

COLD FUSION Developer's Journal

ColdFusionJournal.com

June 2000 Volume: 2 Issue: 6

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

Editorial

From Not Enough to Way Too Much

Robert Diamond page 5

Tips & Techniques

Getting Started with CF

Steve Dan page 16

Product Review

EZwidget

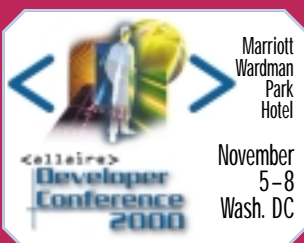
David Schwartz page 42

Journeyman ColdFusion

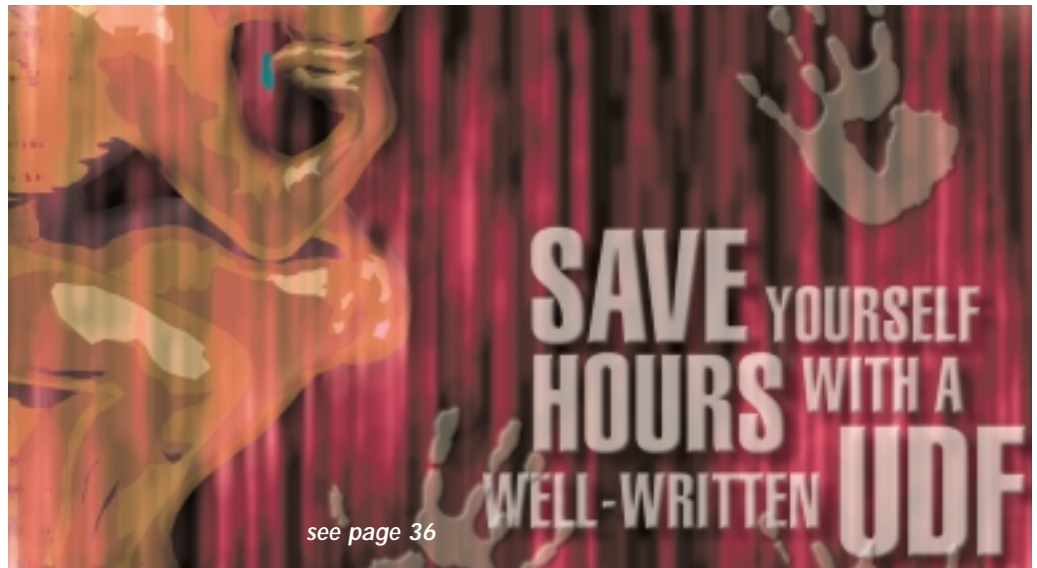
Getting Focus(ed)

Charles Arehart page 48

CFDJ NEWS
page 54



**SYS-CON
MEDIA**



CFDJ Feature: Making the Most of Verity

6

**Build a more scalable search
engine in ColdFusion Server**

Shahriyar Neman &
Mark Murphy

<BF> on <CF>: Access Denied

12

Microsoft Access and its use with ColdFusion applications

Ben Forta

CFDJ Feature: So You Want to Manage a Session on Load-Balanced Servers?

24

Preparing a CF app for ClusterCATS implementation

Marc Funaro

Interview: Jeremy Allaire Shares His Vision

32

The future of CF, Spectra and e-business

Ajit Sagar

CFDJ Feature: The Art of Creating Functions in ColdFusion

36

How to synthesize user-defined functions

John Morgan

Foundations: Tipping Points

44

Little things that make a big difference

Hal Helms

Able Commerce

www.ablecommerce.com

Didital Nation

www.dedicatedserver.com

Able Commerce

www.ablecommerce.com

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 GELAN
PRODUCTION EDITOR CHERYL VAN SISE
ASSOCIATE EDITOR NANCY VALENTINE
PRODUCT REVIEW EDITOR TOM TAULLI
TIPS & TECHNIQUES EDITOR MATT NEWBERRY

WRITERS IN THIS ISSUE

CHARLES AREHART, STEVE DAN, ROBERT DIAMOND,
BEN FORTA, MARC FUNARO, HAL HELMS,
JOHN MORGAN, MARK MURPHY, SHAHRIYAR NEMAN,
AJIT SAGAR, DAVID SCHWARTZ

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 UAT A. KIRCAALI
VICE PRESIDENT, PRODUCTION JIM MORGAN
VICE PRESIDENT, MARKETING CARMEN GONZALEZ
CHIEF FINANCIAL OFFICER ELI HOROWITZ
ADVERTISING ACCOUNT MANAGER ROBYN FORMA
ADVERTISING ACCOUNT MANAGER MEGAN RING
ADVERTISING ASSISTANT CHRISTINE RUSSELL
ADVERTISING INTERN MATT KREMKAU
GRAPHIC DESIGNER ALEX BOTERO
GRAPHIC DESIGNER JASON KREMKAU
GRAPHIC DESIGNER ABRAHAM ADDO
GRAPHIC DESIGN INTERN AARATHI VENKATARAMAN
WEBMASTER BRUNO Y. DECAUDIN
WEB DESIGNER STEPHEN KILMURRAY
WEB SERVICES INTERN DIGNANT B. DAVE
WEB SERVICES INTERN BRYAN KREMKAU
CUSTOMER SERVICE LEIGH VASQUEZ
JDJ STORE.COM JACLYN REDMOND

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,
REUBLISH AND AUTHORIZE ITS READERS TO USE
THE ARTICLES SUBMITTED FOR PUBLICATION.

WORLDWIDE DISTRIBUTION

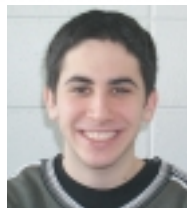
BY CURTIS CIRCULATION COMPANY 739 RIVER ROAD,
NEW MILFORD, NJ 07646-3048 PHONE: 201 634-7400

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.

From Not Enough to Way Too Much



BY ROBERT DIAMOND

A few weeks ago I traveled abroad and visited London for the first time. Upon arrival I followed the hordes of passengers through the long, winding corridors of Gatwick Airport's International Arrivals terminal. With signs pointing the way and chirpy airport officials ushering us along, I couldn't help but feel like a sheep. Finally we – and about eight other flights – arrived at Immigration. The immigration officials were moving rather fast, stamping passports and welcoming people to glorious England as fast as they arrived. When my turn came I wasn't quite as lucky. I'm not sure whether I "looked like a terrorist" or what (I think seven hours on a plane can do that), but I was peppered with questions about the reasons for my trip (vacation), where I was staying, and so on and so forth. I was then asked to produce my return plane ticket to the United States, which revealed that it was purchased by someone other than me – which they didn't like one bit. The fact that it was purchased by another family member didn't help things. I was detained to talk to a supervisor who then asked to see my hotel confirmations. Still not good enough. Finally I had to pull out my travel itinerary and concert tickets for shows that I was seeing while there to convince them of my innocent purposes. Then I was questioned about why I was seeing the same show three times. (Another story there.) Whew, I was finally in. After this, my luggage was delayed for another 45 minutes. (Yet another story.)

What I'd really like to talk about is information collection and when is it too much.

The Immigration people obviously have to be careful – after all, they have the security of a country to protect. But what I think is completely unnecessary is some of these huge marketing surveys that are out there disguised as simple parts of a registration process. Half the questions have me running back and forth between my tax returns, credit card receipts, CD collection and an encyclopedia for the answers. It's scary how big some of these surveys are getting. And it's getting worse. In about a year I think I'll be sending someone pictures of my family as part of the registration process to access their Web site. I do realize and accept that sites need to collect information about their users, both for marketing purposes and to make the site better. I can also accept that in some cases they're collecting my information to sell it, but I do feel that there needs to be a limit on how much information they should be taking.

If I'm at a music site, I can understand them asking what sort of music I enjoy, but they shouldn't be asking me about music on a gardening site. I've spoken about this with several colleagues who also find it annoying; most of them put in fake information for questions such as these. If they like classical music, they'll put down rap. So how accurate is the information anyway? No one really knows. The marketing guys want as much info as possible and it's not a priority to worry about the accuracy. I think it's up to us – those on the technical end of things, and those who are probably spending the most time online – to try and convey that to the higher-ups. I'm not looking to start a revolution, just some awareness about convenience. Who's with me?



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

Robert Diamond

MAKING THE MOST OF

VERITY

BUILD A MORE SCALABLE SEARCH ENGINE IN COLDFUSION SERVER

BY SHAHRIYAR NEMAN
& MARK MURPHY

Search functionality has become the status quo for all major Web sites. The typical search box/button found on home pages across the Net is considered the ultimate in user-friendly design: users type in what they're looking for and the search engine finds it quickly and easily.

By applying the tips and tricks illustrated in this article, developers can augment the Verity search engine that's packaged with ColdFusion to create a more robust – and scalable – search engine. All it costs is a little time and ingenuity.

ColdFusion Server comes packaged with the Verity search engine, a tool that makes short work of indexing, searching and retrieving information stored in virtually any format on Web and file servers. Yet the version of Verity included with

ColdFusion Server provides only a limited subset of the functionality and features that are part of Verity's enterprise-level "Information Server."

This article explores some novel ways CFML and Verity can be implemented to build a more scalable search engine – and in several cases overcome some of the limitations imposed by the built-in Verity search engine.

Background Overview

All Verity functions can be performed through CFML templates using built-in ColdFusion tags. These tags are well doc-

umented within the CFML Language Reference Guide included with ColdFusion Studio. For information purposes we recap these tags here:

- To create, delete, map, repair or optimize a Verity collection:

```
<cfcollection action="action"
collection="collection" path="implementation
directory" language="language">
```

- To update, add or delete keys from a collection or to purge or refresh a collection:

```
<cfindex collection="collection"
action="action" type="type" title="title"
key="id" body="body" custom1="custom1" cus-
tom2="custom2" urlpath="url"
extensions="file_extensions">
```

```
query="query_name" recurse="yes/no" external="yes/no" language="language">
```

- To perform searches on a Verity collection:

```
<cfsearch name="search_name"
collection="collection_name" type="criteria"
criteria="search_expression" maxrows="number"
startrow="row_number" external="yes/no"
language="language">
```

Types of Verity Collections

Three types of collections can be created using Verity:

- **File:** An index of one particular file
- **Path:** An index of all files of a specified extension within a specified directory
- **Custom:** An index of database data

Certain situations arise where it's not clear which of the three types a collection should be. Sometimes a collection needs to be a mixture of different types of data. (The implementation of such a scenario will be discussed later.) Other caveats that occur with the file-type collection are discussed in the Allaire knowledge base, www.allaire.com/Handlers/index.cfm?ID=1600&Method=Full, and they're worth reading.

Types of Verity Searches

Verity searches come in two types – simple and explicit. Depending on the functionality required from your search-engine implementation, one type may be preferred over the other.

- **Simple:** These types of queries allow for simple word or phrase searches using comma-delimited strings and/or wildcards. When using commas, each one is treated as a logical OR; if commas are omitted, the string is treated as a phrase. In addition, simple operators such as AND, OR and NOT can be used in simple queries.
- **Explicit:** These types of queries allow for more refined searches using operators and modifiers. Operators include: <, <=, =, >, >=, Accrue, AND, CONTAINS, ENDS, MATCHES, NEAR, NEAR/N, OR, PARAGRAPH, PHRASE, SENTENCE, STARTS, STEM, SUBSTRING, WILDCARD and WORD

The “Developing Web Applications with ColdFusion” section of the online docs included with ColdFusion Studio provides excellent documentation on the types of searches that can be performed on a Verity search engine.

Implementation Techniques

This section details implementation techniques that can be used to improve

your Verity search engine code and even bypass the apparent limitations set by the watered-down version of Verity. All these examples work under a Windows NT environment with a Microsoft SQL Server 7.0 DBMS, but can be modified to work under any other environment.

Overcoming Two Custom Fields per Collection

First we address the limitation of having only two custom fields per collection. Some situations call for indexing more than two. For example, you may want to index the contents of a database table and include more than four fields to be indexed (four is the limit within a Verity collection because the body, title, custom1 and custom2 fields can hold custom information). A simple solution is to combine several fields into one, separating each field by a selected delimiter. To accomplish this you must be certain that the data in any combined field will never contain that delimiter. An example of how to create such a collection is located in Listing 1.

Combining Database and File Data

Under certain circumstances you may want to create a collection that's a combination of database data and file data. For example, imagine that tbl_image from Figure 1 had an additional attribute, image_text, that represented the filename associated with that image. If we wanted to create a collection that included the text in the file specified by the attribute image_text, we'd first have to query the database for the image information, then create a collection of type “file.” ColdFusion's <cfindex> tag does the rest by automatically looping through the query to index the files from the paths specified in the query. Listing 2 gives an example of how this would be done.

In Listing 2 we've dynamically created a Verity file collection that includes database information as well as data from a text file. This operation isn't limited to text files and can be performed with other types of files that Verity supports. Look closely at the code: you'll notice that when the URL attribute is created by the get_images query, an extra space is appended at the end. At first glance this may seem like a

mistake, but it's deliberate and there's a good reason for it.

When Verity performs searches on a collection, such as the one created in Listing 2, the value of the URL attribute returned by the search is a concatenation of the URL specified when the collection was indexed and the filename searched; that is, if we specified www.foobar.com as the URL, a search might return a result with the URL attribute something like www.foobar.com/file1.txt.

