HTML Tables

```
<CFQUERY DATASOURCE="A2Z" NAME="Employees">
SELECT FirstName, LastName, PhoneExtension, EmployeeID
FROM Employees
ORDER BY LastName, FirstName
</CFQUERY>

<HTML>

<HEAD>
<TITLE>Employee List</TITLE>
</HEAD>

<BODY>

<H1>Employees</H1>

<CFTABLE QUERY="Employees" COLHEADERS>
<CFCOL
 HEADER="Employee"
 TEXT="<A HREF=""empdtl1.cfm?EmployeeID=#EmployeeID#"">#LastName#,
#FirstName#</A>"
>
<CFCOL
 HEADER="Extension"
 TEXT="Ext. #PhoneExtension#"
>
</CFTABLE>

</BODY>

</HTML>
```

LISTING 12: EMPLOY5.CFM – Creating HTML Tables with <CFTABLE>

```
<CFQUERY DATASOURCE="A2Z" NAME="Employees">
SELECT FirstName, LastName, PhoneExtension, EmployeeID
FROM Employees
ORDER BY LastName, FirstName
</CFQUERY>

<HTML>

<HEAD>
<TITLE>Employee List</TITLE>
</HEAD>

<BODY>

<H1>Employees</H1>

<CFTABLE QUERY="Employees" COLHEADERS HTMLTABLE>
<CFCOL
 HEADER="Employee"
 TEXT="<A HREF=""empdtl1.cfm?EmployeeID=#EmployeeID#"">#LastName#,
#FirstName#</A>"
>
<CFCOL
 HEADER="Extension"
 TEXT="Ext. #PhoneExtension#"
>
</CFTABLE>

</BODY>

</HTML>
```


employees who work in 2 departments. The outer <CFOUPUT> block only gets executed twice, once for each department. The first row processed has a DepartmentID of 1, so the <CFOUTPUT> block is executed. The next six rows processed also have a DepartmentID of 1, and so the <CFOUTPUT> block is not executed for them. The eighth row has a different DepartmentID, the value is 2, and so the <CFOUTPUT> block is executed. The next two rows also have a DepartmentID of 2, so no <CFOUTPUT> block is executed for them.

That's just the outer <CFOUTPUT> block. The inner block gets executed for every row, just like the <CFOUTPUT> blocks used earlier.

Now look at the output code in Listing 14. The outer <CFOUTPUT> creates a header for each new group and then starts a new unordered list. The inner <CFOUTPUT> populates that list until the group is completed; then the outer <CFOUTPUT> terminates the list and the process loops to the next DepartmentID.

The results are shown in Figure 17. (*Note:* There is no limit to the number of groupings you can create; you may nest as many CFOUTOUT tags as you need. The only restrictions are in using groups. First, every group must be part of the sort sequence used to retrieve the data. Second, the order that the columns appear in the ORDER BY clause must match the order of the groupings.)

Now you can see why the column you want to group on must be the first in the ORDER BY list. For grouping to work, all rows with the same value in the grouping column must be processed as a group. If the group is broken up, as could happen if you did not sort by the grouping column, ColdFusion would execute the outer block at the wrong times, and the resulting groups would be fragmented.

Table 1: ColdFusion Variable Types

Type	Description
Application	Application scope variables
Attributes	Attributes passed to custom tags
Caller	Caller scope variables
CGI	HTTP CGI variables
Client	Client variables
Cookie	HTTP client-side cookies
Form	HTML form fields
Request	Variable visible to all files that process a request
Server	Server scope variables
Session	Session variables
URL	Parameters passed to an URL
Variables	ColdFusion variables

Specifying Field Types

You have now used two different types of fields: CGI variables and URL parameters. ColdFusion supports several field types, as shown in Table 1, as well as fields that are database table columns retrieved with a <CFQUERY>.

In Parts 1 and 2 of this article you use two of these field types and <CFQUERY> results. If you read the entire book on which these articles are based you'll be using them all regularly.

Sooner or later you are going to run into a name collision. For example, you'll have a form field with the same name as a table column, or a variable with the same name as a URL parameter. How does ColdFusion know which one to use when this happens?

ColdFusion doesn't know, but it can make a best guess by using a known order of precedence; either that or you can specify it. You specify this by qualifying the field name with the field type.

Listing 15 is the same template you created in Listing 13 with one difference. The references to field Name are fully qualified as URL.name. This way, even if you had any other field called Name, ColdFusion would still know which field you were referring to.

Use your browser to view this template. The resulting display should be exactly the same as the one shown in Figure 3.

About the Author

Ben Forta is Allaire Corporation's product evangelist for the ColdFusion product line. In addition to authoring the book excerpted here, he is the author of its sequel, Advanced ColdFusion 4 Development. *Ben also recently released* Sams Teach Yourself SQL in 10 Minutes.

BEN@FORTA.COM

LISTING 13: EMPLOY6.CFM – Creating Tables Manually

```
<CFQUERY DATASOURCE="A2Z" NAME="Employees">
SELECT FirstName, LastName, PhoneExtension, EmployeeID
FROM Employees
ORDER BY LastName, FirstName
</CFQUERY>

<HTML>

<HEAD>
<TITLE>Employee List</TITLE>
</HEAD>

<BODY>

<CENTER>

<TABLE BORDER=5>

<TR>
 <TH COLSPAN=2>
  <H1>Employees</H1>
 </TH>
</TR>

<CFOUTPUT QUERY="Employees">

 <TR>
  <TD>
   <A HREF="empdtl1.cfm?EmployeeID=#EmployeeID#">#LastName#, #First-
```

```
Name#</A>
 </TD>
 <TD>
  Ext. #PhoneExtension#
 </TD>
</TR>
</CFOUTPUT>

</TABLE>

</CENTER>

</BODY>

</HTML>
```

LISTING 14: EMPLOY7.CFM – Employee List Grouped by Department

```
<CFQUERY DATASOURCE="A2Z" NAME="Employees">
SELECT DepartmentID, FirstName, LastName, PhoneExtension, EmployeeID
FROM Employees
ORDER BY DepartmentID, LastName, FirstName
</CFQUERY>

<HTML>

<HEAD>
<TITLE>Employee List</TITLE>
</HEAD>

<BODY>

<H1>Employees</H1>

<CFOUTPUT QUERY="Employees" GROUP="DepartmentID">
 <H2>Department #DepartmentID#</H2>
 <UL>

 <CFOUTPUT>
  <LI><A HREF="empdtl1.cfm?EmployeeID=#EmployeeID#">
[ic:ccc]#LastName#, #FirstName#</A> - Ext. #PhoneExtension#
 </CFOUTPUT>

 </UL>

</CFOUTPUT>

</BODY>

</HTML>
```

LISTING 15: HELLO4.CFM – Avoiding Variable Name Conflicts

```
<HTML>

<HEAD>
<TITLE>Hello!</TITLE>
</HEAD>

<BODY>

Hello,<BR>

<CFIF IsDefined("URL.name")>
 <CFOUTPUT>
 The name you entered is <B>#URL.name#</B>
 </CFOUTPUT>
<CFELSE>
 You did not pass a parameter called NAME
</CFIF>


</BODY>

</HTML>
```

CODE LISTING

The code listing for this article can also be located at www.ColdFusionJournal.com

Q: How many web developers does it take to manage content?

A: That's entirely up to you.

- 100% ColdFusion, 100% Browser-based
- Open, extensible and database driven architecture
- Full-featured and ready-to-go
- Empowering wizard interface for end-user publishing
- Template and object framework
- Granular access control, NT authentication support
- Content personalization/scheduling/expiration, approval workflow, versioning, intelligent caching
- NT/UNIX, Oracle/SQL Server/Access, Hosting support

www.paperthin.com

The answer to efficient content management.

1.800.940.3087

JAVACON 2000

www.javacon2000.com

JAVACON 2000

www.javacon2000.com

Helping Yourself

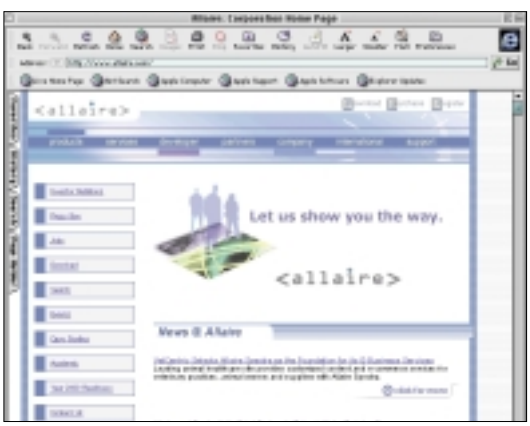
Resources for learning and for getting questions answered

BY CHARLES AREHART

In my work as an Allaire-certified instructor, students often ask questions, and while the courses address most of them, they sometimes have some really interesting ones that I know have been asked and answered before.

Often they'll be the frequently asked ones, but occasionally they have some really specific needs that may not be that common.

In such cases I try not to just answer the question but also point them to resources where they can find such answers themselves in the future.



In this month's **Journeyman** column I'd like to carry on that notion of "teaching a person to fish" by identifying various available resources – Web sites, e-mail services, print materials and so on where you can learn more about working with ColdFusion.

My special focus will be on resources where you can get questions answered. This may mean a searchable resource, or it may mean a place you can ask questions and engage in discussions with others in the CF community.

I won't just list them, though. I'll also discuss some common problems (and "beefs") that some folks have with them – and offer some solutions to help make the most effective use of such resources.

As I'll always intend in these columns, I think even experienced CF developers will find something useful. Since no coverage can ever be exhaustive, I welcome all feedback at carehart@figleaf.com.

The Supportive CF Community

Right from the get-go the CF community has been a strongly self-supporting one. Not only have the Allaire Forums (http://forums.allaire.com) been at the root of much of early Allaire CF support, they have also boosted the very growth of CF as a widely used platform. (For anyone not familiar with the forums, I'll explain them shortly.)

Like so many net-centric entities, Allaire's early and active involvement in this free "support" forum was part of what made people both willing and open to taking in this new platform. As most folks were at pretty much the same level of expertise, the forums' interactive question-and-answer style was perfect not only for answering specific questions but also for spreading knowledge about topics for those simply following the threads for information's sake.

Like so many resources of this kind, the forums have perhaps become victims of their own success. There are now so many discussion threads that searching the forums can take quite a long time. And that's only if the search doesn't "time out" in the first place – again, something I'll discuss later.