ColdFusion sites that access content through URL parameters may not want files that are indexed to show up in the URL field of the returned-search query from Verity. This is where the space at the end of the URL attribute comes into play. It serves as a delimiter so that, when searches are performed, you can get the proper URL sans filename by simply applying the listfirst() function on the URL value returned. For example:

```
<cfsearch collection="image_collection"
name="search_images" type="SIMPLE" criteria="dog image" language="English">
<!-- output all the URL values minus the
concatenated filename added by Verity -->
<cfoutput query="search_images">
#listfirst(url, " ")#
</cfoutput>
```

Certain modifications can be made to Verity searches to make them more efficient. For instance, if you want to perform searches only on a particular image_group_id, you could use the following code:

```
<cfsearch collection="image_collection"
name="search_images" type="SIMPLE"
criteria="(CF_CUSTOM2<starts>#url.image_group_id#&chr(35)#<AND>(#url.search_criteria#<OR>CF_TITLE<substring>#url.search_criteria#<OR>CF_CUSTOM1<substring>#url.search_criteria#<OR>CF_CUSTOM2<substring>#url.search_criteria#)"
maxrows="1000" language="English">
```

With this type of search in place Verity filters out all images that aren't of the image_group_id specified by the url.image_group_id parameter.

Searches can be speeded up by periodically

tbl_image_category	tbl_image
Column Name	Column Name
Category_ID	Category_ID
Category_Name	Image_ID
Category_Description	Width
	Height
	MetaData
	Filename
	Extension
	Alt_Text

FIGURE 1: The relational schema for an image-indexing database

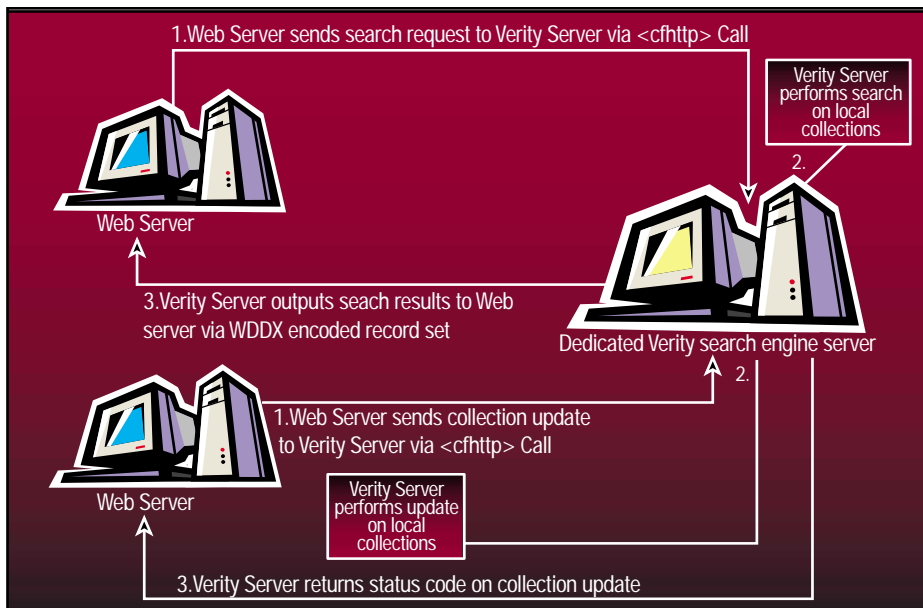


FIGURE 2: Diagram outlining client/server Verity search engine model

cally optimizing Verity collections. Optimization can be performed either programmatically or through the ColdFusion Server Administrator. It's a good idea to create a template that programmatically optimizes your collections and uses the ColdFusion Scheduler to run it every night. A sample template would look like this:

```
<cfcollection action="OPTIMIZE" collection="image_collection">
```

Finally, when performing searches on a Verity collection, certain words and characters in the search phrase will cause the search to error. To avoid this you can "clean" any search strings before you send them to Verity. A simple way to do this is to delete the offending characters and/or words. A utility like this already exists – in the form of a custom tag named `<cf_verityclean>` – and can be downloaded for free from Allaire's Developer's Exchange Site at www.allaire.com/developer/gallery.cfm.

Scaling Verity for Clustered Web Server Environments

Within clustered-server environments traditional implementations of Verity wouldn't be ideal. Under clustered NT

environments collections could be stored on a separate file server that all the Web servers can access via UNC paths or SMB mapped drives. The problem with such an implementation is that the Web servers themselves are doing the searching, that is, the local Verity engine on each ColdFusion Web Server is taking up that server's CPU time to perform searches and updates to various collections. Clearly the main function of a Web server should be to serve Web pages, and any CPU time taken for other tasks is highly undesirable. This situation is analogous to placing a DBMS on each server, then having each one serve Web pages and perform database queries. In our experience making network calls to collections via UNC paths is a slow process.


A more scalable and robust solution to this problem is to designate one server as a Verity server. This server will then take Verity search-and-update requests from all the Web servers through HTTP calls. To accomplish this without purchasing the full-scale version of Verity, CFML client and server templates must be implemented. The client template will reside on each of the Web servers and will be called when a Verity search or update is performed.

Subsequently the client template will call the server template residing on the dedicated Verity server via an HTTP call. Each client HTTP call posts requests to the server template, and once the server template receives a request, it performs the desired action and returns results to the client template. The client template can then use this data in any fashion desired.

Listing 3 provides a sample client template, and Listing 4 illustrates a complementary server template. Figure 2 gives an overview of the entire client/server model.

Note: Although the purpose of the Verity client/server template is to make Verity scale better, calling the `<cfhttp>` tag is a potential bottleneck that limits the scalability of this implementation. Due to problems encountered with the single-threaded nature of the `<cfhttp>` tag, it's good programming practice to place a lock around all calls to it. This locking mechanism, which is responsible for the consequent scalability limit, causes multiple images of templates that call `<cfhttp>` to wait for the release of the lock before execution.

Conclusion

The built-in Verity search engine packaged with ColdFusion can be augmented by implementing the tips and tricks illustrated in this article. The result is a more robust and scalable tool, developed in a relatively short amount of time. Best of all, the scalability attained is free. It's a combination of features any developer will appreciate. 

ABOUT THE AUTHORS

Shahriyar Neman is CTO of the Next Network, an ASP that delivers total computing packages to small- and medium-sized businesses through the Internet. He holds a BA in computer science from NYU and is currently pursuing his master's degree.

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

sneman@thenextnetwork.com mark.murphy@corp.convey.com

LISTING 1: Example of Meshing Together Database Fields into One Verity Collection

```
<cfquery name="get_images" datasource="#datasource#">
SELECT filename, alt_text,
convert(varchar, width) + ',' +
convert(varchar, height) as custom1, metadata,
convert(varchar, category_id) + ',' + extension as custom2
FROM tbl_image
</cfquery>

<cfcollection action="create" collection="images"
path="E:\images_collection\"
```

```
language="English">
```

```
<cfindex action="refresh" collection="images" key="filename"
type="custom" query="get_images" title="alt_text" body="meta-
data"
custom1="custom1" custom2="custom2" language="English">
```

LISTING 2: Example of Meshing Together Database Data and File Data

```
<cfquery name="get_images" datasource="#datasource#">
SELECT
'http://www.ffoobar.com/index.cvy?fuseaction=show_image&image_
_id=' +
```


Conceptware

www.concept.com

```
<cfcollection action="create" collection="images"
path="E:\images_collection\" language="English">
```

LISTING 3: Example of Client Verity Search Engine Template

The path to your future begins here...



hi-tech jobs for hi-tech pros

```
<cfoutput>
<table>
  <tr>
    <td>Image Text</td><td>File
Name</td><td>Extension</td><td>Width</td><td>Height</td>
  </tr>
</cfoutput>
<cfoutput query="search_articles">
  <tr>
    <td><a href="#url#">#title#</a></td>
    <td>#key#</td>
    <td>#listlast(custom2)#</td>
    <td>#listfirst(custom1)#</td>
    <td>#listlast(custom1)#</td>
  </tr>
</cfoutput>
</cfoutput></table></cfoutput>
```

```
<cfswitch expression="#url.action#">
  <cfcase value="search">
    <!---search for within a publication: articles only-->
    <cfsearch collection="image_collection"
name="search_images"
type="SIMPLE" criteria="#url.search_criteria#"
maxrows="1000" language="English">
```

```

        <cfif search_articles.recordcount>
        <cfwddx action="CFML2WDDX" input="#search_articles#"
output="wddx_output">
        <cfoutput>#wddx_output#</cfoutput>
        <cfelse>
        <cfoutput>0</cfoutput>
        </cfif>
    </cfcase>

```

```
<!-- client code for update functionality is not shown
but can be easily figured out -
  simply encode the update query into WDDX and post it to
this template --->
<cfcase value="update">
  <!--decode the query-->
  <cfwddx action="WDDX2CFML" input="#form.imagewddx#" out-
put="imagequery">
```

```
<cfindex action="UPDATE" collection="image_collection"
type="custom"
key="key" title="title" query="pubquery" body="body"
custom1="custom1"
custom2="custom2" language="english">
<!-- return a success status code -->
<cfoutput>1</cfoutput>

</cfcase>
</cfswitch>
```

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

Developers Network

www.developersnetwork.com

Access Denied

BY
BEN
FORTA



I have yet to address a group of ColdFusion developers without somebody requesting my opinion of Microsoft Access and its use with ColdFusion applications

As a rule I try to avoid the Access/CF discussion as it inevitably provokes strong debate and even stronger emotions. Besides, the truth is – regardless of what I might suggest – Access remains an inexpensive and easily implemented solution. So why am I writing about Access now? Because I have come to realize that many users are considering only cost and performance in their decision-making process and are overlooking the bigger issues.

Let me start by making it very clear that I've nothing against Microsoft Access: it's a great desktop database product, perhaps the very best one there is. And despite this column's title (sorry, I couldn't resist that one), the comments that follow apply not only to Access. I'm writing about shared file-based databases in general, *all* shared file-based databases, of which Access happens to be the most popular.

File-Based Databases

You've undoubtedly used a word processor. When you open a document for editing, that file is opened on your computer and remains open until you close the document (or quit the application). From a functional perspective your document is a data file, much like an Access MDB file. When you use an Access MDB file (or a dBase DBF file or an Excel spreadsheet), the application you're running opens the file and directly manipulates the data in it. As you add, delete or update data, the application manipulates the opened data file accordingly. Even if you access the data via ODBC (perhaps using ColdFusion), that data file is actually open on the computer running the application. If two applications (or applications on two computers) open the same file at once, then both appli-

cations open the file (and some sort of locking mechanism is used to arbitrate requests).

Access, FoxPro, dBase, FileMaker and Paradox are all examples of file-based databases.

Client/Server Databases

Unlike file-based databases, client/server database files are never accessed directly. In fact, they're usually on another computer altogether, one with restricted access. So how do you access client/server data? Your application communicates with a driver which in turn communicates with the database server and only the database server ever manipulates data.

As such, the underlying data files used to store client/server data are never opened by client applications. Even if you use your DBMS's bundled management tools (for example, SQL Server Enterprise Manager), all data manipulation is actually performed by the database server. Data access via ODBC (or even native database drivers) works much the same way: ODBC is merely a client of the database server; it can't access the data directly. If multiple client applications access data simultaneously, all requests are sent to the database server and it processes them sequentially or concurrently.

SQL Server, Oracle, DB2, Informix and Sybase are all examples of client/server databases.

Data Integrity

Most of us have experienced computer crashes (GPFs, blue screens of death and so on). When something like this occurs – as you're probably painfully aware – you run the risk of corrupting the data in any open application. If you're lucky you'll lose just

your most recent changes, but often entire documents and files become corrupt (containing random, incomplete or truncated data). If you don't quit applications gracefully you run the risk of losing data. We've all experienced this and hopefully we all hit the save button (or use an autosave feature) to help ease the pain somewhat.

Web servers crash; it's unfortunate but it's a fact. File and print servers seem to stay up forever; most Internet servers don't. I have a NetWare 3.12 server in my basement running on a 386/33 with 8 MB of RAM that has been up for over 200 days and hasn't missed a beat. I don't see many Web servers (running on any platform, not just Windows) that can make that claim. Hopefully servers and their uptime will improve, but for now you must assume that your Web server will crash and will need rebooting. And hitting the reset button is anything but graceful. If you have an Access file open (just as if you had a Word file open), you run a very real risk of trashing that file, rendering it utterly useless and requiring that you restore a backup. Of course, backing up open files is highly problematic itself, so you might not even have a good backup of your data. Not a pleasant thought at all.

Client/server databases don't run this risk. If your Web server is rebooted, all open connections to the database server will be broken but that's it: the database server itself stays up and so the data remains safe. (We're assuming of course that your database server crashes less frequently than your Web server, which is an argument for never running the Web server and database server on the same machine.)

Get PopChart.™ Get Noticed.

PopChart.™ with the revolutionary DataFunnel™ Technology, allows you to quickly create state-of-the-art Web pages, capable of generating dynamic charts and graphs faster and more effortlessly than ever.




www.popchart.com

Competition is fierce on the Web.

We have what you need to stand out.

Only PopChart™ by CORDA Technologies® has the patented new DataFunnel™, the world's fastest, most dependable way to integrate bright, colorful, interactive charts and graphs into your Web designs.

The most powerful charting technology in the world is also the easiest to use.

 CORDA Technologies™



TO EXPERIENCE THE POWER OF POPCHART, logon to www.popchart.com for a free trial version, or call 888.763.0517 and see how CORDA Technologies can illuminate your data @ the speed of light.

PopChart®. MAKE THEM STOP AND STARE.

Security

Web servers are inherently insecure. They are highly visible, their IP addresses are usually known, the software they are running can easily be determined and path information can often be inferred. Couple that with the frequent discovery of new holes in Internet software and it becomes blatantly obvious why Web servers are primary targets for hackers and cyberthieves.

Good practice dictates that Webmasters be somewhat paranoid, always considering their Web servers to be highly vulnerable. The assumption that anything on the Web server can and will be stolen should be the driving force in determining what actually gets put on that server. (On a side note, this is why you must never hard-code login information or passwords in CFM files. Those files may get stolen and that information could end up in the wrong hands.)

So what has all this to do with databases? For ColdFusion to access file-based databases, it needs access to those files. This means the data files must reside on the Web server or on a path that it can access. And if CF can access those files, so can anyone who breaks into the server. If the thought of your code being stolen scares you, then the thought of your data being stolen could keep you awake at night.

Client/server databases don't run any such risk. If a hacker were to gain access to the Web server, the most they'd have access to is information about the database server, but not the database server itself. (We're assuming that the database server itself is secure, as it should be.)

Features

Although data integrity and security are my primary concerns here, there is one other important point to mention. Client/server databases support important features that are a pivotal component of secure and scalable applications. These include:

- **User security:** Used to manage and restrict data access to prevent accidental or malicious damage.
- **Stored procedures:** Used to improve performance as well as provide an additional level of data abstraction and security.
- **Triggers:** Used to perform server-side data manipulation resulting in improved performance and better quality data.
- **Scheduled execution:** Used for the scheduled execution of data housekeeping operations on the server. These execute far quicker than client-side events, with no client impact (and less server impact).

And There's Much More Too...

It's worth noting that these features tend to be very DBMS-specific, and support and implementation vary dramatically from one DBMS to the next. But regardless of the DBMS in question, all client/server DBMSs offer features of this type and they should be used.

Summary

File-based databases are cheap and easy to use and for smaller applications they usually perform quite well, sometimes even outperforming client/server databases. But they're also highly susceptible to data corruption and are terribly insecure. For most organizations there's nothing more precious than data and if this is true of your organization, then file-based databases should never be used with ColdFusion. This alone should be enough to push you toward client/server databases. When you add all those extra features to the equation, the conclusion is obvious: desktop databases belong on desktops, not on Internet servers.



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, as well as Sams Teach Yourself SQL in 10 Minutes. He recently released Allaire Spectra E-Business Construction Kit, and is now working on books on WML and JSP.

Eprise

www.eprise.com

Getting Started with CF

Custom tags, stored procedures and stylesheets are real timesavers

BY
STEVE
DAN



The software development industry demands that we learn constantly and adapt to the new way of doing things. If you've been in this business for more than about five years, chances are you've already had to retool your skillset at least once. After 10 years in this business, I've had to retool a few times. While it's exciting to be on the cutting edge, it can also be frustrating when you're stuck.

When I first started using ColdFusion, I found that there were many tricks to getting it to work the way I wanted it to. If you're new to CF, you may have run up against various development problems; if so, this article is for you. I'll be serving up some tips and techniques that you can use to help you up the learning curve.

Retrieving and Displaying Data from a Database

I've found that the ability to retrieve data and display it on a Web site is one of the services most commonly asked for, so this section of my article is aimed at helping to get you going.

Even if you've only been using ColdFusion for a short period of time, you probably have found that the CFOUTPUT tag is one of the most valuable tags for returning database information or evaluating variables. But that's only half of the equation. This tag is typically used in conjunction with the CFQUERY tag in order to return results. It doesn't make sense to discuss the CFOUTPUT tag until we first go over briefly what the CFQUERY tag does: CFQUERY is the request part of the process and the CFOUTPUT is the results part.

The CFQUERY Tag

The CFQUERY tag is used to specify the SQL statement you want to run. It takes at least two attributes – NAME and DATASOURCE – as well as the SQL query itself in order to successfully run your query. Here's an example. Let's say you want to retrieve all the product names from the products table. The SELECT statement would look like this: "SELECT product_name

FROM products." To put this into a CFQUERY tag, you'd just add the CFQUERY wrapper around it like this:

```
<CFQUERY NAME="get_products" DATA-  
SOURCE="Ecommerce" >  
    SELECT product_name  
    FROM products  
</CFQUERY>
```

I named the query "get_products" because that's what this query does. It's a good practice to name your queries in such a way that anyone could guess what the query does just by reading the name. While I used quotes around all the values in the CFQUERY statement, it's not required in the CFQUERY tag. It does help readability to put the quotes in, especially in ColdFusion Studio, because once the values are in double quotes, they are color-coded blue to help separate values from the actual CF tag attributes. Whether you put these values in single or double quotes, it's processed the same by ColdFusion, but only the double quotes will be color-coded. Unless you're connecting to different databases in the same application, you'll use the same DATASOURCE name for all your queries.

Notice that the actual SQL select statement isn't in quotes. Depending on your environment, you may need to prefix the table name with the database name and the table owner name (dbo) as in Prod_DB.dbo.products for an MS SQL server environment. If we had some WHERE criteria, the value we're looking for would need to be in quotes (if the value were character data). Say, for example, we wanted all the products from Allaire. Our

WHERE criteria would look like this:

```
<CFQUERY NAME="get_products" DATA-  
SOURCE="Ecommerce" >  
    SELECT product_name  
    FROM products  
    WHERE manufacturer = "Allaire"  
</CFQUERY>
```

Before ending our discussion of the CFQUERY tag, there's one other attribute worth mentioning and that's the CACHEDWITHIN attribute. If you have some relatively static information in a long database table and it takes a long time to retrieve that data, users may find it annoying. If the data were to be cached and retrieved only every few hours (or at even longer intervals), then those using the system between those refresh times would find the retrieval near-instantaneous (depending on their connection).

How do we implement this CACHEDWITHIN capability? First check to be sure that query caching is enabled in the CF Server, then add the attribute into your CFQUERY tag as follows:

```
<CFQUERY NAME="get_products" DATA-  
SOURCE="Ecommerce"  
CACHEDWITHIN="#CreateTimeSpan(1,0,0,0)#">
```

Notice that the value of the CACHEDWITHIN attribute is a CF function named CreateTimeSpan, used to create time and date calculations. It takes four parameters, which are (in order) days, hours, minutes and seconds. As defined here, the function will create a time object of 24 hours or one day. Using this piece of code ensures that the query will not be reretrieved within

Allaire

www.allaire.com

a 24-hour period. If you decide you want to make your retrieves immediate (in effect changing all time span values to zero), it's better to remove the attribute rather than set all values to zero as there's some amount of overhead involved in resolving this attribute.

The CFOUTPUT Tag

Now that we have our query defined, we get it to display all the results by means of the CFOUTPUT tag.

The CFOUTPUT tag has a few attributes. The most important one to know is the QUERY attribute, used to name the query that you want to display. In our example we called our query "get_products" so we use that same name in the QUERY attribute. If we didn't use the QUERY attribute – and this is very important to know – only one row of data would be retrieved. By adding the QUERY attribute, we've told CF to retrieve all the results, not just one row. Our CFOUTPUT tag to display the results of our query would look like this:

```
<CFOUTPUT
QUERY="get_products">#get_products.
product_name#<BR></CFOUTPUT>.
```

Notice that the QUERY attribute value is in quotes once again but the actual variable we're displaying isn't. Also notice that I've used the name of the query as a prefix for the column name being displayed. While this isn't always necessary, it's required if you have more than one query in the same page – otherwise CF won't know which query you're referring to. Also note that I surrounded the entire variable name in pound (#) signs. Without the pound signs, the variable won't be resolved (by which I mean that the variable *name* won't be traded for the *value* of the variable).

Other useful attributes for CFOUTPUT tag include MAXROWS and STARTROW. By using them together you can control how many rows a user sees on a screen. This is most commonly used when there are many rows of data but you only wish to display, say, 10 at a time. You set the MAXROWS attribute to 10 and increment the STARTROW

“”

What's important to note here is that the IsDefined function... uses parentheses for the value it is to evaluate

attribute as necessary. Keep in mind that when you are scrolling through data, the data retrieval is repeating itself each time. You may only be displaying 10 rows of data but the entire result set is being retrieved each time – each time, that is, unless you are caching the query. This is another occasion when you'd want to cache the query to reduce the amount of work that the database server needs to perform. Retrieve the data once, then cache it for however long you think your user will be scrolling through the data, whether it's 10 minutes or an hour. Just balance the cache time with the frequency of data update so that, while you're saving the database management system from rerunning the same query, your users aren't missing out on the most recent data.

The IsDefined Function

How do you find out if a user clicked on a particular button? You check to see if the button exists using the IsDefined function. How would that look in code? Imagine you had two buttons that a user could choose between – to sort some data either by product name or by manufacturer, for example. Let's say the buttons were named product_sort and manufacturer_sort. Both would be submit buttons, but the user can click on only one of them at a time. To determine which one was selected you would perform an IsDefined check on the FORM variable passed to the action page in the middle of your SELECT statement, as shown in Listing 1.

What's important to note here is

that the IsDefined function, like all CF functions, uses parentheses for the value it is to evaluate. Also note that the value is in quotes. This is critical. If the quotes aren't used in the IsDefined function, an error will result because CF will try to evaluate the expression and that's not what you want it to do. You want to pass the name of the variable so CF can check to see if that variable exists. To evaluate it before you know it exists would be putting the cart before the horse.

Two things to keep in mind when using this function: use quotes but don't use pound signs. The IsDefined function can also be used to determine which radio button has been chosen by the user. You can determine which choice the user made by checking to see which one exists in the action page. The radio button selected will exist in the action page; unselected radio buttons will not. On the action page, you'd check the following:

```
<CFIF Not IsDefined
('FORM.agree_to_terms')>
    You must choose to accept the
    terms of the agreement in order to
    proceed.
<CFABORT>
</CFIF>
```

Go Back and Try Again

Here's a tip for dealing with data entry errors. Imagine a user hasn't made a selection for credit card type or some other important value. How can you send users back to the previous page once they've already hit the submit button? The secret lies in using the onclick attribute of the input control in conjunction with a blank form control. By surrounding a button inside a blank form control and telling the button to go back to its most recent form, you have in effect pressed the users' back button for them. This is better than just refreshing the form since most of the data (except some variables like passwords) will be saved on the form, meaning that the user won't have to reenter it all. Listing 2 shows how the code would look.

Using the CFABORT function inside the CFIF makes sure that no further processing is performed.

JavaOne

www.java.sun.com/javaone/

Does EQ Equal = ?

When do you use EQ and when do you use the equal sign? CF will generate an error if the wrong equal operator is used. So what's the deal? The way it works is this: to set the value of a variable, use the equal sign inside a CFSET tag as in `<CFSET TempCartID = 1>`. If you want to test a condition, use the IS or EQ operator, as in `<CFIF #FORM.country# IS "USA">`. You can't use the equal sign here – it would give you an error. The equal sign is used in CFSET statements, in assigning values to attributes and in SQL statements. The EQ, EQUAL or IS operator (all synonyms) is used for comparing two values in condition-testing situations like CFIF statements. If you're not used to the way CF does things, this may trip you up, so watch out for it.

Stored Procedures

You may be new to SQL, in which case here's a performance tip: use stored procedures as often as possible. People unfamiliar with stored procedures may not understand the advantages of using them or may not bother using them because they don't know how to implement them. A stored procedure is simply a SQL statement that has been saved and preoptimized by the server. All SQL statements need to be put through a query optimizer each time they're run. That means that your database server needs to analyze the query, determine the best way to retrieve the data, then retrieve the data. What's saved by using stored procedures is all that time spent figuring out how to get you your results. That may not seem like much time, but consider the following analogy. Imagine someone were to ask you a question that you didn't know the answer to. First you'd have to figure out where to look up the answer, next you'd need to actually do the looking up and last you'd give them the answer. By way of comparison, imagine someone were to ask you a question you already knew the answer to. You'd just tell them the answer. See the savings? That's why you should always use stored procedures. The server already knows the best way to retrieve the results when the query has been optimized and saved on the server as a

procedure. Okay, great, so how do we write a stored procedure? The same way you write any other SQL statement except that, once you have your query, you need to wrap it with a procedure statement. The following example uses the SQL server syntax:

```
CREATE PROCEDURE sp_products AS
    SELECT product_name , manufacturer
    FROM products
```

Frequently you may need to pass WHERE criteria to a query. How do you do this with a procedure? SQL server requires that you place the @ symbol before a variable passed to a procedure. You also need to indicate at the beginning of the procedure the data type of the variable that will be used. Do so before the AS clause, as in the following:

```
CREATE PROCEDURE sp_products_manufacturer
@manufacturer VARCHAR (100)
AS
    SELECT product_name , manufacturer
    FROM products
    WHERE manufacturer = @manufacturer
```

Procedures can be very complex but they don't all have to be. Once you have created your procedure, how do you use it in CF? There are two methods. The first way is to use our old friend the CFQUERY tag, as in:

```
<CFQUERY NAME="Products_manufacturer" DATASOURCE="#DSN#" cachedwithin
= "#createtimespan(1,0,0,0)#"> {CALL
sp_products_manufacturer ('#FORM.manufacturer#')} </CFQUERY>
```

To display the results of your stored procedure, simply use the CFOUTPUT tag as you would to display any other query.

The other method uses a tag called CFSTOREDPROC, which has some big advantages over CFQUERY especially for complex procedures, multiple result sets and generating return codes. But the first method will get you started and that's what I'm trying to help you do.

Does a Row Exist in the Database?