As the overall knowledge of the community grows, another problem is that the forums bring questions from new users at a never-ending rate. The loyal following of folks who used to contribute to the forums have now either moved on (perhaps they're self-sufficient) or have created alternative forms of finding questions and answers (such as the CF-Talk list, also discussed later).

Sadly, even these spin-off resources eventually meet a similar fate. The CF-Talk list generates on average 150 messages per day, making it nearly impossible to digest on a daily basis – for anyone but the most avid reader.

Don't Give Up Hope!

All this may lead some to conclude that such resources are simply no longer useful. But that would be a sad conclusion since both the forums and the CF-Talk list still provide tremendous value (as their continued use suggests). Even if you've given up on them as a daily source of info, I'm going to show how you can still make use of them on a need-to-know basis. In other words, I'll show you how you can search rather than simply browse them.

Yes, there are searchable archives for both these resources!

There are also ample alternative sites and resources that have stepped

in to fill the void or to create new subareas of support. I'll list some of the more interesting new resources out there and point out other non-Web resources that may be useful alternatives.

I'll conclude by offering a step-by-step list of places to search for answers in a pinch.

Resources for Help

Let's look at the five main types of resources:

- ***Web resources*** (including forums, e-zines, informational sites, metasites and online tutorials)
- ***E-mail resources*** (including list servers, newsletters and tip lists)
- ***Print resources*** (books and magazines)
- ***Training*** (computer-based and instructor-led)
- ***Online help***

Web Resources

Web-based resources include sites of all sorts. Some provide a place for interactive discussion. Others offer online magazines, provide detailed technical information, serve as lists of other sites or offer online tutorials. What follows are some of the more popular and useful of those sites.

Among the Web-based forums, where questions are asked and answered by many people, are:

- http://forums.allaire.com (Allaire Developer Forums)
- www.cfadvisor.com, www.defusion.com

These are useful resources when you have a question, as there may be people available to answer your question. You can also search for information gleaned from questions already asked by others. (We'll discuss searching these resources later.)

The next category of Web-based resources is online magazines (e-zines). There are a few in the CF world, including:

- www.cfadvisor.com, www.defusion.com,
- www.coldfusionjournal.com, www.cfmasters.com

These resources offer regularly scheduled content that changes weekly, monthly or sometimes even daily. You'd do well to read them regularly for their informative article-style content.

Still other Web sites offer general-interest CF information. These sites include those mentioned above but also:

- www.allaire.com (DevCenter, Knowledge Base, etc.)
- www.houseoffusion.com, www.fusionauthority.com, www.teamallaire.com
- www.cfdev.com, www.cfscripts.com, www.cfmcentral.com, www.cfspot.com
- www.cfprogrammers.com
- www.forta.com/cf/ (Ben Forta's site)

These sites have frequently updated (if not regularly scheduled) content.

Some sites are devoted to subareas of CF interest, including:

- www.hshelp.com, www.hsguide.com (for help with the HomeSite editor)
- www.fusebox.org, www.wddx.org

Sites devoted to providing a constantly updated list of the kinds of sites mentioned above – metasites – are useful directories for monitoring the ever-changing world of online CF resources. They include:

- www.houseoffusion.com (see their "bookmarks" section)
- www.cfmasters.com (their entire left navigation bar is a list of such links)
- www.ihs.gov/misc/links_gateway/sub_categories.cfm?Sub_Cat_ID=050102 (Indian Health Service site – worth a visit!)
- www.forta.com/cf/links/ (Ben Forta's list of links)
- www.webring.org (ColdFusion Web Ring)

Along the same lines, there's also a "metasearch" site devoted to ColdFusion. Andy Edmonds, of Fig Leaf Software's Atlanta office, put together a resource that indexes several of the sites we've mentioned here. Check out:

- http://psyberspace.net/coldfusion/search/

You can enter a word or phrase and the site will return a list of sites that mention it.

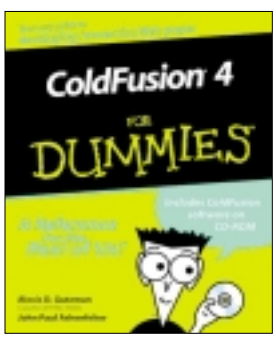
The final set of Web-based resources includes online tutorials. There are at least three:

- http://alive.allaire.com/ (Allaire Alive)
- www.figital.com (Spectra tutorials)
- www.cftutorials.com

Spotlight: The Allaire Developer Forums

The Allaire Developer Forums (http://forums.allaire.com) are a primary resource for asking and seeking answers to questions. The forums are a ColdFusion-built tool that provides a means for threaded conversations. You (or others) ask the questions, and enthusiastic fellow-developers – and occasionally Allaire staff – provide the answers (or debates).

One aspect of the forums is often misunderstood: they are a repository of much more than the couple of days' worth of material shown at the site's address. That front page is just a list of the most currently edited discussions (with two days' worth being the default). It is *not* the sole listing of messages – far more than just two days' worth of material is available for researching.



You may notice that some recent threads include discussions from quite a while ago, perhaps weeks, months or even years. This is because what gets listed are those threads that have had conversations in the last two days.

If you ask a question and it's not answered, the question doesn't "disappear." It merely scrolls off that front page. Anyone can still find the question by searching the topic. Allaire staff and Team Allaire members also have a mechanism to list still unanswered questions.

One great feature of the forums is that once you post (or reply to) a question, you will (by default) be flagged to receive by e-mail any other messages posted to that particular thread of conversation.

Note too that the forums are indeed searchable. At the top of the screen you'll see a small "search"

button. This allows you to search all the archives – three years' worth! – a treasure trove of questions answered. (Check the date of any answer you find, as old messages may sometimes be obviated or made obsolete by later changes in the CF product.)

Sometimes you'll find that your search "times out." This could be because the server is busy, so try again later. Or maybe your search is too broad. Consider refining it to limit which forum you intend to search (there are several to choose from), or to limit the number of days back you wish to search (the maximum you can limit is 90 days, but often the most relevant messages are the most recent ones).

Finally, you should be aware that there's a "login" feature for the forums. If you use that option, the system will remember your name and e-mail address so that when you create or reply to messages, you needn't identify yourself. A cookie will remember your previously entered contact information. Also, once you log in, you can then edit any messages you created while logged in.

E-Mail Resources

Moving from Web-based resources, which rely on your remembering to visit them if you intend to enjoy their benefits, there are also several e-mail resources. The benefit of these is that they are brought to you on a regular basis, in the form of mailing lists, newsletters or daily tip lists.

The mailing lists, or list servers, include several arranged by Michael Dinowitz at the houseoffusion.com site. Mailing lists are a resource by which folks participating in the list send e-mail messages to each other via a central server, which forwards the messages to all participants in the list. The lists include the very popular – and busy – CF-Talk list (more on this later), as well as Spectra-talk, wddx-talk, jrun-talk, cf-linux-talk and a fusebox-talk list. Directions for finding and joining these can be found at www.houseoffusion.com, under the "mailing lists" link on the left.

The e-mail newsletters include Allaire's developer newsletter (sign-up information at the Allaire Developer site), Hal Helms's newsletter (available at www.teamallaire.com/hal/index.cfm?fuseaction=newsletter) and a newsletter available at www.cfdev.com.

Yet another e-mail resource is the TipWorld daily CF tips, which can be requested at www.tipworld.com (along with daily tips available for sign-up on all sorts of topics).

Spotlight: CF-Talk Mailing List

The CF-Talk list, king of e-mail–based resources, is a list server that generates some 150 messages per day on average. If you're considering signing up for the list, bear this in mind. The quality of the discussion is worth the effort, however; it's a great place not only to ask and get answers to your questions, but also to simply "lurk" – reading the messages for their occasional gems of information (especially from frequent contributors like Dave Watts, Michael Dinowitz, Cameron Childress, Raymond Camden and others).

While the volume may perhaps be overwhelming, there's good news on a couple of fronts. When signing up, you can request that you receive just one e-mail a day in the form of a "digest" of the day's messages. It's a little hard replying to any one message when using the digest format, but for "lurkers," it's a great way to read the day's "news" easily.

Even better for managing the volume of information is the Web-based resource at egroups.com that serves as an archive to the CF-Talk list. Listed as a hyperlink from the CF-Talk area of the houseoffusion.com site, it can serve as a place to read the day's messages at your own pace (allowing you to skip days and not flood your mailbox when you simply can't keep up) and, more important, to search against the list archive.

Unfortunately, it seems that the search (and archive) only keeps a month or two of information from the list. Someone has promised to create a full archive of all 65,000+ messages. It'll be no easy feat, but let's keep our fingers crossed! Being able to search resources like the Allaire Forums and the CF-Talk lists provides the best of both worlds in finding answers that include rich debate and considered replies from many people.

Print Resources

For all the power and "coolness" of online resources such as Web sites and e-mail resources, sometimes it's easiest or more effective to learn from print resources, such as books and magazines. And the CF community has several such resources available.

As for print magazines, the best (and only one, currently) is the one you're reading. ***ColdFusion Developer's Journal*** (***CFDJ***) is published by **SYS-CON Publications** – the same folks who bring you ***Java Developer's Journal, PowerBuilder Developer's Journal,*** and others. ***CFDJ*** is well regarded for the depth and quality of its articles, and its broad range of popular and creative authors. Learn more at www.coldfusionjournal.com, or perhaps from the technical magazine section of your neighborhood bookstore.

As for books, the most popular has been and continues to be *The ColdFusion Web Application Construction Kit*, from Que, now in its third edition (1998). Edited by Allaire product evangelist Ben Forta, and written by him and several other authors, the book is considered by many the "bible" for CF development. It covers all the important topics useful to new

CF developers, but also database design, SQL and lots more.

For more experienced developers, the same group of people came out with *Advanced ColdFusion 4 Application Development* (1998). This book takes on several intermediate and advanced topics, and focuses especially on topics introduced by new features in CF 4.0.

Both books are widely considered to be must-have resources for every CF developer.

A more recent book, with similar breadth and depth, is *Mastering ColdFusion 4* (1999), from Sybex, by longtime Forum contributor Kristin Aileen Motlagh and Arman Danesh. A 4.5 update version of this book is due in April.