If you've ever had to determine whether a parent row exists in a database before trying to insert a child row, you're already familiar

with SELECT COUNT(*). This syntax also plays an important role in determining the number of records based on some WHERE criteria. But how do you get the results of that query into a variable that can be used in CF? By using the AS operator, as in `SELECT COUNT(*) AS num_products FROM products`. You can refer to this variable by prefixing it with the query name, which in this case is called count_products. The code would look like this:

```
<CFIF #count_products.num_products#
IS 0>
```

Creating an Easy-to-Maintain Look and Feel

If creating a consistent look and feel to your Web site is important to you, then one good way to achieve it is by using templates for your headers and footers. If your Web site has at least several pages, then you can really save development time (plus rework time later) by storing all your header and footer code in separate templates, then using the CFINCLUDE function to insert them into the main body of all your templates. This way, if ever you want or need to change your header or footer, you'll need to make that change in only one place, your header or footer template, and because you've INCLUDED a link to those templates in all your main body templates, your changes automatically show up in the headers and footers of all your pages. How do you set this up? I created a sub-directory called TEMPLATES under my main project directory. That's where I store the header and footer templates. Then I create a header page. For this example my header page includes some basic site navigation and some graphics (see Listing 3).


In every one of my templates I include the following line: `<CFINCLUDE template="/templates/header.cfm">`. This code allows me to set up a mapping in the CF Administrator listing /templates/ as a logical path and setting C:\projects\dc\templates\ as the directory path. Once this is set up, your header and footer templates will appear in all the main templates by just adding that one line

Allaire

www.allaire.com

Let's say for now that you don't

The last thing you need to do is

Though ColdFusion is a great tool for developing Web sites, figuring out how to get it all to work can sometimes be challenging. Remember to use all your resources to help you solve problems. There are many user groups around the country. See if there's one in your area and join it so you can discuss issues or tips you've learned with fellow developers. On the Web, the Allaire site (www.allaire.com) has a whole section devoted to developers where you can find answers to common questions or post your own question. There are several other Web sites that promote ColdFusion development including CFSpot, HouseOfFusion and CFAdvisor. Happy coding ! 



Allaire

www.allaire.com

So You Want to Manage a Session on Load-Balanced Servers?

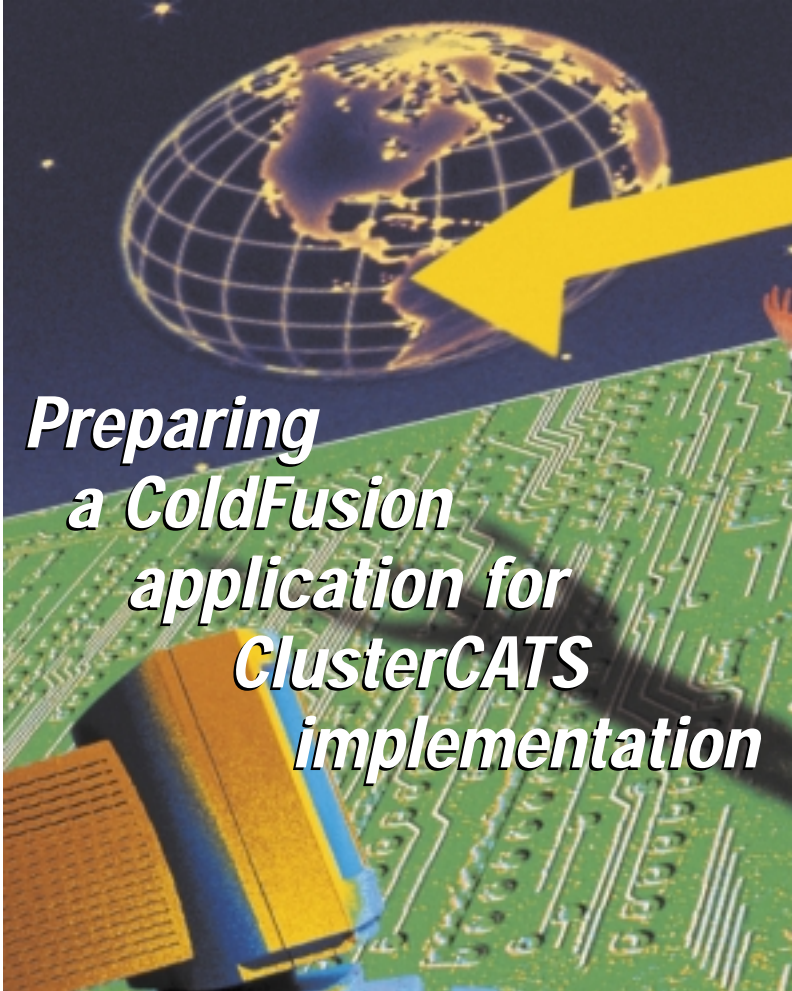
BY MARC FUNARO

Recently I was tasked with converting a ColdFusion application to run on a ClusterCATS Web server cluster. Many questions that hadn't been addressed in the initial design of the application arose during this process. As I resolved each of the problems, I began thinking that other developers might benefit from my experience if I were to document the process, address the issues and write a "white paper." The result is this article, which I hope will assist ColdFusion developers in developing or converting a CF-based application to support "client state management" (see sidebar) in a clustered server environment.

Some of you may never run into the problems that can occur when an application is developed using session variables for client state management. Moreover, this article isn't intended to make "converts" out of developers wishing to use session variables. Every application and situation is unique (that's what makes application development so much fun, right?). Not all situations call for the changes I'm going to discuss in detail here. It's up to you, as a developer or project leader, to decide whether your application will be designed to use client or session variables to manage client state; this article will arm you with facts that can help you make an informed decision.

Assumptions

I'm going to assume you're familiar with the basic concepts of ColdFusion programming, including setting variables, outputting variables to a Web page and understanding variable "scope." I'll also assume you understand simple database transactions like updating, selecting, inserting and deleting data; basic database administration on your chosen database platform; and basic database terminology. It will also help for you to be familiar with ColdFusion Administrator and you certainly should have a working knowledge of your selected operating system, Web server and clustering software.



Preparing a ColdFusion application for ClusterCATS implementation

This article targets users of Microsoft SQL Server 7 with Enterprise Manager, ColdFusion 4+, Windows NT, IIS and ClusterCATS (included with CF Enterprise Edition). However, with knowledge of your particular applications, the steps outlined here can be applied to almost any combination of operating system, database platform, Web server and clustering technologies.

"State of the Union"

It's probably safe to say that most CF developers start out using session-scoped variables to manage client state. Session variables are fast and easy to use. In addition, there seems to have always been some confusion over the differences between client and session variables. Since session variables are most often discussed in the newsgroups, and are most frequently found in the various code examples in help screens, newsgroups and books, the implementation of session variables can be considered the most common method of client state management for new to intermediate CF developers.

Furthermore, it's also safe to say that "newbie" CF developers have no inherent need to work with any other method of client state management, since they're working primarily within a single-server environment. Since a single-server development or production platform poses no problems for developers using session variables for client state management, the session scope is an excellent choice.

A Clustering "Primer"

There are times, however, when using session variables can get you into trouble – specifically when running the application in a clustered-server environment. Before I address this problem, let's talk about what a clustered server environment really is. While an in-depth discussion of clustering is beyond the scope of this article, I will offer this brief "primer."



Configuring a cluster involves installing either hardware or software into the network infrastructure that will load-balance a Web site or Web application across multiple servers. This provides scalability: as usage of a Web site or Web application grows and server traffic increases, more horsepower can easily be added to the mix to compensate.

This article considers a specific and common configuration: ColdFusion 4+ with BrightTiger's ClusterCATS. In the CF/ClusterCATS configuration, the ClusterCATS software component connects two or more ColdFusion Web servers together, allowing them to share the work of processing and serving up Web pages. It also provides an element of fault tolerance; should one of the machines become unavailable (because of a hardware, software or network failure), the "load" is shifted to the remaining machine(s). All of this clustering functionality is (or should be!) transparent to the users.

There are two primary means of balancing the load across a cluster: "session-sensitive" and "true" load balancing. Session-sensitive load balancing occurs when a visitor is initially load-balanced to a particular server based upon traffic at that particular moment, then remains on that server for the duration of his or her session – even if another server in the cluster eventually carries a lighter load (see Figure 1). True load balancing occurs when, at any time during the session, a visitor may be receiving files from any server in the cluster. This has the advantage of balancing the load actively, using all servers in the cluster at all times for all visitors (see Figure 2).

ClusterCATS, as well as other clustering technologies, can function in both the true and session-sensitive modes I've described. With ClusterCATS it's as easy as changing one checkbox.

Some of you may disagree with my use of the somewhat biased term *true load balancing*. Your own experience may be that session-sensitive load balancing is just as good a choice

when it comes to ease of use and performance.

I agree with you, up to a point: with a high-volume site, a session-sensitive clustering configuration might "all average out" and provide good performance for all the users. At any given moment a few servers may actually have more load than others, because more users have started a session and remained on a particular server longer. However, with a great many users, it might balance out in the end, without any noticeable performance issues. But there's no guarantee.

The Problem with Session-Scoped Variables in the Cluster

The heart of the problem is this: with each shift to a different server in a cluster, CF session variables are "lost." For example, if I log in to your application or Web site and end up on server `www1` and you have set a session variable called "Session.IsLoggedIn" to keep track of me, that variable will be stored in the memory of the server `www1`. Since there's no central memory repository available to CF, if at any time during my session I'm load-balanced to server `www2`, that session variable will be lost because it doesn't exist in the memory of server `www2`.

As far as `www2` is concerned, I never logged in. Most likely, your application will deal with me just like any other user who's trying to load a page that requires a login: it'll redirect me to a login screen instead of showing me the page. But as a user who already logged in when I was on `www1`, I won't be at all pleased. To put it mildly, this isn't good. Leave such a problem unresolved and your users will probably want to know where you live!

You might ask, "So what's the big deal?" Why not just use session-sensitive load balancing and keep the user on a particular server, with all their session variables stored in *that* server's memory for the length of their session? Wouldn't that be easiest?

Here's where I stand by the term *true load balancing* and can show it to be a better configuration. Session-sensitive load balancing has one very real pitfall that occurs when a user has been load-balanced to a particular server and has established a session there, but that server then becomes unavailable. Should that happen, the user will inevitably be shifted to another server in the cluster, with the added hassle of losing their session variables in the process. You'll need to change your application code to resolve this problem, so why not use the more robust (true) load-balancing solution as well?

As you may have guessed by now, I'm no advocate of session-sensitive load balancing. I think of it this way: the expense of hardware, software and configuration of a server cluster can be justified only by gains in performance and failover protection; session-sensitive load balancing can potentially be an obstacle to both of those goals, while true load balancing actively involves all of the servers in your cluster on a continuing basis, fully achieving both of those goals. Most important, use of session-sensitive load balancing inappropriately encourages the use of session variables, which will quickly detract from the user-transparent failover abilities of the clustering solution.

The bottom line? With either method of load balancing, use of session variables is unacceptable. In reality, it doesn't matter which load-balancing solution you choose – as long as you don't use session variables, either one will work. However, it's my belief that if you really want to gain the full benefits of server clustering, you'll want to compose your ColdFusion application to support true load-balancing, while still being able to maintain client state. That's what this article is about.

Using Client-Scoped Variables for Application Scalability

Since session variables can only be stored in server RAM and since application and server variables aren't user-specific, we

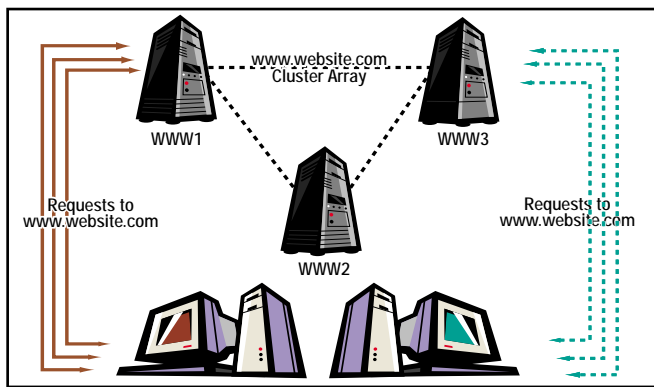


FIGURE 1: Session-sensitive load balancing

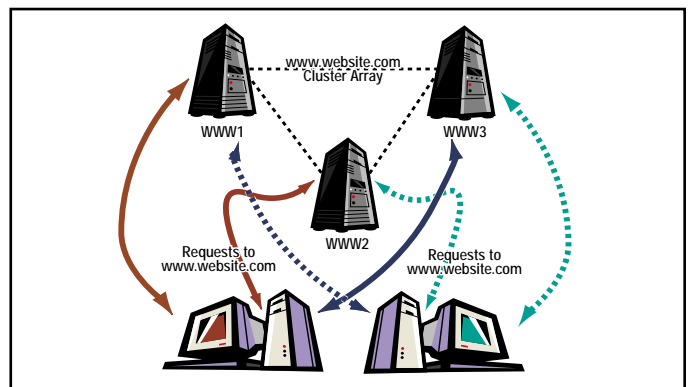


FIGURE 2: True load balancing

can't use those scopes to maintain client state in a load-balanced environment. We must therefore turn to client-scoped variables to solve this problem. More specifically, these client-scoped variables need to be stored in a specially designated database as opposed to the server registry (the default option for client variables). In this article I'm going to provide the systematic process for designing or converting a CF application to run on a Web server cluster, using database-stored client-scoped variables.

Configuration Outline

There are three basic areas you'll need to address in order to implement database-stored client-scoped variables within your application. The three steps are as follows:

1. **Database configuration:** Since all members of your ColdFusion Web cluster can "see" your database server cluster, this is where your client variables will be stored. You'll create a database on your server for use as the client variable repository (CVR).
2. **CF server(s) configuration:** All the servers in your cluster will need to be configured *not* to store client variables in the registry, but to use instead the database you'll have configured in Step 1. This involves using the ColdFusion Administrator to create an ODBC data source and the connection to your CVR. I'll also describe in detail how to disable the session-aware load-balancing feature within ClusterCATS.
3. **Application design/conversion:** If you'll be developing a new application, this step will help you design your application to support server clustering right from the start. If you have an existing application, this step will describe the changes that are required to make your application "clustering friendly."

If your application is already running in some form and you're tasked with planning or implementing a conversion, I would strongly recommend that you verify that all current features of your application are in working order prior to beginning this process. If you know everything worked before you started, you'll have a much easier time determining what problems may have surfaced as a result of the conversion process.

STEP 1: Database Configuration (MS SQL Server 7, Enterprise Manager)

This is a very easy step. It basically involves creating an empty database to run in parallel with the database(s) currently being accessed by your CF site. Storing client variables in tables of your existing database isn't recommended or supported.

You'll need to know the name of the SQL server that's being made available to the CF cluster and you'll need the appropriate permissions to create a new database on that SQL server. Configuring access to your SQL server via Enterprise Manager is beyond the scope of this article – you'll need to refer to your SQL server Books Online or your DBA for assistance.

Here is how to complete Step 1:

- a. Start Enterprise Manager (usually found by clicking Start | Programs | Microsoft SQL Server 7.0 | Enterprise Manager).
- b. Drill down through your configured SQL server groups and your chosen SQL Server until you see the Databases folder.
- c. Right-click on the Databases folder, and choose New Database. The Database Properties window will appear.
- d. Keep all default values in this window, and enter a name for your new database (in this example, "CVR"). You can call this database anything you wish and create additional separate databases for each application that will be using DB-stored client variables. Click OK to create the database.
- e. You can configure additional options for this database later, including a regular backup/maintenance plan. (That's a nice bonus!)
- f. Don't add any tables or other objects to this database until you've completed all the steps in this document. The required objects will be automatically created in later steps, after which you may add additional objects such as stored procedures, views or additional tables. However, I do recommend that you keep this database optimized and try your best to minimize the overhead created by adding too many objects or functions.

STEP 2: Configuration of ColdFusion Server

Note: Processes 2.1 and 2.2 will need to be carried out on each server in your cluster – with the exception of step (e) in process 2.2. By following these instructions, you'll configure your servers to use the database created in Step 1 above to store client variable data.

2.1: Create the data source.

For this step you'll need full access to the ColdFusion Administrator application on each server in the cluster. Here's what to do in detail:

- a. Start the ColdFusion Administrator (usually found by clicking Start | Programs | ColdFusion Server 4.x | ColdFusion Administrator).
- b. Log in using the password for administrator access that you chose when CF Server was installed.
- c. Click on ODBC under the Data Sources section on the left sidebar.
- d. Type the data source name ("CVR" in this example), choose the ODBC Driver ("Microsoft SQL Server Driver" in this example) and click the Add button.
- e. Enter the name or IP address of the database server in the Server field.
- f. In the database field enter the name of the database you created in Step 1 ("CVR" in this example).
- g. Click the CF Settings button and complete the username and password fields to access this database.
- h. Make sure Maintain Database Connections is checked.
- i. Click the Create button. After the data source is created you

should see your list of data sources again, with “verified” next to the data source you just created. If you don’t see “verified,” you must take whatever steps are necessary until this server is able to connect to the data source. The connection must be verified before you can continue.

- **2.2: Configure client variable storage.**

Now that you’ve created the data source, you will configure the server(s) to use that data source to store client variables. Here’s how to do so in detail:

- Under the Server category in the left sidebar, click Variables.
- You will see a select box at the top, under Name. Choose the data source you created above (“CVR” in this example) from the select box, and click Add.
- On the page that loads next, choose the number of days you wish to have client variables stored in your database. The default is 10 days.
- For higher-traffic sites, you may wish to select Disable Global Client Variable Updates. In the beginning, try leaving them enabled; if performance suffers, you can place a check in this box later. Keep in mind that if you write code that depends upon the automatic updating of such client variables as “LVISIT” (the last date/time the client with that CFID loaded a page) and “HITCOUNT” (the number of page impressions by a particular client), your code will break if you disable global updates later.
- (*This is a very important step.*) If this is the first server you are configuring, you need to leave the checkmark next to Create Client Database Tables. This will create the necessary objects in the empty database you created in Step 1. However, if this is *not* the first server, you must remove the checkmark and turn this feature off; otherwise you will receive an SQL error indicating that the tables already exist and therefore could not be created. If you receive this error, your configuration will fail.
- Click on Create. Your Client Storage area will appear in the list on the resulting page.
- Repeat these steps for every server in your cluster, noting the exception in step (e) above.

- **2.3: Configure ClusterCATS.**

This is not by any means an in-depth discussion on the setup and configuration of ClusterCATS; it’s only a specific set of steps to accomplish the goal of enabling true load balancing within the cluster. If you’re using a different clustering solution, contact your vendor or system administrator for help in finding a similar setting.

The detailed steps are as follows:

- Open the ClusterCATS explorer (usually found by clicking Start | Programs | BrightTiger | ClusterCATS Explorer).
- In the left pane you’ll see Cluster Manager. Beneath the Cluster Manager you should see your configured cluster(s).
- Right-click on the cluster your application is running under and choose Properties.
- Click the Administration tab at the top of the window that appears.
- Under the section labeled Session State Management, turn off Enable Session Aware Load Management, then click OK (see Figure 3).

Your cluster will now serve pages from any server at any time, rather than “parking” users on a particular server for their entire session.

The list below references other server settings that you may wish to adjust, depending upon your situation:

- 1. Default Client Variable Storage (Server category, Variables page):** The default is Registry. If you want all the Client variables for all applications to use the same central database, change this from

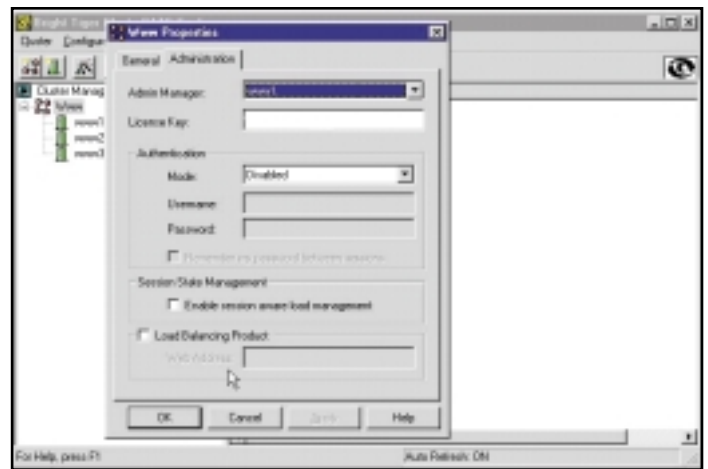


FIGURE 3 Disabling session-aware load management in ClusterCATS

Registry to the data source you created (CVR). If you choose to leave this as Registry, you must make sure all of your CFAPPLICATION tags specify the CLIENTSTORAGE attribute (see next section). The registry, just like server RAM, is specific to each server, making it inappropriate for storing client variables in a cluster.

- 2. Session and Application Variables (Server category, Variables page):** You may wish to disable session and application variables, so that if you accidentally write them into your application, you’ll receive an error rather than allowing the application to function. Since session and application variables are not centrally stored, if you should accidentally use session and application variables, your application will function on the cluster but with undesirable results. Disallowing session and application variables adds another level of protection.

STEP 3: Application Design Conversion

This will probably be the most challenging step. Since I can’t be there to help you with your code, I can’t provide screenshots and other helpful tools to aid you in the design or conversion of your application to use client variables. However, there are some basic bits of code that you’ll need to use, starting with the CFAPPLICATION tag.

The CFAPPLICATION Tag

This is an important setting. In every Application.cfm in your application (if you are using Fusebox, chances are this setting will exist in your “app_Globals.cfm” file), you should have a CFAPPLICATION tag. You need to make sure that you have enabled client variables for your application, and turned off the ability to use session variables in each instance of the tag. A sample CFAPPLICATION tag, using our database created above as an example, would look like this:

```
<CFAPPLICATION
  NAME="YourAppName"
  CLIENTMANAGEMENT="Yes"
  SETCLIENTCOOKIES="Yes"
  CLIENTSTORAGE="CVR">
```

Make sure you specify a separate name for each application running on your cluster – the CVR database uses that name to keep client variables unique, should you choose to store all client variables for all applications in one database (i.e., using the Default Client Variable Storage location in the ColdFusion Administrator).

It remains a matter of choice whether or not you will enable “SETCLIENTCOOKIES” – the use of cookies makes coding easier

Do You Know Where Your Users Are?

Managing "client state"

Most CF programmers are eventually faced with the issue of client state management. Simply put, the stateless nature of the Internet (discussed in so many documents on the Internet) lends itself poorly to developers wishing to keep track of what the user is up to "right this moment." Nonetheless, we as developers do still need to know that a certain user is logged in, for example, before that user is allowed to access certain pages/functions within our application.

In that example we need to know that the user is logged in without having to write a great deal of additional code. Both the session and the client variable scopes available in ColdFusion allow us to set user-specific variables that are consistent across the pages of our application by simply calling the variable.

because you don't need to pass CFID and CFTOKEN in the URL throughout your application. Passing CFID and CFTOKEN in the URL presents a couple of issues:

- You could get a potential problem with a user trying to "jump sessions" by hacking different numbers into the CFID/CFTOKEN variables.
- CFID and CFTOKEN can cause problems when someone wishes to bookmark a page in your application.

Personally, I require cookies on all of my sites and then provide adequate security and cookie information to the users who are refusing or unable to accept cookies so they know why we require them. I prefer setting CFID and CFTOKEN to a "session-only" cookie so that the session is destroyed when users close their browsers:

```
<CFCOOKIE NAME="CFID" VALUE="#CFID#">
<CFCOOKIE NAME="CFTOKEN" VALUE="#CFTOKEN#">
```

I then detect whether users are able/willing to accept cookies, and redirect them to an information page if they are not, using JavaScript:

```
<SCRIPT>
document.cookie = 'CookiesEnabled=1';
if (document.cookie == ''){
document.location.href='myCookieInfoPage.htm';}
</SCRIPT>
```

You can choose any method you like to detect whether browser cookies are enabled. (First you'll need to decide for yourself whether or not cookies should be a function of your application, based upon your user demographics and other factors.)

Notice that in the CFAPPLICATION tag above, we have set the value for CLIENTSTORAGE equal to the data source name that we configured in Step 1. If this is *not* specified, and any one of the servers in the cluster is not configured to use your database as the Default Client Variable Storage location, then you may get unexpected results because your client variables will end up someplace other than in your database (most likely the registry, which is the default location). In my opinion, it's safest to distinctly specify the CLIENTSTORAGE value.

Existing Session Variable Code

If you're designing a new application, simply set and retrieve all the variables that you would have scoped as session, using the client scope instead. Some examples:

```
<CFSET Client.IsLoggedIn = 1>
```

```
<CFOUTPUT>#Client.IsLoggedIn</CFOUTPUT>
```

```
<CFIF NOT Client.IsLoggedIn>
    <CFLLOCATION
        URL="notloggedin.cfm"
        addtoken="no">
</CFIF>
```

If you're converting an existing application – back up your application first please! – perform a global, case-insensitive find (notice I did not say *replace*!) for all occurrences of "Session." in all of your ColdFusion templates within your application. Inspect each occurrence to make sure that the text string "Session." is actually referring to a session variable, making notes of the exceptions. (For example, a sentence on your template might read "Click Log Out to terminate your session." With a global replace, this would end up reading "Click Log Out to terminate your Client.") While this process can be tedious (checking all the session variables in your application before replacing them all could mean opening almost every page in the application), it will be rewarding knowing exactly what has been changed when you do finally perform the global replace.

After making note of the exceptions (if any), perform a global replace, replacing all occurrences of "Session." with "Client." Return to the pages where you noted exceptions and change them back to the proper text.

Managing Session Timeout

Client variables are set to expire after a certain number of days. This may not be appropriate for certain variables of your application. For example, perhaps your session variable for "IsLoggedIn" used to time out after 20 minutes because it was a session variable. If users closed their browsers without logging out (a common problem), you couldn't guarantee that they were logged out, but at least you'd know that eventually their session variable would expire after a reasonable time. In contrast, the default client variable setting would only remove the client variable after 10 days – they could go to your site or application and be logged in already, up to 10 days after they first logged in!

If this is inappropriate for your application, there are a several options. In one application I designed, I actually ran two additional "hidden frames" in addition to the pages of the application. One loaded a page that kept the session alive using a META refresh tag (for user convenience – so they wouldn't time out as long as they kept their application open). The other frame loaded a page that contained some JavaScript. The script executed a logout routine and logged users out when they left the site or closed their browsers ("onUnload"). This is a very reliable method if you have some control over the user environment (this particular application ran on an intranet where Internet Explorer 5 with JavaScript and cookies was a requirement for the organization).

Assuming that you don't have some kind of cosmic power over the browser and settings your site visitor chooses to use, you'll need to find another solution. The easiest one I can provide requires that either the Global Client Updates (Step 2.2 (d), above) is on, or that you are updating the client LVISIT data within strategic parts of your application. This way, if a certain period of time has elapsed (30 minutes in these examples) beyond their last page activity, the user is logged out by destroying their client variables (see Listing 1).