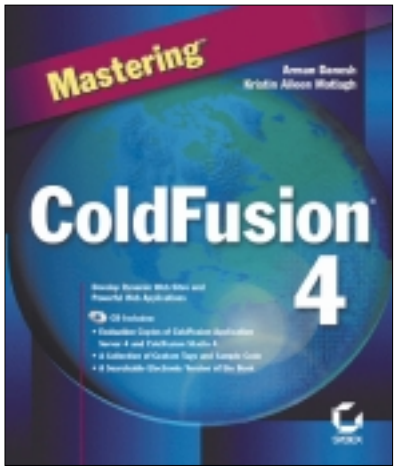
Another recent book is *ColdFusion 4 for Dummies* (published in February) by Alexis Gutzman and John Paul Ashenfelter, which is clearly less ambitious – and much lighter! – than the other two, but suitable for its intended audience.

Two other books worth mentioning, though not specifically on CF, are *Teach Yourself HomeSite in 24 Hours* (which covers the HomeSite editor and will be of value to Studio users as well) and *Teach Yourself SQL in 10 Minutes*. Both books are by Ben Forta, and the latter is an especially well-done book that describes the simplest and most important SQL needed for typical CF developers. Though not CF-specific, it takes an approach to describing SQL that is well suited to CF development.

Training Resources

Useful as all the resources discussed to this point may be, for some people there's no substitute for instructor-led training. This realm harbors quite a few choices. Allaire publishes a computer-based training CD, *Skill Building with ColdFusion*, that covers the most basic topics of CF development. It provides a multimedia approach – with instructor videos, interactive walkthroughs and more.

For live, instructor-led training, Allaire has a series of several topics that are taught regularly (often a few



times per month) in major cities throughout the country (and world). See the Allaire site, under Services, for a complete training schedule and descriptions

Online Help

For all the usefulness of the aforementioned resources, they lack the immediacy of a resource that you can turn to in an instant while at your computer. Fortunately, Allaire offers online help for both ColdFu-

sion Server and Studio. And the help for both is available if you have either product installed.

In Studio, help is available by choosing Help>Contents (or choosing the "Help" tab in the resource toolbar). This shows the available online references (which include CFML as well as HTML references, and lots more). You can also get immediate access to help on any tag (CF or HTML) or function by pressing F1 with the cursor on the intended tag or function (as of release 4.01 of Studio).

If you have ColdFusion Server on your workstation, you can access the help using Start>Programs>ColdFusion Server>ColdFusion Documentation. If you don't have the server on your workstation, but have it available on another server in your network, you may be able to access the help via http://<yourdomain>/cfdocs/dochome.htm. (Actually, if the server is one that can be reached publicly on the Internet, it may be best *not* to be able to access the help this way. An Allaire Security warning recommends that the help not be made available on publicly accessible servers.)

You can also download the online help from Allaire's site at their Developer section's "documentation" link.

Allaire's online help feature, both in Studio and Server, is a complete and faithful HTML rendering of the printed manuals that come with the product. Both versions offer a search feature that provides rapid searching for any topic in all the included manuals.

So...Where to Search?

You now know about a number of resources where you may find answers to your questions – or at least be able to learn more about a given topic. Which one would you turn to in a pinch? I'd recommend the following list to find the most specific to the most general answer:

- The Allaire online help/docs
- The Allaire Knowledge Base
- The Web-based CF-Talk Archive
- Andy Edmond's CF "metasearch" engine
- The Allaire Developer Forums

I mention the forums last only because the resource they constitute is so large that it can sometimes take a long time to search, and your search may even timeout before completion. Alternatively, you may get a set of results so large as to be overwhelming. I do think it's a valuable resource, but you may find the answer more quickly using one or more of the previously listed resources. And don't forget to look in the index of the CF books you may own!

The bottom line is to remember that you aren't alone. Any question you may have has almost always been asked and answered by others. The challenge is trying to find that answer. To that end I've described a wide variety of resources, each with different challenges and benefits. I hope you'll check them out – and may they prove useful to you!

ABOUT THE AUTHOR

Charles Arehart is an Allaire-certified instructor and developer working with Fig Leaf Software and is a frequent speaker at user groups throughout the country.

CAREHART@FIGLEAF.COM

www.WeAreCFHosting.com

CLEAR AND SIMPLE

Redundant DS3 Connection
99.9% Uptime Guarantee
30-Day Money Back Guarantee
50 MB Disk Space
2500 MB Data Transfer
20 Mailboxes with Unlimited Forwarding
Domain Name Registration
ColdFusion 4.5
Access to SSL Secure Server
Daily Tape Backup
Real Time Statistics

ColdFusion Hosting
$34.95/ month
Set Up: $40 (Free with yearly plan)

<allaire> Alliance Partner
Microsoft Certified Solution Provider

MS Access, FoxPro
MS SQL Server 7.0
Active Server Pages
CyberCash
Shopping Cart
Reseller Program

SITEHOSTING.NET
Toll Free: (877) NTHOSTING
(877) 684-6784
*InterNIC domain registration fee extra

Sys-Con Media

www.sys-con.com

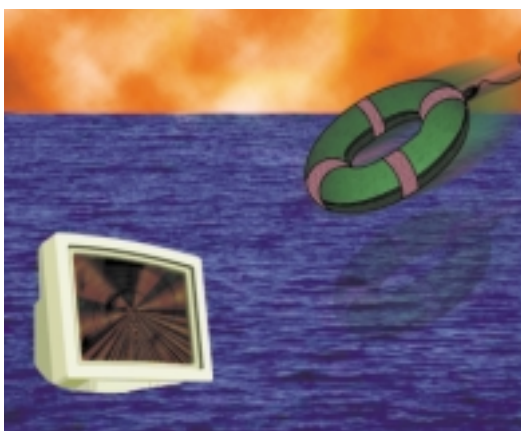
Session Management **The Hard Way**

If you're having trouble on a shared server, dumping session management can be a lifesaver

BY
MATT ROBERTSON

Session management is an essential tool for any Web application; however, one crazy coder on a shared server can kill the tool, your session variables – and maybe a sale to a customer. The problem isn't uncommon. Session, application and server-scope variables are stored in server RAM.

Sites on a shared server coexist with many others on the same box. With perhaps 200 ColdFusion apps written by coders of varying experience levels, it's not uncommon for ColdFusion Server to run out of memory or hit an error condition as a result of improper coding. When this happens, CF times out and restarts itself. This typically takes 30



to 90 seconds, and visitors may never notice...but your in-RAM variables get blown out as part of the restart. The effect on your app can range from nonexistent to disastrous.

Proper use of resources by everyone would solve the problem, but don't bank on that. A dedicated box is the real answer, but what if it's not an option? A better ISP helps a lot, but still doesn't address the inherent risk of the situation. This article shows a way to solve this problem by taking server RAM out of the picture. It isn't a solution for everyone, or for every need. However, this scheme is my first choice when designing a small-business, low- to medium-traffic site.

Create Your Own Token

We'll create a session ID token (variable) that doesn't use session management or client management (cookies might be shut off on the visitor's browser) or require database connectivity. Those other sites may blow up but yours will chug on. That leaves us in the Dark Ages – passing variables via links and hidden FORM commands. It's not pretty, but it works every time if you do it right.

We'll limit this article to the creation of the session ID. If you want to pass other variables independent of server RAM, you can do so via the URL (the leanest approach in terms of server resources), or perhaps use the session ID as a unique key in a database table where you store "session" variables in discrete records.

Let's get started. Listing 1 shows a file I call *topper.cfm*. For this to make sense the place to start is the end rather than the beginning. The code wrapped in the CFIF surrounding Lines 39–44 is actually the core of this show. Everything revolves around these six lines, which create a session ID token that consists of up to 23 digits.

#Dstamp# is the current date. *Note:* The NumberFormat applied to it gives us a "serialized" date: the number of days since January 1, 1900. This makes it easy to do date-spanning calculations. The next three lines set #Tstamp#: the current time to the second with all nonnumeric characters stripped away. The #IP# variable is the originating IP of the visitor with its nonnumeric dots removed.

#Dstamp#, #Tstamp# and #IP# are combined to create the #Visit# variable – the session ID token. We have our unique ID. Now what?

Now Plug It In

To start with, this routine has to be plugged into your application. Put this CFINCLUDE statement at the top of your template:

```
<CFINCLUDE TEMPLATE="topper.cfm">
```

It's important to note that there may be times when you don't want Topper.cfm to test the value (like during the middle of a cart checkout procedure), so just leave it off those templates and pass #Visit# properly through them. Here are some examples of passing #Visit# from page to page:

```
<A HREF="mypage.cfm?Visit=#Visit#">
<FORM ACTION="dostuff.cfm?Visit=
 #Visit#"
   METHOD="POST">
<INPUT TYPE="hidden" NAME="Visit"
 VALUE="#Visit#">
<CFLOCATION URL="page.cfm?Visit=#Visit#"
   ADDTOKEN="NO">
```

Now the Hard Part

We have our key for reliably maintaining state through most short-term server mishaps, and can pass it around the site. Expiring this beast – and making sure it doesn't get shared between visitors – is the tough part. *Now* we start from the beginning of topper.cfm.

First, #IDMake#, used to flag the refresh of #Visit# down on Line 38, is initialized. Next, we have a great big CFIF that begins by asking if #Visit# exists. If not, execution ends on Line 3 with #IDMake# getting set to "Y", and skips to Line 38 where #Visit# is created.

Where Are You From?

If, however, #Visit# does exist, the remainder of the routine continues at Line 5 where the next tests begin. Since at this point we've determined that #Visit# is defined, we'll look to see if this visitor is coming from the same Web site. To do this, we'll use ColdFusion's cgi-scope variable HTTP_REFERER. Lines 5–7 test HTTP_REFERER for the value in #MySite# in the referrer string. The case-insensitive search used on Line 6 is looking for my domain name ("mysecretbase"). Insert your own here. I don't use the ".com" as often as I park the ".net" on the same site, and want the test to work for both.

If not found, the value of #Rfr# will be zero and we'll proceed to a backup test to make sure this really is a first time visit. Why? If someone is visiting my site and types in the URL for the next page instead of clicking a link, this will be a directly accessed page, not a referred one, thus triggering a new #Visit# value. You could also cause this by moving the cursor to the end of the URL string in the address window and pressing ENTER. While this isn't the most likely user behavior, let's make it a lot less likely to cause trouble.