The code in Listing 1 needs to be run at every page impression (usually included in app_Globals.cfm or Application.cfm) or at least at points where you wish to ensure that the user has remained "active." If you want to optimize this code, you can create a stored

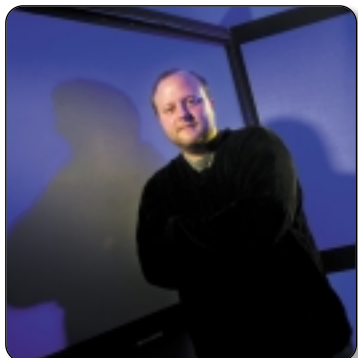
XML DevCon 2000

www.xmldevon2000.com

XML DevCon 2000

www.xmldevon2000.com

Mapping the Future of CF, Spectra, E-Business and the Internet



JEREMY ALLAIRE, Cofounder and CTO of Allaire Corporation, Inc., shares his vision

AN INTERVIEW BY AJIT SAGAR

CFDJ: Allaire has recently been in the acquisition and partnership mode. Can you give us a brief history of the events over the last couple of years and the rationale behind these decisions?

Allaire: One of the primary reasons for going public over a year ago was to create a base of assets that could be leveraged into mergers and acquisitions. We saw a number of areas for expansion for our platform, all of which underlined a long-term strategy for becoming one of the leading Internet business platforms. In 1999 we undertook three acquisitions. The first, BrightTiger Technologies, makers of advanced scalability and Web systems management technology, underpinned our efforts to bolster our Enterprise-level application server offering. The last two, LiveSoftware and Valto Systems, helped to accelerate Allaire's entry into the Java server marketplace, and have formed the foundation for our next-generation products. As you know, both LiveSoftware and Valto were 100% focused on pure, standards-based Java architectures, including JSP, Servlets, EJB, JTA and JMS. Unlike a lot of other players in the Java server field who were carrying forward their own proprietary Java offerings, we saw that the end of 1999 was the time for a pure-play, standards offering.

CFDJ: You also recently acquired a commerce software vendor, OpenSesame, makers of a profiling and personalization solution. Most application server vendors are making similar moves. Cases in point: Commerce Server (Sun-AOL's iPlanet product suite) and WebLogic's Commerce Xpert. How do you plan to compete with such vendors?

Allaire: The OpenSesame acquisition expands on the Allaire Spectra foundation, which is our packaged-applications product line. Spectra 1.0 has been available since December and is having fantastic success out of the gate as a solu-

tion for content and commerce sites. As articulated in our roadmap, this product line is expanding with additional application modules focused on the needs of running an Internet business.

CFDJ: How do these developments affect ColdFusion developers? Do they now have to learn Java to fully leverage the Allaire product suite?

Allaire: CF developers definitely won't have to learn Java, though the CFML runtime will be built on top of a Java foundation. There will be a lot of benefits that flow out of that, including that the Allaire server platform will be able to handle pretty much every type of project in any company. For CF developers you'll see us enhancing CFML with more object-based constructs, and the CF server itself will include more advanced scalability and reliability features based on the transaction and messaging server infrastructure in our core Java platform.



Allaire's new headquarters will be located at Riverside Center in Newton, Massachusetts.

CFDJ: It seems to me that this rapid evolution has confused the marketplace. Is there a uniform message Allaire wants to send out to the Web community?

Allaire: A lot has happened with Allaire in the last year. We've evolved from being a leading provider of tools and application servers to supplying a comprehensive Internet software platform covering core server infrastructure, packaged applications as well as development and productivity tools. If there's a uniform message for the Web community, it's that Allaire intends to be a dominant provider of Internet software platforms, enabling any organization to successfully build their business on

the Web. We intend to provide top-to-bottom platform infrastructure as well as a wide range of horizontal packaged applications necessary to running an Internet business.

CFDJ: Going back to my first question, how does this message map to the products you are offering?

Allaire: In our recently released Allaire Technology Roadmap, we discuss the customer requirements for e-business software infrastructure and then map this to our existing products and platform, as well as provide a roadmap for future Allaire offerings. Today we're providing a very robust set of application server products, covering both front-tier and back-end application server requirements. This goes from free products for small dynamic sites up through enterprise offerings, including distributed transaction and message queuing infrastructure. On the packaged applications front we released Allaire Spectra 1.0 in December, providing a comprehensive suite of modules for content management, e-commerce and customer relationship management on the Web. We also announced the acquisition of OpenSesame, an early pioneer and leader in the customer profiling, analysis and personalization space.

CFDJ: Allaire started out with site-building software that has now evolved into an industry-strength application server. How would you define an application server in today's market?

Allaire: Again, we discuss this in great detail in our Technology Roadmap. It's our belief that the application server portion of e-business software infrastructure is rapidly becoming something of an "operating platform" for all Web applications, replacing the role that was once provided by classic middleware and network operating systems. Functionally, application servers really provide four key areas of functionality: Core Services, such as clustering, security, session management and logging; Integration Services, such as connectivity to Internet protocols, ERP systems, database systems, directories and so on; Web Application Logic services, providing a dynam-

ic page environment that's used for scripting and is generally the basis for about 80% of applications built; and finally Business Logic services, providing a container or hosting environment for complex business logic, executing in a distributed, transactable fashion. This will certainly be the minimum requirements for any offering into 2000 and 2001.

CFD: Nowadays, in order to get wide acceptance in the computer industry, companies try to adhere to technology standards. Does Allaire plan to standardize any of its technologies so as to get a larger audience?

Allaire: Broad adoption of any platform requires a unique combination of proprietary innovation and open standards. This has certainly been the case in the Internet world. For Allaire, this means building and supplying infrastructure based on Internet and industry standards, innovating beyond standards in territories that aren't developed and in turn collaborating to ensure that that innovation eventually contributes to open standards efforts. Our efforts in the Servlet and JSP community are indicative. The JRun team has consistently extended what's possible with server-side Java and have been aggressive to contribute that work to the Sun Community Process. Likewise, with XML protocols Allaire is committed to a similar effort.

CFD: How about ColdFusion? Do you plan to submit CFML to a committee for a standards process?

Allaire: We don't plan on submitting CFML proper to a standards process. However, as will be evident in the marketplace in 2000, we're very actively involved in defining the Tag Extensibility mechanism in the JavaServer Pages standard and you'll start to see some de facto standardization for the ColdFusion language model through this kind of process. Over time, CFML will become a suite of language modules built on top of a standards-based architecture.

CFD: What's your roadmap for the coming year regarding ColdFusion?

Allaire: We've got a ton of things in the works with ColdFusion and our server platform in general. The biggest move will be the shift of CF to be built on the J2EE platform, which will spawn all kinds of benefits for developers. During the coming year or so, you'll also see some of the programming framework aspects of Allaire Spectra, such as the ContentObject API and the ContentObject Database, become generally and freely available to ColdFusion

developers. And then there's the obvious stuff, the incremental enhancements to core tags and server features.

CFD: Does Allaire plan to get into application design or do you plan to always be application enablers?

Allaire: I assume by this you mean actually getting into the application development business through a consulting organization. This isn't a big focus for the company, though we're expanding our services offerings to better ensure the success of our enterprise customers.

“
Allaire has been
one of the most
active proponents of
the standardization
to tag-based
scripting within JSP

CFD: How does your entry into the Java middle tier affect the company's direction vis-à-vis ColdFusion?

Allaire: The easiest way to think about this is that Allaire is broadening its role in the Internet software platform market to cover all of the tiers of technology necessary for building an Internet business. Historically, Allaire hadn't supplied core object middleware, such as transaction and messaging infrastructure, and it became very clear that for Allaire to become a true platform company, these were aspects of the broader solution. For our ColdFusion customers, in essence, this means that the CF server will become even more strategic to their overall Internet infrastructure. At the same time, of course, we're deeply committed to supplying server software for every tier of the market, ranging from free, low-end servers like ColdFusion Express up to the highest-end servers necessary for running a company's Internet back office.

CFD: Don't Sun's technologies for the client, such as JSP and Servlets, overlap with your own, such as CFML? After all, both are mechanisms for producing dynamic HTML.

Allaire: CFML, JSP and ASP all provide mechanisms for delivering dynamic pages to browsers.

I wouldn't put Servlets proper in this category, as they're really a Java-focused alternative to CGI or NSAPI/ISAPI and really don't provide a scripting and page template environment that's necessary for Web applications.

Currently, CFML provides the highest level of abstraction and therefore the highest degree of productivity of developing interactive applications. Both JSP and ASP are still at the object-scripting level and don't yet provide the same kind of productivity advantage, though they're ages ahead of writing server-side interactivity logic using Java or C. As you may be aware, Allaire has been one of the most active proponents of the standardization of tag-based scripting within JSP, contributing to the Taglib architecture and working actively on reference implementations as well as next-generation architectures. As we move ColdFusion to a J2EE foundation, you'll also see us move CFML to sit as an abstraction on top of JSP and Servlets.

CFD: Your recent acquisition of Valto Systems clearly broadcasts the message that you want to compete with the big boys of Java enterprise computing, such as BEA Systems, Sun and IBM. How do you plan to gain entry into an already defined market?

Allaire: We expect to become a major supplier in the enterprise tier of the Java application server market and I think you'll see a very different approach from Allaire in pursuing that space. First, our entire offering is based 100% on Java and 100% on J2EE implemented services. Unlike many of our competitors who have existing C++, Java and CORBA-based systems that are migrating to J2EE, everything we've done has been built from the ground up on Java2. This means that we've got an exceptionally lightweight, clean-room implementation that comes in at around a 1MB memory footprint. We're going to leverage that into ISV and OEM customers, who are increasingly looking for embedded J2EE services. We're also going to continue the Allaire tradition of pricing and packaging our platforms for mass adoption – so I think that, from a developer's perspective, the Allaire J2EE offering will be the most competitive in the marketplace in terms of price/performance.

CFD: I've heard from several folks that the ColdFusion Application Server can't handle high-volume transactions. Is this true? If so, how are you addressing the issue in your migration to the Java middle tier?

Allaire: In early 1998 that was certainly true. In late 1998 and throughout 1999 we delivered

ColdFusion Enterprise, our first enterprise-level application server offering. This included all of the components you would expect at this tier, including advanced scalability and clustering services, native database connectivity, CORBA and Java support, and a lot of server tuning, caching and configuration options. With this offering we were able to support dozens of large dot-com customers handling millions or tens of millions per day, and had a couple of the top 10 holiday e-commerce sites deployed entirely on our platform. In addition, our move into the transaction middleware space will ensure failsafe deployment and reliability for customers implementing large-scale transactional systems.

CFDJ: Would you say that ColdFusion is a direct competitor to JSPs? Is there room for both? If so, could you give me examples of the kinds of applications that could leverage both these technologies?

Allaire: Again, today you really would choose one or the other for the Web application logic layer. Going forward, with CFML built on a Java foundation, it will be incorporated into the same runtime environment as JSP. What it will really come down to for a developer is what level of abstraction they want or need to work at. CFML is simply the fastest way to develop dynamic Web applications, and we will carry this tradition forward into the standards-based community.

CFDJ: Is it fair to say that ColdFusion can provide an abstraction layer on top of JSPs, ASPs, JavaScript and other scripting environments? Can ColdFusion help developers stay agnostic about scripting environments?

Allaire: ColdFusion – or CFML, more specifically – provides the highest abstraction layer on top of any server-side or client-side language. In particular, CFML Custom Tags enable a developer to encapsulate pretty much any client or server-side logic in a straightforward, declarative component. As we broaden our server platform to include JSP, JavaScript and so on, you'll see that Allaire is deeply committed to providing a multilanguage platform, not just CFML.

CFDJ: How would a company providing e-commerce solutions partition their project so that they can leverage your different products? Would they be forced to use all of your products?

Allaire: There is obviously a broad range of configurations and uses of our platform, ranging from using ColdFusion Express for a small-scale departmental intranet up to using all of our

server products and our applications to implement a comprehensive, enterprise-wide e-business initiative. One approach we're starting to see goes something like this: the front-end Web site, including content management, user interactivity, shopping cart functions and so on, all based on ColdFusion and Spectra, with back-end components (with greater degrees of complexity) based on server-side Java, using either Servlets or EJBs running in JRun.

“
We rewrote a lot of our integration components for protocols such as SMTP, POP and HTTP, providing a lot of incremental enhancements customers wanted

CFDJ: You made some really big announcements with Spectra. How's the evolution of that product coming along?

Allaire: It's going fantastic. Relative to our competitive peer group (e.g., BroadVision, Vignette, ATG and Open Market), we're overnight becoming a major player in the packaged applications marketplace. We built a highly competitive product with a very broad feature set and leveraged the heck out of the technology foundation already established with ColdFusion. As a 1.0 release, we're very pleased with the success that the product is seeing, and are well on our way to incremental releases as well as a major point release later this year.

CFDJ: Do you have any examples of real-world implementations using Spectra?

Allaire: Yeah, definitely. We're just starting to see the first wave of sites come online. A number of these are vertical portals, such as Vetricentric.com, Planetmedica.co.uk, OperationBass.com, NetRadio.com, among others. We've signed up over 300 major customers and we'll be seeing some major e-commerce sites and enterprise portals come live over the summer.

CFDJ: Will all ColdFusion developers have to gain an understanding of Spectra and your other commerce products?

Allaire: No, definitely not. A similar parallel can

be found with Microsoft or Oracle. For example, thousands of customers use the Oracle database server but never purchase or implement Oracle's applications. Likewise, millions of customers use Windows but many don't use Microsoft Office. Over time, we expect to see more and more customers purchasing application modules from Allaire that run on our platform. Certainly, with a customer standardizing on the Allaire Business Platform, they'll most likely be using all of our software products.

CFDJ: How can our readers start playing around with some of your new products?

Allaire: Everything is downloadable from our Web site. Allaire Spectra, in fact, is included for free with ColdFusion Studio under a developer's license that never times out. JRun 3.0 may be out by the time this article runs but customers can get to the beta at beta.allaire.com/jrun30.

CFDJ: When do you think ColdFusion developers (and not Java developers) switch to CF 4.5? What are the advantages of doing so?

Allaire: There are gobs of advantages to moving to CF 4.5, the most obvious of which is that CF 4.5.1 is our first release on Linux. We rewrote a lot of our integration components for protocols such as SMTP, POP and HTTP, providing a lot of incremental enhancements customers wanted for a long time. There are dozens of new functions and a wide range of scalability-related features, including full-blown service-level failover and integration with local-director. We added support for BLOBs and binary data in WDDX and also introduced a huge range of Java integration and extensibility options.

CFDJ: Don't you think that with the new directives that Allaire has undertaken, the name of this magazine should be changed to something like *Allaire Developer's Journal*? After all, you don't do just ColdFusion anymore!

Allaire: I haven't thought about this too much. We definitely want to stay focused on ColdFusion as a developer technology and platform, in addition to our other products and technology. Perhaps we could create an *Allaire Systems Journal* focused on our overall platform, but you'd still see journals focused on specific tools and technology, just as there are focused journals for Visual Basic and so on. Most importantly, what do the readers and customers think?

Ajit Sagar, A frequent contributor to *CFDJ* and *Java Developer's Journal*, is also editor-in-chief of *XML Journal*.

Allaire

www.allaire.com

MANY HAPPY RETURNS

BY JOHN MORGAN

THE ART OF CREATING

From my very first line of ColdFusion code I was in heaven. There were lots of powerful tags and functions that made Web site creation a dream. What more could I ask for? User-Defined Functions (UDFs), that's what!

I was surprised that a language that was so extensible omitted the ability to create functions. I knew that if CF was going to be all that it could be, I needed to find a way to synthesize a UDF.

Why Bother with UDFs?

Don't get me wrong, ColdFusion has many built-in functions – over 200 at the last count. So why would you ever need to define your own function? Well, take for example two very useful functions, Ucase() and Lcase(). These convert a string to uppercase and lowercase, respectively. But what if you want to convert to proper case? CF doesn't have a Proper() function. This calls for a UDF.

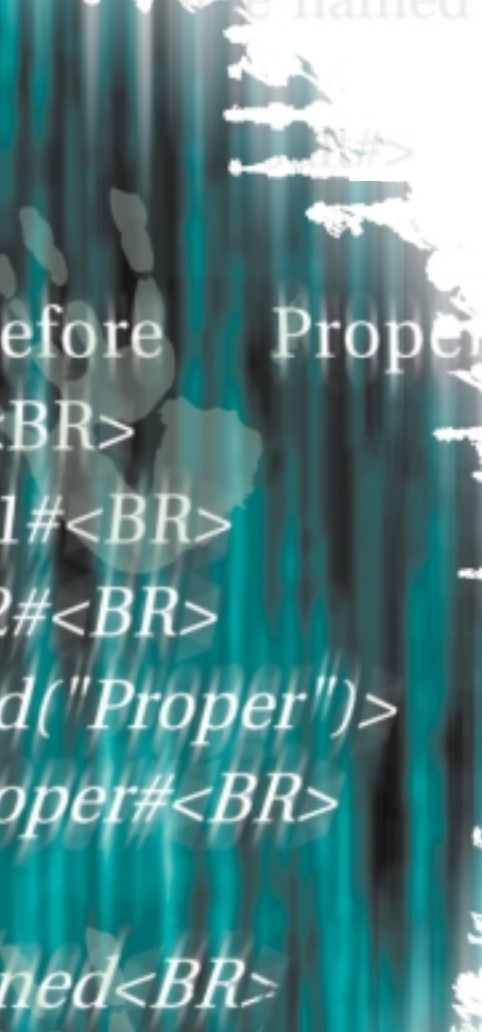
So What Is a UDF?

Before we can create a UDF we need to define what a UDF is: it's a subroutine that can receive one or more parameters and return a meaningful result. Technically, a function or UDF could have no parameters whatsoever (e.g., The Now() function). Regardless of how many parameters a UDF takes, its power is gauged by the meaningful result.

The only "subroutine" that CF has is

```
<CFOUTPUT>
  <B>B
  ProperBR</B><
  TEST1 = #Test1
  TEST2 = #Test2
  <CFIF IsDefined
    PROPER = #Pro
  <CFELSE>
    Proper is Undefi
  </CFIF>
  <HR>
</CFOUTPUT>
```





Listing 1). The attributes are the collection of parameters that are passed into the custom tag. For each parameter passed, you'll have one similarly named item in the attributes collection. The example UDF gets a string passed via `Attributes.PARAM1`. The `<CFSET>` tag takes the string stored in `Attributes.PARAM1` and concatenates asterisks to both ends of it.

Next, we need to return the result. We'll need to create a variable that's at the same scope as the calling template. We use the Caller scope to create the variable. When you use the Caller scope it's as if the `<CFSET>` tag was in the template that called the UDF. Once created, the calling template can use the value in the newly created BasicUDF local variable like any regular local variable. I named the new variable BasicUDF to match the name of my function but this isn't a requirement, just a personal convention.

You might be wondering why the calling template couldn't just read the value of the Result variable. Well, the "calling" template can't even access the Result variable. The Result variable is local to the custom tag and is therefore "out of scope."

makes for an easy demonstration. One important note: if you want to retain the value stored in the BasicUDF variable, you need to copy it to another variable before calling `<CF_BasicUDF>` a second time or the value will be overwritten.

Localizing Variables

I've found that my code is easier to read if I localize my parameters. I do this by copying the values of the attribute parameters into local variables. To do this, I added a line to copy the `Attributes.PARAM1` parameter into the local variable "par_Param1" (see Listing 2). You might have noticed that I named the local variable "par_Param1". This lets me know at a glance that I'm dealing with a parameter value without my having to type "Attributes" over and over again. I also changed the line that sets the Result variable to use the new par_Param1 variable.

Passing by Value

The way we passed the parameter is called *passing by value*. That means the value being passed is copied into `Attributes.PARAM1`. Since we're dealing with a copy of the original value, any changes we make do not affect the original value. Depending on what the function does this may be just fine, but if the function needs to modify the original variable, we need to pass it by reference.

Passing by Reference

Passing by reference is when a reference (usually via a pointer à la C/C++) to the original value is passed to the function so that the function can change the original value. In reality, ColdFusion has no provision for passing by reference (as there are no pointers in CF) but we can pass the name of a variable. If we know the name of a variable, then we can write code to modify it. But how can we write code that uses the variable name if we won't know what it is until the template is executed?

Reading the Values

There's a built-in function named "Evaluate" that will resolve the string passed to it for any CF functions or pound(#) delimited variable names. It then takes that result and evaluates it one last time for a final result. Take for example this line of code:

```
<CFSET RoomID = 2>
<CFSET RoomName = Evaluate("Classroom#RoomID#")>
```

The above Evaluate function would resolve "RoomID" to the value "2," which would result in the string "Classroom2". The resulting string would be evaluated

the custom tag. Can a custom tag act as a function? The answer is yes – the manual even drops hints about this ability. Custom tags meet all the requirements: they can receive any number of parameters and they can return a meaningful result. It's the result part that's a bit tricky.

A Very Basic UDF

Let's pick apart a very basic UDF (see

This is a good thing because it means that variables in the calling template are protected from the custom tag and vice versa.

Calling the UDF is simple if you use `<CF_BasicUDF PARAM1="Hello">` in another template. The result would be that a local variable named BasicUDF would have the string "Hello" assigned to it. You could then display or manipulate the "BasicUDF" variable as you see fit. Okay, so it's not very useful – but it

and the value stored in the “Classroom2” variable would be assigned to the “Room-Name” variable.

Writing the Values

Now we need a way to write to the variable passed by reference. ColdFusion has provided two ways to change a variable when we get the name dynamically.

The first is to use the built-in `SetVariable()` function. This takes two parameters: first, a string containing the name of the variable to modify; second, the new value to assign. If we had a variable named “VarRef” that had the value “MyVar” and we wanted to set MyVar’s value to “Hello” we could use this code:

```
<CFSET void = SetVariable(VarRef,
"Hello")>.
```

The second way is to use inline interpretation. Inline interpretation is when you use a string – not a string variable but a literal string – on the left side of the equal in a `<CFSET>` tag. The string gets resolved to a variable name and that variable receives the assignment. So to assign a value to the variable “named” in the MyVar variable, I would use this code:

```
<CFSET "#MyVar#" = "Hello">.
```

Listing 3 creates a tag called “CF_BasicUDF3” that demonstrates how to read and write variables when they’re passed by reference. The first `<CFSET>` tag reads the value of the string parameter. The second `<CFSET>` tag modifies the value and the third `<CFSET>` stores the modified value back to the “original” variable.

Checking the Parameters

A well-written function should never rely on the calling template to use the function correctly. Even the best programmers make mistakes like forgetting

to pass a parameter or misspelling a parameter name. It’s good practice to test if all the required parameters have been passed.

What would happen if I were to call `<CF_BasicUDF3>` without the `StrVarName` parameter? I’d get the message “Error resolving parameter ATTRIBUTES.STRVARNAME.” I need to test whether “PARAM1” exists and if not take appropriate action.

CF has two functions that can assist in this endeavor, `IsDefined()` and `ParameterExists()`. `IsDefined()` is the one we want to use; `ParameterExists()` is provided only for backward compatibility. The `IsDefined()` function takes one parameter, a string value set to the name of the variable that we want to test to see if it’s defined. We can add some code to test if “StrVarName” exists and if it doesn’t we can display a less cryptic error message (see Listing 4).

While this works, a better method would be to raise a ColdFusion custom error – but that would be beyond the scope of this article.

Setting Parameter Defaults

At some point you’ll most likely want to create a function that has parameters that are optional and/or have parameters with a default value. To achieve this you use the `<CFPARAM>` tag. This tag has two parameters: the first is “NAME”, the name of the parameter to assign the default value. The second parameter is “DEFAULT”, the default value to assign. The `<CFPARAM>` tag tests if the variable specified in the “NAME” parameter is defined. If it isn’t, the variable is created and assigned the value specified in the “DEFAULT” parameter. If the variable specified in the “NAME” is defined, the `<CFPARAM>` tag is ignored. To demonstrate this concept I modified the UDF to take a second parameter. The second parameter, “Symbol”, is the character to put around the string. I set the

default for “Symbol” to an asterisk (see Listing 5). Now, if I called the tag as follows:


```
<CF_BasicUDF5 StrVarName="Hello">
```

I will get “*Hello*” as the return result.

Making the Proper Functions

Pulling all this information together, I created two proper-case UDFs. The first one, `<CF_ProperBV>`, passes by value (see Listing 6). The second, `<CF_ProperBR>`, passes by reference (see Listing 7). I also created the `ProperTest.cfm` template (see Listing 8) to demonstrate how the functions are called and how they affect the local variables. Both UDFs are virtually (no pun intended) identical. They both check each character in the passed string, looking for characters that follow a space. Characters that follow a space are capitalized; those that don’t are lowercased. The resulting string is returned to the calling template. This is by no means the best way to write a proper case function. It doesn’t even take into account “Mc” versus “Mac.”

Return on the Investment

It may seem like lot of work to create a function in CF – and, well, it is. But a well-written UDF, like a well-written custom tag, will save hours in the long run. Maybe someday CF will add the ability to create “real” functions. Until that day comes, I hope that you’ll benefit from custom tag UDFs as much as I have. 

ABOUT THE AUTHOR

John Morgan writes courseware at Blue Star training when he's not busy training programmers, Web developers and database developers. He also speaks at conferences, workshops and the San Diego ColdFusion Users Group, which he hosts.

johnmorgan@bluestarcorp.com

LISTING 1: BasicUDF.cfm

```
<!-- Surround the string passed via Param1 with asterisks -->
<CFSET Result = "*" & Attributes.PARAM1 & "*"
<!-- Create a variable named BasicUDF and assign it the result -->
<CFSET Caller.BasicUDF = Result>
```

LISTING 2: BasicUDF2.cfm

```
<!-- Localize Variables -->
<CFSET par_Param1 = Attributes.PARAM1>
<!-- Surround the string passed via Param1 with asterisks -->
<CFSET Result = "*" & par_Param1 & "*"
<!-- Create a variable named BasicUDF2 and assign it the result -->
<CFSET Caller.BasicUDF2 = Result>
```

LISTING 3: BasicUDF3.cfm

```
<!-- Get the value of the parameter that was passed in Cap-
```

```
tion --->
<CFSET par_StrVar = Evaluate("Caller.#Attributes.StrVar-
Name#")>
<!-- Surround the string passed via Param1 with asterisks -->
<CFSET Result = "*" & par_StrVar & "*"
<!-- Assign the return value to the parameter that was
passed --->
<CFSET "Caller.#Attributes.StrVarName#" = "#Result#">
```

LISTING 4: BasicUDF4.cfm

```
<CFIF IsDefined("Attributes.StrVarName")>
<!-- Get the value of the parameter tha was passed in
Caption --->
<CFSET par_StrVar = Evaluate("Caller.#Attributes.StrVar-
Name#")>
<!-- Surround the string passed via Param1 with aster-
isks --->
<CFSET Result = "*" & par_StrVar & "*"
<!-- Assign the return value to the parameter that was
passed --->
```


JavaCon 2000

www.java2000.com

JavaCon 2000

www.java2000.com

REVIEWED BY
DAVID
SCHWARTZ



EZwidget

by Webdriver Corp.

Create a basic e-store quickly, easily and inexpensively

It seems everyone I meet lately wants to set up an online store. "It would be great if I could sell my handmade Barbie doll dresses or sell ties, direct, on the Web!" The e-store is hot and the fire is spreading.