Lines 8–12 test the current originating IP value against the one stored in the currently defined #Visit# variable. Line 8 pulls the value and Line 9 trims any extra space. Lines 10–11 get the current IP and put it into the same format. Line 12 tests these two values for equality. If they don't match, #Visit# is flagged for a reset to take place on Line 38. Is this a fail-safe test? No. AOL rotates IPs during a visitor session, sharing several hundred IPs among many thousands of users. My AOL tests (my first joyous 500 hours were free!) generally worked, and I haven't seen any trouble at the sites where this system is implemented.

If our tests get past this point, topper.cfm figures this is the same surfer who was originally issued this ID, and we move to the test for expiration.

How Old Are You?

Topper.cfm now looks at the age of #Visit# to see if it should be refreshed. ColdFusion session variables expire after a period of inactivity. #Visit# expires after a preset time...period. This can't be fixed with a single-variable solution, and one that uses an inactivity test is a bit much to hit you with in the same article. Before I conclude, however, I'll describe one way to do it in case you want that feature.

When deciding on a timeout for a site, I look at their stats and use an interval longer than the maximum amount of time visitors spend there, but not too long. Make sure you make a similar evaluation and judgment call.

Topper.cfm now tests the age of #Visit# to see if it was created more than two hours in the past. Line 14 pulls the serialized date out of #Visit#, creates a new one from the current date in Line 15 and compares the two in Line 16. I like serialized dates. Their short, five-digit size isn't something to take for granted in a 23-digit string.

Lines 17–20 pull #Visit#'s #Tstamp# value back out (minus the seconds), and create a new current time value in the same format. These will be used in two of the three cases in the CFSWITCH statement spanning Lines 21–35. There are three

possibilities regarding the number of elapsed days since #Visit# was originally set.

First Case

It's the same day (#DDiff# is zero, Lines 22–26). CFIF asks whether the current time is greater than #Visit#'s time plus 200 units. Shouldn't that be 120 minutes? No. We can cheat here and use straight math coupled with a "military" time format.

Originally the time format in #Visit# was a six-digit number showing hours, minutes and seconds. For this test the seconds can be ignored and only #Visit#'s hour and minute values are pulled, giving a number like 0600 (you know: "We hit the beach at 0600," and all that). Then another hour-and-minute value is created from the current time (like 1710 or 5:10 p.m.).

Like the serialized date, this number can be evaluated in a simple fashion and the special needs of time-spanning calculations can be ignored. Using military format, a difference of 200 "units" equates to two hours in a relative calculation. Once again, if this test fails, #IDMake# is set to "Y" and execution continues on Line 38.

Second Case

It's the next day (#DDiff# is 1, Lines 27–31). We'll play the same military time unit game as above but calculate things differently given the difference in dates. This time Topper.cfm subtracts 2200 units from #Visit#'s time value, takes that number and compares it to the current time value. Huh?

With military time a value of 2200 represents 22 hours, so we're subtracting 22 hours from whatever time value is currently in place. Since we decided on a two-hour timeout, and 2400-200=2200 (yes, I know, the day ends with 2359, but the calc takes care of that), we set the value to subtract from #Visit#'s time value at 2200. A three-hour timeout would use 2100, and a one-hour timeout would use 2300. You may have to play with this concept a bit to get the hang of it, so here are three examples laid out in English.

- ***Example 1:*** Original Time: 11:50 p.m. Current Time: 1:30 a.m. 2350-2200=0150. Is 0150 less than 0130? No. Don't reset #IDMake#.

- ***Example 2:*** Original Time: 10:30 p.m. Current Time: 1:30 a.m. 2230-2200=0030. Is 0030 less than 0130? Yes. Reset #IDMake# to "Y".

- ***Example 3:*** Original Time: 3:30 p.m. Current Time: 3:30 p.m. (remember, we already know there is a difference of 1 in the day). 1530-2200= 0670. Is -0670 less than 1530? Yes. Reset #IDMake# to "Y".

Third Case

More than one day has elapsed (Lines 32–34). A CFDEFAULTCASE handles this. Any #DDiff# that's not zero or one falls through to here and once again #IDMake# flags the creation of a new #Visit# variable.

So there you have it. Used properly, this is a manually generated, fat-free session token you can rely on that can't be shared between users.

But Wait, There's More

Here's an idea on building in inactivity-based expiration. Use a second, four-digit time variable. Let's call it #Last#.

```
<A HREF="mypage.cfm?Visit=#Visit#&Last=#Last#">
```

Change it to match the current time on every page access. Test its age first, though. If it's older than your predetermined expiration value, set #IDMake# to "Y". Your homegrown session ID now behaves closer to ColdFusion's.

If you're having trouble on a shared server, dumping session management can be a lifesaver. Topper.cfm can be the centerpiece in this strategy, to a less extreme system involving storage of session variables in a database, or perhaps something else that better suits your needs.

ABOUT THE AUTHOR

Matt Robertson is vice president and chief designer at MSB Designs, Inc. (www.mysecretbase.com), a firm specializing in small-business CF-based sites.

MATT@MYSECRETBASE.COM

LISTING 1

```
1 <CFPARAM NAME="IDMake" DEFAULT="N">
2 <CFIF NOT isDefined("Visit")>
3  <CFSET IDMake = "Y">
4 <CFELSE>
5  <CFSET Rfr =
6 FINDNOCASE("mysecretbase",CGI.HTTP_REFERER,1)>
7  <CFIF Rfr EQ 0>
8   <CFSET VIP = MID(#Visit#,12,12)>
9   <CFSET VIP = #RTRIM(VIP)#>
10   <CFSET IPT =
11   REPLACE(CGI.REMOTE_HOST,".","","ALL")>
12   <CFIF VIP NEQ IPT><CFSET IDMake = "Y"></CFIF>
13  <CFELSE>
14   <CFSET VDate = MID(#Visit#,1,5)>
15   <CFSET DTest = "#NumberFormat(Now(),"_____")#">
16   <CFSET DDiff = DTest - VDate>
17   <CFSET VTime = MID(#Visit#,6,4)>
18   <CFSET HTest = MID(Now(),16,6)>
  <CFSET HTest = REPLACE(HTest,":","","ALL")>
19   <CFSET HTest = REPLACE(HTest," ","","ALL")>
20   <CFSWITCH EXPRESSION="#DDiff#">
21    <CFCASE VALUE="0">
22     <CFIF HTest GT (VTime + 200)>
23    <CFSET IDMake = "Y">
24     </CFIF>
25    </CFCASE>
26    <CFCASE VALUE="1">
27     <CFIF (VTime - 2200) LT (HTest)>
28      <CFSET IDMake = "Y">
29     </CFIF>
30    </CFCASE>
31    <CFDEFAULTCASE>
32     <CFSET IDMake = "Y">
33    </CFDEFAULTCASE>
34   </CFSWITCH>
35  </CFIF>
36 </CFIF>
37 <CFIF IDMake is "Y">
38  <CFSET DStamp = "#NumberFormat(Now(),"_____")#">
39  <CFSET TStamp = MID(Now(),16,9)>
40  <CFSET TStamp = REPLACE(TStamp,":","","ALL")>
41  <CFSET TStamp = REPLACE(TStamp," ","","ALL")>
42  <CFSET IP = 43   REPLACE(REMOTE_HOST,".","","ALL")>
44  <CFSET Visit = DStamp & TStamp & IP>
45 </CFIF>
```

CODE LISTING

The code listing for this article can also be located at www.ColdFusionJournal.com

CF USER GROUP LISTING

User Group	Contact	e-mail	Phone
NORTH AMERICA			
Alabama-Birmingham	Jeff Bryant	Jeff.Bryant@SouthTrust.com	(205) 667-5204
Alabama-Huntsville	Tasia Malakasis	tasia.t.malakasis@syntegra.com	(256) 971-9104
Arizona-Scottsdale	Sean Tierney	sean@pubcrawl.net	(480) 421.1270
California-Bay Area	Nathan Dintenfass	nathan@changemedia.com	(415) 793 9304
California-Sacramento	TJ Downes	tj@jel.net	(916) 447-5463
California-San Diego	John Morgan	Johnmorgan@Bluestarcorp.com	(619) 622-1201
California-Southern	Leon Chalnick	lchalnick@advantasolutions.com	(562) 491-0212
Canada-Edmonton	Jeff Acker	jeff.acker@stjus.ecs.edmonton.ab.ca	(800) 422-9772
Canada-Ottawa	Shaun St. Louis	shauns@newforce.ca	(613) 599-5786
Canada-Toronto	Kevin Towes	ktowes@PangaeaNewMedia.ca	(416) 922-1600
Canada-Vancouver B.C.	Cameron Siguenza	cameron@catouzer.com	(604) 662-7551 X303
Colorado -Northern	Robi Sen	robi@granularity.com	(970) 266-9125
Connecticut-Stamford	Barry Hyman	bah@wspthomes.com	(203) 221-0828
Florida-South	Tony Schreiber	dune@cfsf.org	(561) 999-1575
Florida-Orlando	April Fleming	cfug@itcenter.org	(407) 658-5060
Florida-Tallahassee	Benjamin Bloodworth	ben@electronet.net	(850) 222-0229
Florida-Tampa	Dale Mulert	dale@eng-tech.com	(888) 382-1900 x58
Georgia-Atlanta	Cameron Childress	cameronc@mcrae.com	(770) 460-7277
Georgia-Atlanta	Seth Land	Sland@figleaf.com	(404) 995-6030
Illinois-Chicago	Bob Burns	bob-burns@mediaone.net	(630) 961-6261
Indiana-Indianapolis	Lisa Sturgeon	ltsturgeon@yahoo.com	(317) 786-7846
Maryland -Baltimore	Steve Drucker	sdrucker@figleaf.com	(202) 797-5477
Maryland-Bethesda	Michael Smith	Michael@teratech.com	(301) 424-3903
Massachusetts-Boston	Steve Casco	steve@ateaze.com	(781) 275-6171
Minnesota-Minneapolis	Dan Chick	chickd@fastcart.com	(612) 561-1500
Missouri-St. Louis	Mike Stevens	mike@psiwebstudio.com	(618) 624-6694
Montana-Great Falls	Greg Snortland	snorty@unlimiteddata.com	(406) 452-8290
Nebraska-Omaha	Mark Kruger	dtnmarkk@earthlink.net	(402) 390-7173
Nevada-Las Vegas	Brian Thornton	brian@fiax.net	(702) 310-3600
New Jersey-Newark	Cindy Pomarlen	cindy_pomarlen@hotmail.com	(973) 540-9672
New York-Albany	Thomas McKeon	thomas.mckeon@cgiusa.com	(518) 434-0294 x464
New York-Binghampton	Robert Hanno	robert.hanno@lmco.com	(607) 741-3559
New York-NYC (advanced)	Michael Dinowitz	mdinowit@houseoffusion.com	(212) 647-8901
New York-NYC (basic)	Greg Narain	chairman@cfsig.org	(718) 768-3245
New York-Syracuse	Dave Santeramo	dsantera@zdnetmail.com	(607) 756-0393
North Carolina-Charlotte	Glynn Wallace	gwallace@carolina.rr.com	(828) 267-0013
North Carolina-Raleigh	Karen Andrews	kandrews@propoint.com	(919) 329-6434
Ohio-Columbus	Angelo McComis	Angelo@CompuServe.com	(614) 538-4539
Ohio-Cleveland	Jennifer Nagy	JNagy@newchanneltech.com	(330) 220-1558
Ohio-Mid Ohio Valley	Dave Hannum	hannum@ohio.edu	(740) 597-2524
Oklahoma-Tulsa	David Barnett	dfb@tdis.com	(918) 298-8949
Oregon-Portland	William Cupps	billc@dev-tech.com	(503) 294-4488
Oregon-Roseburg	Sharen Bland	sbland@douglasesd.k12.or.us	(541) 957-4822
Pennsylvania-Pittsburgh	Kyle Jenkins	kjenkins@steelcitywebs.com	(724) 846-2250
Pennsylvania-Philadelphia	Chris Swansen	cswansen@etechsolutions.com	(610) 458-2460
Rhode Island-Providence	Michael Rowan	mrowan@castlerea.com	(401) 792-7017
South Carolina -Greenville	Patrick S. Little	patricklittle@home.com	(864) 244-8965
Tennessee-Memphis	Michael Kubicki	mkubicki@univ-solutions.com	(901) 737-3406
Tennessee-Nashville	James Pecora	jpecora@radsoltn.com	(615) 269-4493
Texas-Austin	Dave Williams	david.williams@ci.austin.tx.us	(512) 499-2835
Texas-Dallas	Patrick Steil	pmsteil@imailbox.com	(940) 321-6162
Texas-San Antonio	Chris Montgomery	monty@astutia.com	(210) 490-3249
Virginia-Charlottesville	Steve Nelson	m@secretagents.com	(804) 293-2060
Virginia-Hampton Roads	Raymond Camden	jedimaster@creativeis.com	(757) 318-7042
Virginia-Northern	Steve Drucker	sdrucker@figleaf.com	(202) 797-5477
Washington D.C	Bret Peters	Bpeters@figleaf.com	(202) 797-5477
Washington -Seattle	Shannon Smith	president@cfseattle.org	(206) 343-9445
Wisconsin-Milwaukee	Steve Booth	steve.booth@wepco.com	(414) 221-4303
INTERNATIONAL			
Asia; South East	Roydean Osman	Roydean@gtemail.net	
Auckland	Amanda Irvine	amanda.irvine@delta.co.nz	09-309 9400
Australia-Barton	Peter Tilbrooks	peter@safetyweb.com.au	
Australia-Melbourne	Marc Dimmick	marc@tved.net.au	613-9670-2055
Australia-Sydney	Brent Pearson	b.pearson@morganbanks.com.au	
Belgium	Francois Lamotte	f.lamotte@switchon.be	
Europe (Central)	Sven Slazenger	slazenger@interlake.net	49-7542-1204
Ireland-Belfast	Daryl Fullerton	daryl.fullerton@cfug.ie	0044 1232 223224
Ireland-Dublin	Colm Brazel	cbweb@iol.ie	+353 (01) 4941625
Netherlands	Simon Slooten	slooten@prisma-it.nl	+31 (0)10 458 7148
Singapore	Chou Hao Chan	chiouhao@intracomm.com	
United Kingdom-London	Paul Underwood	paul@cfug.co.uk	44 (0) 171 589 4

Pending Fuseactions

Confessions of a Web developer

BY HAL HELMS

I had been working on the architecture of a large, complex e-commerce site when I ran into a situation. "It's always a situation," I muttered.

What would I do when users requested a service offered on the site, but they needed to register for it first? I could just write code that stops and advises them of the registration requirement, but this seemed to me awkward and annoying. Surely the site should be smarter than that.

I knew this was a common problem in many Web applications. I could state it as "A requires B" or I could use the shorthand notation my old professor of logic taught me: A—>B. Momentarily cheered by the thought that all that money had gone for something useful, I proceeded. What I wanted was a way of stopping the processing of A while the user performed B and then automatically sending the user back to A. Of course, things are seldom as simple as they first appear. I quickly ran into a wrinkle – A—>B—>C; this was complicated because C, a multistep process, is made up of C_1, C_2 and C_3.

I came up with a concept I call *pending fuseactions* (appropriate as the site was done in Fusebox). Pending fuseactions (PFAs) queue actions that need to take place in a specific order. They're like sequential log entries that the system executes in order. If, at any point, a certain action depends on a prior action, I place the requested fuseaction in the PFA log and handle the prerequisite action first.

So, in the case I outlined above, when the user comes to action A, I check to see if action B has already been performed. If not, I store action A as a PFA entry and go on to action B. At B I ask if action C has been performed. If not, action B joins action A in the PFA queue and action C is implemented. That sounded good, but I needed some way to remind the system to check for PFAs while retaining the continuity of C's multistep process. It occurred to me that I could create a variable that would act as a type of switch. If the switch was on (or *True* in computer lingo), the application would check for PFAs. I figured the normal state of the switch should be on. Then, when a multistep process began, I could flip the switch to False so the site would stop checking for PFAs. Once the multistep process completed, I could reset the switch.

I decided to implement the switch as a client variable. I thought of giving it an exotic name. My old professor loved to quote Cicero, so perhaps something Latinate like *terminus* would work. Finally I decided on the more mundane *checkPFAs*. If the system found checkPFAs set to True, it would check for the existence of another persistent variable that I'd create to hold queued fuseactions.

What if my user arrived at point A by clicking a link or submitting a form that sent certain information to that page – as URL or form variables? Argh, more complications! In that case, simply referring the user back to the fuseaction wouldn't be sufficient. I needed a way to capture the current information passed to the fuseaction and return it and the user to the appropriate page. From somewhere in the back of my DNA-based semipersistent storage, the words "request variable" appeared.

Introduced in ColdFusion version 4.1, "request" provides a structure that's global in scope and persists for the duration of a single page request. Variables that are scoped with the "request." prefix are accessible from the main page as are any custom tags, CFINCLUDEd files and files in a CFMODULE tag. I had this idea that if I could figure out how to get my hands on any variables passed into the file and save them in one handy packet, this would get me a lot closer to my goal.

The request variable is implemented in ColdFusion as a structure. Structures are familiar to all of us, though we may not realize it since we're so used to seeing and working with them. Anything that has a property name and a value for it is a good candidate for a structure (see Table 1).

Name	Value
firstName	Hal
lastName	Helms
Web site	www.teamallaire.com/hal

TABLE 1: Structures are made up of name/value pairs

In my case I needed variable names and variable values – perfect for the request structure.

I set about writing code that would add the individual form and URL variables to this structure (see Listing 1). This would ensure that whenever I needed to save a fuseaction as a PFA, I could place all the current form and/or URL variables that were passed to the fuseaction into the request structure.

Now I needed to save the fuseaction name and the request structure as variables that would last for a while. And since I could have multiple PFAs, I needed something that

could handle the concept of "rows" of data, from which I could easily peel off individual rows.

That sounded like a good definition of an array, so I opted for a two-dimensional array as a storage container. Still mired in the Latin theme, I toyed with the name *impendium* before naming it the simpler *PFAs*. While one-dimensional arrays have a single column to hold data, two-dimensional arrays have both columns and rows (see Table 2).

Each "row" of my PFAs array would represent a distinct PFA and contain two "cells" – one for the fuseaction name and the other for my request structure (since array "cells" can contain structures as well as other arrays and simple variables). When I needed a new PFA, I could put one together and add it to my array. Things were definitely looking up. I'd just create a client variable of an array type that would provide persistent storage.

Of course, client variables can't be array type variables (or any other complex variable type). Through the magic of WDDX, I could take a really complex item that contained structures, like an array, and convert it into a neat little string that could be saved as a client variable. I'd be able to casually name-drop to my fellow developers how I was "using XML to encode values for persistent storage." Sounded like a win-win.

This was it so far. I'd have my application code check for a prerequisite (see Listing 2). If the requisite task hadn't been performed, I'd put current fuseaction and variables into my PFAs array. If client.PFAs didn't exist, I'd create them.

I took this code along with the code that converted all form and URL variables into the request structure and saved them as a custom tag called *createPFA.cfm*. Now I had saved a PFA, added it to my client.PFAs array and was ready to move on to action B. Action B would implement the same scheme, sending the client on to action C. At action C I would set checkPFAs to False. When it was finished I turned checkPFAs back on.

All that remained was to write the code in the fusebox that would check for PFAs. I decided that if I added a simple client variable called *PFAnumber* I could save some CPU cycles by not deserializing the WDDX packet when it wasn't needed. Instead, I could check to see if PFAnumber was greater than zero. If it was, I had a PFA in the works; if not, I could skip the process. So I went back to my createPFA.cfm file and added the line:

```
<cfset client.PFAnumber = client.PFAnumber
+ 1>
```

Now my fusebox code (index.cfm) would check for the existence of client.checkPFAs and client.PFAnumber before running the CFWDDX code.

a_fuseaction_name	request_structure
a_fuseaction_name	request_structure
a_fuseaction_name	request_structure

TABLE 2: PFAs are contained in a 2D array of structures

```
<cfif client.checkPFAs>
 <cfif client.PFAnumber>
  <cf_getLastPFA>
 </cfif>
</cfif>
```

getLastPFA.cfm then deserializes the WDDX packet and stores the information into a request structure (again), making it available to the fusebox (see Listing 3).

If you're having a little trouble following the flow of this last bit of code, the chart in Figure 1 may help.

So there it was. I had run into the situation where A—>B—>C…and so forth. When I found that the required action hadn't been performed, I called a custom tag named *createPFA.cfm* that created a new structure to hold the variables passed to the fuse. I took this structure and saved it along with the fuseaction to the client variable PFAs. I also incremented another client variable called *PFAnumber* that set the number of pending fuseactions in the queue.

When the client variable checkPFAs was set to True and the client variable PFAnumber was greater than zero, I called on the custom tag, getLastPFA.cfm, to get the last pending fuseaction in the PFA queue and execute it with any variables that were originally passed to it. Finally, I decremented my PFAnumber variable.

Of course, all of this took a good deal longer to do than to write. I was just finishing the code when the phone rang. I looked out the window to see the sun had long since set. Where had the time gone? I picked up the phone. "Err…yes, dear, I know it's late and I promised to pick up dinner. Yes, dear. Yes. Well, what can I say but *tempus fugit*?" Somewhere, far away, my old professor smiled.

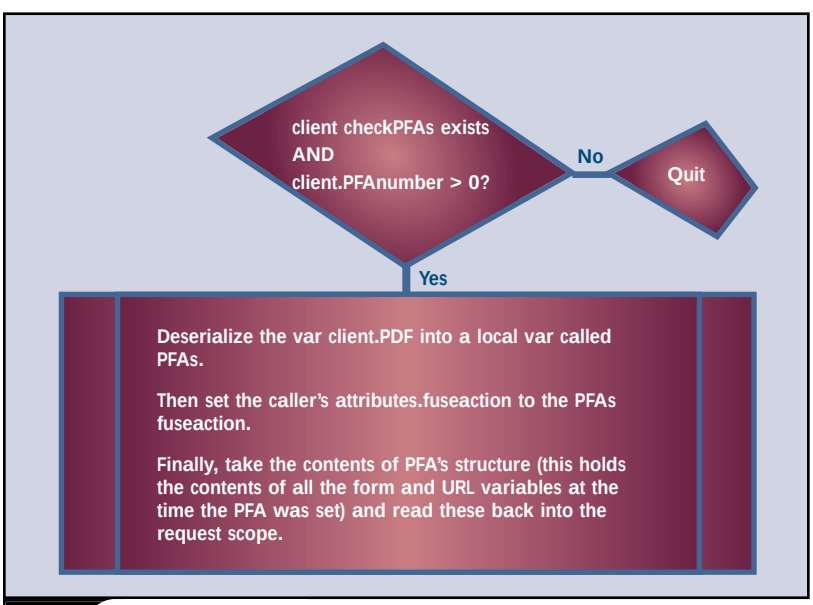


FIGURE 1: Executing a pending fuseaction

ABOUT THE AUTHOR

Hal Helms is a Team Allaire member living in Atlanta, Georgia. A frequent writer on ColdFusion and Fusebox, he also offers training and mentoring on these subjects.

HAL.HELMS@TEAMALLAIRE.COM

SHOP ONLINE AT JDJSTORE.COM FOR BEST PRICES OR CALL YOUR ORDER IN AT 1-888-303-JAVA

JDJStore.com

GUARANTEED BEST PRICES FOR ALL YOUR JAVA RELATED SOFTWARE NEEDS

Take a look at our specials this month!

EASTLAND DATA SYTEMS

Internet Shopping with Java Shopping Cart

…Described as the most progressive and interactive form of shopping on the web today…This Java Applet provides a complete user interface package for Internet Shopping Web Sites. Using Java technology we produce a drag-and-drop shopping user interface that is fun and easy to use, encouraging shoppers instead of frustrating them with confusing controls that are hard to follow. And the easier it is to shop, the more you sell.

. . . . . $294.99

Hybrid Shopping Cart

This Java Applet provides a complete user interface package for Internet Shopping Web Sites. A "Hybrid" is defined as an offspring of two varieties. A blending of the best features from our CGI and Java shopping products, we took the most powerful aspects of Java technology; real-time, on-screen updating and computational capabilities. And combined those with the most desirable features of our CGI shopping Cart, namely it's flexibility and compatibility with web designers with artistic talent.

. . . . . $294.99

CGI Shopping Cart

The Shopping Cart automates the Shopping Process to make shopping on your site intuitive, straight forward, and enjoyable! It's one of the most affordable Shopping Carts because it was designed for small businesses. Specifically for entrepreneurs who are testing the Internet waters, and can't or don't want to make large investments into bells and whistles for their site. But simply want to make shopping on their site easy for the customer.

. . . . . $294.99

Guaranteed Best Prices

JDJ Store Guarantees the Best Prices.
If you see any of our products listed anywhere at a lower price, we'll match that price and still bring you the same quality service.

Terms of offer:

- Offer good through October 31, 1999
- Only applicable to pricing on current versions of software
- August issue prices only
- Offer does not apply towards errors in competitors' printed prices
- Subject to same terms and conditions

Prices subject to change.
Not responsible for typographical errors.

Attention Java Vendors:
To include your product in JDJStore.com, please contact jackie@sys-con.com

ALLAIRE

ColdFusion 4.0

With ColdFusion 4.0, create Web applications for self-service HR solutions, online stores, interactive publishing and much more. It's the integrated development environment that has all the visual tools you need to create Web applications quickly and easily. From simple to sophisticated ColdFusion gives you the power to deliver the Web solutions you need – faster, and at a lower cost.

ColdFusion Studio 4.0 . . . . . $354.99
SkillBuilding with ColdFusion Interactive Training CD $284.99

ALLAIRE

HomeSite 4.0

HomeSite is the award-winning HTML editing tool that lets you build great Web sites in less time, while maintaining Pure HTML. Unlike WYSIWYG authoring tools, HomeSite gives you precise layout control, total design flexibility and full access to the latest Web technologies, such as DHTML, SMIL, Cascading Style Sheets and JavaScript. HomeSite 4.0 is the only HTML editor featuring a visual development environment that preserves code integrity.

HomeSite 4.0 . . . . . $87.99

ALLAIRE

JRun - Live Software

JRun is the industry-leading tool for deploying server-side Java. JRun is an easy-to-use Web server "plugin" that allows you to deploy Java Servlets and JavaServer Pages. Servlets form the foundation for sophisticated server-side application development. Java servlets are platform independent, easy to develop, fast to deploy, and cost-effective to maintain.

JRun $558.99

INSTALLSHIELD

InstallShield Java Edition 2.5

InstallShield Java Edition 2.5 is the powerful tool developers require to produce bulletproof InstallShield installations with Java versatility. You can target your application for multiple systems with cross-platform distribution. And InstallShield Java Edition 2.5 offers the key features and functionality designed to let developers go further in distribution and deployment.

InstallShield Java Edition $474.99

PROTOVIEW

Java Enterprise Editions

The Java Enterprise Editions offer you a choice between comprehensive packages of award-winning AWT or JFC components along with enterprise-level support and subscription service. Powerful, extendable, lightweight components built on the foundation of JFC, the ProtoView JFCSuite contains JFCDataCalendar, JFCDataExplorer and JFCDataInput. The JSuite (AWT) includes the DataTableJ grid component with JDBC and Visual Café database support. Also includes TreeViewJ, CalendarJ, TabJ and WinJ.

JSuite Enterprise Edition $818.99
JFC Enterprise Edition $818.95
JSuite $328.99
JFCSuite $408.99

KL GROUP

JProbe Suite

JProbe Profiler is the most powerful tool available for finding and eliminating performance bottlenecks in your Java code. JProbe Coverage makes it easy to locate individual lines of untested code and reports exactly how much of your Java code has been tested. JProbe Threadalyzer lets you pinpoint the cause of stalls and and deadlocks in your Java applications and makes it easy to predict race conditions that can corrupt application data.

JProbe Profiler w/ Standard Support (inc. JProbe Memory Debugger) . . $464.99
JProbe Coverage w/ Standard Support $464.99
JProbe Threadalyzer w/ Standard Support $464.99
JProbe Suite w/ Standard Support $934.99

GALILEO DEVELOPMENT SYSTEMS

Intr@Vision Foundation

Intr@Vision Foundation helps bring ColdFusion development to the next level. It provides an out-of-the-box application security architecture for handling your most complex intranet and extranet needs. Instead of spending 30% of your development time adding security to every application you build, it gives you a proven solution with a single line of code. Intr@Vision Foundation allows your developers to focus on building business solutions, not infrastructure.

Intr@Vision Foundation $3499.99

ABLECOMMERCE

AbleCommerce 2.9

Store Building Software
AbleCommerce's award winning electronic commerce software features complete browser based site building and administration for merchants with limited or no HTML skill up to Web Designers and programmers. AbleCommerce gives web entrepreneurs the ability to create a full-featured, customizable, commerce web site in just a few hours.

AbleCommerce 2.9 Store Building Software $795.00

LISTING 1

```
<cfif isdefined("cgi.query_string")>
 <cfloop
  list="#cgi.query_string#"
  delimiters="&"
  index="aNameValuePair">
  <cfset temp = SetVariable(Evaluate(DE('request.' &
GetToken(aNameValuePair, 1, "="))), GetToken(aNameValuePair, 2, "="))>
 </cfloop>
</cfif>

<cfif isdefined("form.fieldnames")>
 <cfloop
  list="#form.fieldnames#"
  index="aFormVar">
  <cfset temp = SetVariable ("request.#aFormVar#", Evaluate('form.' &
aFormVar))>
 </cfloop>
</cfif>
```

LISTING 2

```
<!--- I'm checking to see if the prerequisite is met…--->
<cfif NOT IsDefined("cookie.currentUser")>

<!--- …and if not, I'm going to see if I already have a client var called PFAs.
If not, I'll just make one. --->
 <cfif NOT IsDefined("client.PFAs")>
  <cfset PFAs=ArrayNew(2)>
 <cfelse>
  <cfwddx
   action="WDDX2CFML"
   input="#client.PFAs#"
   output="PFAs">
 </cfif>

 <cfset temp = ArrayAppend(PFAs[1], "#attributes.fuseaction#")>

 <cfset temp = ArrayAppend(PFAs[2], "#request#")>

 <cfwddx
  action="CFML2WDDX"
  input="#PFAs#"
  output="client.PFAs">
</cfif>
```

LISTING 3

```
<cfwddx
 action="WDDX2CFML"
 input="#client.PFAs#"
 output="PFAs">

<cfset lastPFA = ArrayLen(PFAs[1])>
 <cfset caller.attributes.fuseaction = "#PFAs[1][lastPFA]#">
 <cfset pendingRequestVars = "#PFAs[2][lastPFA]#">

 <cfloop collection="#pendingRequestVars#" item="aVar">
  <cfset SetVariable("request.#aVar#", "#Evaluate('pendingRequestVars.'
& aVar)#")>
 </cfloop>

 Now, I need to decrement the client variable "PFAnumber".
<cfset client.PFAnumber = client.PFAnumber – 1>
```

CODE LISTING

The code listing for this article can also be located at www.ColdFusionJournal.com

Virtualscape

www.virtualscape.com

CFDJnews

On-Line Data Solutions Releases iMS/FusionMail 1.1

(*Centereach, NY*) – On-Line Data Solutions, Inc., announces inFusion Mail Server version 1.1 – a free upgrade from version 1.0. Building on the iMS 1.0 platform, iMS 1.1 now includes flexible outgoing mail priority, MX caching, specific IP binding, ESMTP support, RCPT cache and iMS Test Suite.
www.coolfusion.com/iMS.htm

Allaire and Productivity Point Partner on Training

(*Cambridge, MA*) – Allaire Corporation and Productivity Point International have announced a distribution agreement to offer Allaire-authorized training on ColdFusion and HomeSite. Under the terms of this agreement, Productivity Point will provide training on Allaire products in more than 100 cities across North America.

Training is being rolled out between January and April. Allaire will continue to offer training directly as well as through the existing network of Allaire-authorized training centers. www.allaire.com.

Sitech Launches SiteMaker Development 2.0

(*Birmingham, AL*) – Sitech Software has announced the availability of SiteMaker Development 2.0, a Web development and management tool designed specifically for professional Web developers.

Key features and benefits include open source architecture, centralized site management and preprogrammed modules.
www.sitechsoftware.com

Catouzer Launches Synergy 2.0

(*Vancouver, BC*) – Catouzer has released Synergy 2.0, the latest version of their flagship product. Some of the new features developed for 2.0 include support of LDAP and NT Domain directories, which allows users to be registered into the Synergy system using LDAP or NT usernames and passwords, enhanced reporting features that provide statistics on the usage patterns within Synergy, and improved role and group management features.
www.catouzer.com

eye media Eyes Virtual Auctioneer Patient

(*Dallas, TX*) – eye media, inc., recently filed with the U.S. Patent & Trademark Office to patent certain aspects of its Virtual Auctioneer product, which offers a packaged solution to support a variety of e-commerce transactions, including Internet auctions, exchanges and online stores.

Virtual Auctioneer has been licensed to a broad range of companies for use in such diverse net market auctions as the purchase and sale of oil and gas properties, retail products, online music, overstocked inventories, and other vertical marketing and B2B activities.
www.eyemedia.net.

Isite Design
www.isitedesign.com

BiznizWeb
www.webpublishingtools.com

Infoboard
www.infoboard.com

Career Opportunities

ADVERTISING INDEX

ADVERTISER	URL	PH	PG
ABLE SOLUTIONS	WWW.ABLECOMMERCE.COM	360.253.4142	2
ADHOST	WWW.ADHOST.COM	888 ADHOST-1	31
BIZNIZ WEB	WWW.WEBPUBLISHINGTOOLS.COM	281.367.4016	50,52
CAREER OPPORTUNITIES		800.582.3089	51
CATOUZER	WWW.CATOUZER.COM	604.662.7551	55
COMPUTERJOBS.COM	WWW.COMPUTERJOBS.COM		3
DIGITAL NATION	WWW.DEDICATEDSERVER.COM	703.642.2800	4
EKTRON	WWW.EKTRON.COM	603.594.0249	9
ENHANCED TECHNOLOGIES	WWW.ENHTECH.COM	800.368.3249	39
ENTERACT	WWW.ENTERACT.COM	312.955.3000	23
EPRISE	WWW.EPRISE.COM	800.274.2814	17
INFOBOARD	WWW.INFOBOARD.COM	800.514.2297	50
INTELIANT	WWW.INTELIANT.COM	800.815.5541	21
INTERMEDIA.NET	WWW.INTERMEDIA.NET	650.424.9935	56
ISITE DESIGN	WWW.ISITEDESIGN.COM	888.269.9103	50
JAVACON 2000	WWW.JAVACON2000.COM		34-35
ON-LINE DATA SOLUTIONS	WWW.COOLFUSION.COM	516.737.4668	39
PAPERTHIN INC.	WWW.PAPERTHIN.COM	800.940-3087	33
RSW SOFTWARE	WWW.RSWSOFTWARE.COM	508.435.8000	15
SAISOFT	WWW.SAISOFTONLINE.COM	860.793.6681	52
SD 2000	WWW.SDEXPO.COM		19
SITEHOSTING.NET	WWW.SITEHOSTING.NET	888.463.6168	40
VIRTUALSCAPE	WWW.VIRTUALSCAPE.COM	212.460.8406	49
WATCHFIRE	WWW.WATCHFIRE.COM	613.599.3888	11
WINMILL SOFTWARE	WWW.WINMILL.COM	888.711.MILL	43
XMLDEVCON 2000	WWW.XMLDEVCON2000.COM		26-27

SaiSoftonline

Your E-Business Outsourcing Partner

Our clients speak for us:

"SaiSoft exceeded all the expectations of all our outsourcing needs"
Alex Sheshunoff, President, e-thepeople.com

"Key Player in our revolutionary product development"
Greg Imbruce, CEO, buyroad.com

<allaire> Alliance Partner

Microsoft Certified Solution Provider

ColdFusion
Allaire Spectra
JRun

- Content Mgmt.
- B2B Commerce
- XML/WDDX
- Secure Server Integration
- Credit Card Processing
- Store Front Builders
- Allaire Forums
- E-mail Mgmt.
- Instant Messengers/Chat
- Search Engines
- Oracle/SQLServer Backends

860.793.6681
www.saisoftonline.com

Able Solutions

Enter the realm of browsable store building and administration – from your browser. Build "your_site.com" with secure Merchant Credit Card Processing. Maintain inventory, add discounts and specials to keep your customers coming back. Increase sales with cross selling and membership pricing.

11700 NE 95th Street, Suite 100, Vancouver, WA
www.ablecommerce.com • 360 253-4142

Adhost Internet Advertising

Adhost provides complete web hosting solutions for over twelve hundred business clients. Small firms to multi-nationals, startups to long established companies - every business with which we do business receives the unparalleled level of service and range of products that has set Adhost Internet apart from the pack since 1995.

400 108th Avenue NE, Suite 700, Bellevue, WA 98004
www.adhost.com • (888) ADHOST-1

BiznizWEB inc
DYNAMIC WEB PUBLISHING
Internet Publishing
Big Bucks? Heavy Web Expertise? Operations Nightmare?
It doesn't have to be that way!
www.webpublishingtools.com
ad server polls
recruitment directories
local portals city guides calendars classifieds

Catouzer

Catouzer develops web-based intranet and Customer Relationship Management software solutions. With Synergy 2.0, Catouzer continues its lead in providing secure web-based work environments. ColdFusion developers now have the most advanced framework to develop secure web-based projects.

www.catouzer.com • 604 662-7551

Computerjobs.com

ComputerJobs.com is the leading Internet-based job search company in its current markets. ComputerJobs.com is solely dedicated to helping computer and information technology ("IT") professionals find great jobs with companies in need of IT employees. We provide thousands of job listings for candidates seeking jobs in the IT field. We also provide companies who list jobs with us access to the resumes of the hottest IT professionals on the Web. Our user-friendly, proprietary Web site provides IT job seekers and hiring companies a convenient, effective way to connect.

3200 Windy Hill Road, Suite 700 West, Atlanta, GA 30339
www.computerjobs.com

digitalNATION - a VERIO company

digitalNATION, VERIO's Advanced Hosting Division, is the world's leading provider of dedicated server hosting, with over 1,650 servers online today. dN's superior connected network and service abilities have earned dN a solid reputation as a first-choice provider of dedicated server solutions (Sun, Windows NT, Linux and Cobalt). digitalNATION has been providing online and network services for over six years. One of the first ISPs to provide dedicated servers running Microsoft Windows NT, the dN staff has unparalleled experience in this industry.

5515 Cherokee Ave, Alexandria, VA 22312-2309
www.dedicatedserver.com • 703 642-2800

Ektron

Ektron supports the next-generation needs of e-businesses by providing dynamic Web infrastructure solution tools designed for use by nontechnical staff. Ektron's flagship offering, eContentManager, gives staff members across an organization the hands-on ability to make real-time additions and updates to Web content without requiring knowledge of a programming language -- while still allowing for centralized administrative control and security. With competitive advantages such as ease-of-integration and drag & drop everything, Ektron is looking to provide these empowering products to customers, resellers and integrators.

5 Northern Blvd., Suite 6, Amherst, NH 03031
www.ektron.com • 603-594-0249

Enhanced Technlogies

Enhanced Technologies is a state of the art Web Consulting, Development, and Marketing provider built on the principles of ease of use, strength of service, and power of presentation. ETI is not just another firm offering Web development services. We are a professional Web Development company with many more services and capabilities than most of our competitors. We offer HTML (Hypertext)

design, Web graphics design, and CGI programming, but we also offer 3D graphics, inline, Java, and Macromedia Director animation.

6422 Grovedale Dr., Suite 301E, Alexandria, VA 22310
www.enhtech.com • 800-368-3249

EnterAct

EnterAct is the Business Services Group of 21st Century Telecom - Chicago's only single-source, facilities-based provider of bundled telecommunications. Combine this with EnterAct's excellent customer service and technical support and it's easy to see why EnterAct is the premier Internet Service Provider in Illinois.

407 S. Dearborn, 6th Floor, Chicago, IL 60605
www.enteract.com • 312-955-3000

Eprise Corporation

If your customers are looking for a content management solution, Eprise Participant Server FastStart Kit for Allaire ColdFusion developers can save you time and resources. Participant Server is a flexible-content management framework that enhances high-value business relationships through the delivery of timely, targeted, Web-based communications.

1671 Worcester Road, Framingham, MA 01701
www.eprise.com • 800 274-2814

Inteliant

Inteliant Corporation, a leading ColdFusion consulting firm, has an outstanding reputation for providing highly skilled developers for Internet, Intranet, Extranet, Software Development, or any ColdFusion application. Our national practice has emerged to meet the evolving needs of our clients by providing resources onsite or developing remotely. Our company provides the most cost effective service in the industry and we strive to add value to your projects by minimizing expenses whenever possible. Inteliant…"Delivering Intelligent Solutions

1150 Hancock Street, Suite 4, Quincy, MA 02169
www.inteliant.com • 800-815-5541

Intermedia, Inc.

Our advanced virtual hosting packages (powered by Microsoft Windows NT and Internet Information Server 4.0) offer an environment supporting everything today's advanced Web developer or sophisticated client could ask for. Complete ODBC support is available on plans B and C. We support Microsoft Index Server on all hosting plans.

953 Industrial Avenue, Suite 121, Palo Alto, CA 94303
www.intermedia.net • 650 424-9935

ISITE Design

ISITE Design is a customer-focused New Media firm celebrating two years of serving the Portland, San Francisco and Los Angeles markets. Our diverse team provides a full range of services for our clients. From conceptualization and original art generation to database integration and online marketing, our team is poised to provide your organization with the highest level of customer service and results.

615 SW Broadway Ste. 200, Portland, OR 97205
www.isitedesign.com www.buyspectra.com • 888-269-9103

On-Line Data Solutions

CoolFusion.com is dedicated to providing unique and powerful add-on solutions for ColdFusion development and implementation. The site is hosted and maintained by On-Line Data Solutions, Inc. – a leader in ColdFusion integration. Our ColdFusion integration products line is called inFusion – a combination of "infuse" and ColdFusion. For information about our flagship product, inFusion Mail Server, we invite you to read the online information (where you can also download the latest beta) and join the inFusion Mail Server support list.

24 Elm Street, Centereach, NY 11720-1706
www.coolfusion.com • 516 737-4668

PaperThin, Inc.

PaperThin offers a 100% ColdFusion-based packaged solution empowering users with out-of-the box Web publishing and dynamic content management functionality for internets, intranets and extranets. Rich, browser and role-based tools allow users to easily create, update, schedule and personalize content without HTML knowledge, while eliminating the need for client software, technical training or complex scripting. Open and extensible, developers can quite readily integrate custom ColdFusion code and applications.

267 King Caesar Road, Duxbury, Massachusetts 02332
www.paperthin.com • 800-940-3087

RSW Software

RSW Software is a wholly owned subsidiary of Teradyne, Inc., and specializes in Web application testing software. Established with the goal of providing best-in-class testing products, RSW offers a suite of products called the e-TEST Suite, which automates the process of testing business-critical Internet and intranet applications.

44 Spring Street, Second Floor, Watertown, MA 02172
www.rswsoftware.com • 508 435-8000

SaiSoft

As a recognized Allaire, Microsoft and IBM Solutions Provider, SaiSoft's Strategic focus is to become the most definitive Internet Architect, by building long lasting e-business development partnerships. Our target markets are organizations seeking to establish a competitive and Interactive e-business presence.

With development operations in India & the UK, SaiSoft also undertakes off-shore consultancy projects where a 'four-step implementation' model is adopted to meet client needs satisfactorily. The software training services provided by SaiSoft to its clients are not wholly exclusive to existing clients, but employees from other organizations and individuals alike are also catered to.

446 East Street, Plainville, CT 06062
www.saisoftonline.com • 860-793-6681

Sitehosting.NET

Successful electronic commerce starts at SiteHosting.net; a division of Dynatek Infoworld, Inc., which provides total Web development services. We offer personal and efficient customer service with reliability at value prices. All our plans include access to SSL (Secure Socket Layer). We support ColdFusion, Active Server Pages, Real Audio/Video, Netshow Server, and more. Our hosting price starts at $14.95/month.

13200 Crossroads Parkway North, Suite 360, City of Industry, CA 91746
www.sitehosting.net • 877 684-6784

Virtualscape

Why host with Virtualscape? Nobody else on the Internet understands what it takes to host ColdFusion like we do. Virtualscape is the leader in advanced Web site hosting. From Fortune 500 extranets to e-commerce sites and more, developers recognize our speed, stability, reliability and technical support.

215 Park Avenue South, Suite 1905, New York, NY 10003
www.virtualscape.com • 212 460-8406

Watchfire

Watchfire is the leading provider of software solutions that give organizations the power to ensure the quality and usability of their websites. Watchfire was recently voted one of the fastest growing Independent Software Vendors in North America (Information Week) and one of Canada's 25 "Up and Comers" (Financial Post). The Company currently sells into more than 60 countries and has over 50,000+ installed customers including over 50% of Fortune 500 companies and many of the largest e-commerce, government and education sites on the Web

135 Michael Cowpland Dr, Suite 400 , Kanata, Ontario K2M 2E9 Canada
www.watchfire.com • 613-599-3888

WinMill Software

WinMill Software is the premier resource for Internet and Intranet development, expert consulting, certified classroom training, Web-based education, e-commerce, development software, and technical support with expertise in site design and development, instructional design and delivery, application development, site hosting, training and site management. WinMill Software was founded on the principle of supporting the entire project lifecycle and our hallmark is a knowledge transfer process that maximizes the skills and intellect of your Internet development associates. We will work with your team to build a stable system architecture.

420 Lexington Avenue, Suite 307, New York, NY 10170
www.winmill.com • 888-711-MILL

What's the Well-Dressed Web Developer Wearing?

BY DAVID T. WATTS

The life of a Web developer is never boring. Technology shifts so quickly that we refer to "Internet years" the same way we refer to "dog years." We're constantly bombarded with new buzzwords. It's a full-time job just keeping track of what they mean, much less knowing how to integrate them into development.

That's one reason I like ColdFusion so much – it allows me to focus on other technology issues by making the middleware so simple.

Developers are usually too busy performing their daily tasks to keep on top of new technologies; as a CTO, I don't have a real job, and can do that for you this month. What should CF developers learn about? If I knew for sure, I'd make a killing in the stock market – no one can predict what will succeed, and the merits of a technology don't necessarily matter, as anyone who bought a Betamax will tell you. Nevertheless, here are my guesses.

At the top of the buzzword food chain, XML will play many important roles for Web developers. CF already uses XML in several ways – from WDDX to encapsulate data to VTML within the CF Studio environment. XML will be used on the Web client, with XHTML, XSL formatting rules and raw XML data. It will also be used in mid-tier communications. Databases will output XML; object brokers will communicate using it.

On the client, expect Flash interfaces to increase in popularity. Most Web applications are an ugly thing to behold – disjointed parts such as HTML, CFML and JavaScript sewn together to build applications that would make Frankenstein proud. Programmers have shaken their collective heads in disgust for years, faced with screen-rendering technology that is a flashback to the '80s. For as many years, many potential panaceas have been held out to us, such as client-side Java and ActiveX. Flash, on the other hand, has already infiltrated the desktop, and doesn't suffer the performance problems of Java or the platform specificity of ActiveX. The programming environment is far less sophisticated, but it works, and provides the functionality we've been waiting for. Most CF developers have left Flash for the "artists," but it's time to rethink that if you're involved in any sort of Web interface design.

Spectra is the 800-pound gorilla – a brand-new product, but already a market leader in its niche. People are flocking to it for inexpensive content-management and workflow solutions. For veteran CF developers Spectra requires a lot of learning: how to write Spectra applications, how to deal with the issues found in any version 1.0 release, and which applications are suitable for Spectra and which aren't.

New kinds of Web-enabled devices will significantly change how we write applications. For years most Web developers have written applications so they'd work in the broadest range of browsers rather than write custom browser-specific code. These days will soon be over; devices will present so many different interface specifications that one size will no longer fit all.

Enterprise IT departments have long dreamed about managing all their network information in a single location. Directory servers make this a reality. They've been around for a while now, but with the hype around Active Directory, more people will implement some directory service, whether it's AD, NDS or native LDAP. The ins and outs of directories will become very important to developers. Using CF's Advanced Security engine to integrate with them will also be important.

Most CF applications are two-tier: database and application server. This works well for most Web applications, but not as well in a distributed environment, where different databases are spread across a vast area. Object broker systems like DCOM/MTS, CORBA and message queues allow the distribution of application logic across disparate systems, but increase the complexity of application development.

Java is already a technological success. Many of CF's competitors rely on it, and it's a standard component in server-side applications. The biggest arena for Java within CF applications may be integration with other systems. CF now allows you to use Java CFXs, which should transport easily between various platforms. Java is a logical choice for middleware components, interfaces to object brokers and other niches that aren't easily filled with CF alone. Servlets, while a nifty technology, don't offer much to the typical CF developer. It'll be interesting to see how Allaire's purchase of Live Software and the Ejipt component server platform will affect CF.

Many developers think that the rate of change of Web technologies has decreased now that the "browser war" is essentially over, but you're likely to see more real change now than ever before. The typical overworked Web developer may not have time to learn about all these changes, but you should certainly keep them on your radar. Start cracking the books and writing the code, and stay tuned to ***CFDJ*** to see how you can use these new toys!

ABOUT THE AUTHOR

David T. Watts is cofounder and CTO of Fig Leaf Software, one of the leading ColdFusion development firms. He coauthored the first and second editions of the ColdFusion Web Application Construction Kit, *and is a certified Allaire instructor.*

DWATTS@FIGLEAF.COM

Catouzer

www.catouzer.com

Intermedia.NET

TAKE control

TEST DRIVE FREE DEMO ACCOUNT

FOR FREE SET UP
USE PROMO CODE CFDJ2000

Advanced NT web hosting
Advanced email hosting with DeskPilot™
Advanced Cold Fusion 4.5 hosting

New! Microsoft Exchange/Outlook hosting

Applications hosted:

MS Exchange Server
MS Outlook 98 and 2000
MS SQL Server 7.0
MS Access, FoxPro
Active Server Pages
FrontPage 98 and 2000
Cold Fusion 4.5
StoreFront
Drumbeat
Cybercash, Signio
WebTrends

Instant Control!

Add domains
Add disk space
Add email accounts
Add mailing lists
Register Cold Fusion Custom Tags
Register ODBC
Register Active X DLL and OCX

sign up online
and activate your account in 10 minutes

www.intermedia.net

Intermedia.NET, Inc.
Palo Alto CA 94303
sales@intermedia.net

800.379.7729

NETWORK SOLUTIONS Gold Partner

Microsoft Certified Solution Provider

<allaire> Alliance Partner

Copyright Intermedia.NET, Inc. 2000. All Rights Reserved. Microsoft, FrontPage and Windows NT are registered trademarks of Microsoft Corporation. ColdFusion, WebTrends, CyberCash and all other trademarks are property of their respective holders.