Most people, even developers (yes, we're people too), lack the experience needed to create and maintain an online store, or the time and capital to invest in hiring a consulting company. Others want to test the e-waters before sinking thousands of e-bucks into a store.

If you're starting to feel like this review is being written for you, then EZwidget from Webdriver Corp. may be your e-wish come true. Using EZwidget, you can add a real store to your Web site, complete

with item photos and pricing within five minutes.

Would You Like Fries with Your E-Store?

EZwidget is a clever, wizard-driven interface for creating an online store. (A wizard interface guides you step by step through the process.) It's intuitive and easy to use. I'm confident that users at any level can successfully create a store. I was able to create a coffee shop with three items in less than five minutes. If you have images associated with items, they can be uploaded via your browser. In fact, the entire store can be maintained from a browser – a key feature many expensive systems lack.

Each step shows you what the store will look like when it's complete. For example, the selections for menu styles and colors are displayed as if in a browser, which makes it easier and more interactive to configure.

If you want to accept credit card orders in real time, you can opt for the live credit card authorization feature at around \$50 per month

VITALS

EZwidget: Webdriver Corp.

Web: <http://ezwidget.com>

Address: New York, New York

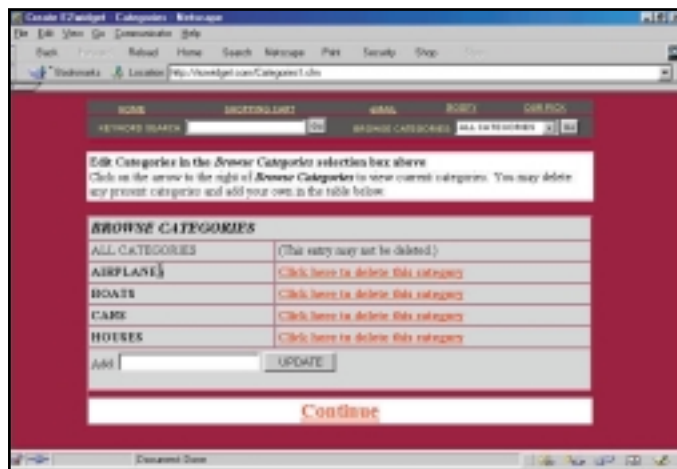
E-mail: info@webdriver.com

Test Environment: Desktop

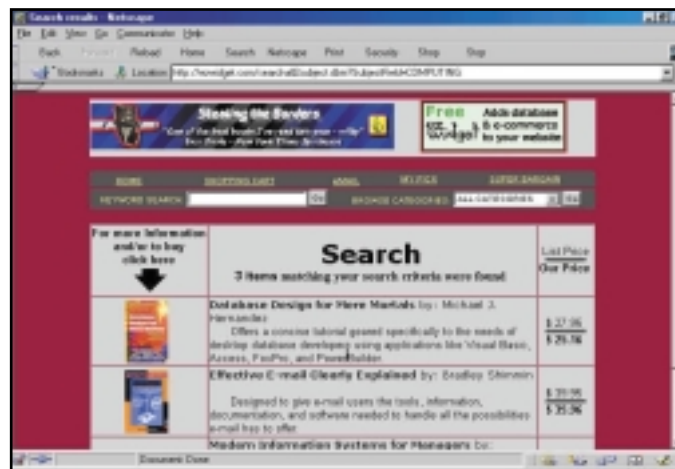
AMD k62, 128 MB RAM, 13 GB hard drive, dedicated Internet line

with a \$250 initial setup fee. If you select this option, your shoppers will automatically be directed to a secure page where they can enter their credit card information. After the order is verified, you're e-mailed a form with a list of the items ordered. Imagine that: a site that takes orders for you while you sleep! Even if you don't use the real-time billing option, you still get copies of all the orders. In addition, all sales are logged and stored on the EZwidget server so you can retrieve them at any time.

As in a "regular" store, merchan-



The category creator screen



The store demo screen

dise is displayed in aisles or sections, keeping the store neat and organized. EZwidget has a great category creator that allows you to easily define your own aisles within the store. When shoppers enter your e-store, they'll see a list of aisles and be able to make a selection. (In my e-coffee shop I created by-the-pound and coffee mug sections.)

Any Strings Attached?

As I always tell my children, nothing in life is free. The basic EZwidget store displays two banner ads on the top. EZwidget controls which ads are displayed and their frequency. While browsing several EZwidget e-stores I didn't find them too intrusive. With all the cross marketing on the Web, shoppers may be getting used to it. If you don't want the ads on your e-store, there's a \$25 monthly charge to remove them.

All EZwidget sites are hosted on a commercial host. I always recommend researching your host. Uptime,

bandwidth, concurrent user load and backup schemes are all relevant, especially when you're considering an e-store that you want open 24x7.

E-This, E-That

Since EZwidget is a generic store generator, the stores it creates look generic. It lacks all the customization options found in more expensive store creators. Most e-tailers are struggling to make their storefronts sexier and prettier to pull and keep the e-buyer in. As you may know from experience, the hot storefront won't necessarily keep you inside long enough to open your wallet. It's critical to make it easy to buy as well. Usually when I shop I take the in-and-out approach. Grab what I want, pay and get out!


The EZwidget store has a few extra steps. For example, once I find an item, I have to click on the photo and view all of the details before I can buy it. After I select an item, my filled shopping cart is displayed. If I want to continue shopping, I have to go back and

begin browsing again. Most e-stores flow this way. Drives me crazy. So many steps, so little time to shop!

I once evaluated an e-store with a shopping cart that always displayed on the left frame. I could continue shopping and glancing at my cart at the same time, just like I do at Pathmark. (Not Pathmark.com – I have to get out sometimes!)

Everything and the Kitchen Sink

Overall, EZwidget does fill a need: it creates a basic e-store quickly, easily and inexpensively, one that can be maintained by a novice user, and processes credit cards in real time. While every e-store product has benefits and drawbacks, few give you all the features of EZwidget for free.

If this is what you're looking for, I think EZwidget is well worth evaluating. 

ABOUT THE AUTHOR
David Schwartz is president of Array Software Inc., a New Jersey-based software company that creates global data-driven Internet and intranet Web sites using ColdFusion, Oracle, MS SQL Server and Java. David has been developing turnkey custom database software for 13 years.

DS@ARRAYONE.COM

VirtualScape

www.virtualscape.com

Tipping Points

Little things that make a big difference



BY
HAL
HELMs

In his new book, *The Tipping Point: How Little Things Can Make a Big Difference* (Little, Brown), Malcolm Gladwell makes the case for critical stages in the development of an endeavor.

Like tugboats steering a huge ocean liner, these “tipping points” wield enormous leverage, achieving results far beyond their apparent importance.

When, for example, the New York subway authorities decided to make subway riding safer and more comfortable, they placed their greatest efforts on two widespread but seemingly minor problems: graffiti and fare cheaters. The results were dramatic reductions in all subway infractions including violent crime and robbery. What caused this, Gladwell argues, is that these small crimes were tipping points that, when allowed to grow unchecked, poisoned the whole system.

I found the book interesting – so much so that I began making a list of the tipping points I see in software development. These are the little things I’ve found that create an inordinate impact on a project. See what you think.



1. **Explain your development process to the customer:**

At the beginning of a new project customers are excited about the boundless possibilities. Visions of what you will produce for them dance tantalizingly in their heads. In such an atmosphere it’s pretty easy to get caught up in the euphoria. I sometimes think our conference room, where most of these meetings take place, should have a sign posted that echoes the warning on ancient maps about uncharted waters: “Here there be dragons.”

It’s important to remember that customers rely on us for more than just technical prowess. They come to us for guidance on the development of a software project, and they’re almost never fully aware of its scope and complexity. It’s my job – and maybe yours – to offer a firm, accomplished and helping hand along the way. Explaining the whole development process helps the customer understand what each of us will need to do.

2. **Set clear project milestones spaced at short intervals.**

I don’t like the idea that we meet with a customer a few times, jot down some sketches and a few acronyms, then disappear into our coding lairs. Instead, I lay out our process graphically, explain the steps along the way and detail what each of us will be responsible for at each point. I tell them that as we continue along this path, we’ll reach certain points or milestones that tell us where we’ve been and where we’re going. Where the path winds back on itself, I explain the iterative nature of this section and

that, to go forward, we must agree the milestone has been met.

Each milestone has a deliverable and marks a passage from one phase of the project to another. Now we get a little crazy with this. At each stage both the client and the development team sign off, certifying the milestone has been met. The prototype we’re working on has visual indicators that we’ve moved from one phase to the next. At first glance it may all seem a little silly. I can only say that I prefer this harmless silliness to the outright lunacy of a development process that proceeds one step forward and two steps back.

3. **Never mind involving customers – pester them.**

I’ve done a lot of software projects of all types – amazingly, all for the same customer. At least it seems that way. Whether the customer is a large corporation developing an e-commerce site, a government agency or a small start-up, they all want the same thing: “Build the application and then we’ll tell you what we want.”

Of course, they don’t actually say this. Part of my job is to explain our development process, as described above. I tell clients that too often software developers involve customers at two points in the process. The first time is in the beginning, when it’s too early to get their feedback on the work in progress. The next time, sadly, is when it’s too late to do much about it.

My advice is to pester the customer. I like to say that customers and developers talk through prototypes. The customer tells us: “This is

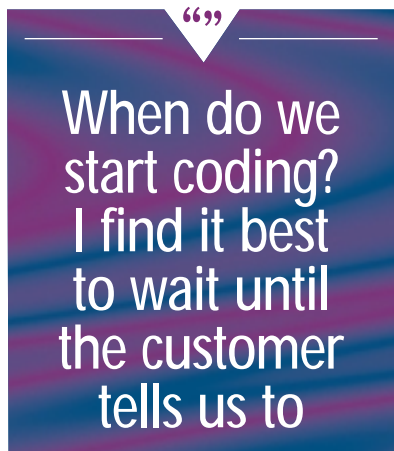
what I want,” and we go back and put together a mock-up.

“Like this?” we ask – and we keep asking until we get it just right.

4. **Resist the urge to code too soon.**

It follows that we shouldn't code until we know what it is we're supposed to be coding! Somehow, though, that's easier said than done. I've often found myself chomping at the bit to start coding when I knew more upfront work had to be done. But hey! I like to code! Over time I've trained myself to wait until our prototyping efforts bear ripe fruit. If you're working with less experienced developers, you may find that they need a little encouragement to trust the process and wait....

You may have noticed that the skills and temperament needed to be an excellent coder often don't match the ones required of those folks who work with the customer to unearth requirements. It seems to work best if we don't put coders in this position. Coders should code, letting others skilled in analy-



sis and user interface design add their expertise to the development mix.

When do we start coding? I find it best to wait until the customer tells us to. During my explanation of the development process I explain the notion of an “architectural freeze” – the point at which all prototyping and requirement gathering stops and coding actually begins. All projects must negotiate between the competing goals of speed-to-market and completeness of specifications. We let the customer deter-

mine the proper tradeoff by telling us when they want to go to architectural freeze. Our contracts specify delivery *n* weeks after architectural freeze.

5. **Document your code before you write it.**

This one seems a bit odd to many developers but you might want to give it a try. It's a wonderful discipline that helps you understand if you fully grasp both the broad lines and the individual details of a project. You'll probably uncover gaps in your knowledge, and the resulting documentation will help you standardize on naming conventions as well as reveal what functional objects are candidates for reusable components.

What documentation should you include? This depends on your methodology as well as your shop standards, but I generally suggest to people that they include:

- Responsibilities of the fuse – what it's supposed to do
- Author and subsequent editors of the code file

Fusion. We've got it down cold.

- ColdFusion Hosting
- In-house ColdFusion Developers
- Microsoft FrontPage, ASP, Media Player
- Co-location Services, SSL Secure Server
- RealAudio, RealVideo, RealFlash G2 Server
- Adhost Merchant, a scalable, customizable storefront
- Redundant DS-3 connections, bandwidth scalable to OC-3



www.adhost.com www.adhostmerchant.com sales@adhost.com 888-234-6781

- The variables directly passed to the code
- Variables directly passed from the code
- Global variables the coder needs to know of (e.g., session-, client- and application-scope variables)
- Global variables the coder is responsible for setting
- Subsidiary files (includes query files, functions, stylesheets, etc.) that the coder needs to be aware of

If you haven't already standardized a documentation specification, you might consider my Fusedoc spec, outlined in the April issue of **CFDJ** (Vol. 2, issue 4).

6. **Adopt a consistent coding style.**

Admittedly, this is much more important if you're working with a team of programmers than by yourself, but all of us can benefit from this. I find it's most important when it's 3:00 a.m., five hours before a delivery deadline, and I'm debugging a particularly troublesome bit

of code. Being able to scan a page of code that's consistently laid out is immensely helpful. When a new coder is brought onto a project, both this person and the existing team will benefit from a specified way of formatting code.

It's not a matter of one way or the other being right. A former boss of mine used to shake his head and tell me about coders that were so obsessive they would take the time to reformat the code before they would work on it. "Imagine that..." I would mutter, looking away guiltily. So, to avoid the "curly braces war," we need to sit down at the negotiating table and agree on a common standard of practice.

7. **Test everything you write.**

Now I know this is second nature for you, but I have to have a prescribed method for doing this – or it'll stay an unrealized good intention. I do so by insisting that all unit-level code I write has a "test harness" accompanying it.

The test harness sets up the appropriate environment (with the requisite variables defined and initialized) and runs my unit code.

Of course, using Fusedoc is a great help as I can immediately tell what the proper environment should consist of. Your test harness can test for such common errors as nulls and boundary conditions (extremes on both ends of a continuum of values). I sometimes write the test harnesses before I write the code itself – gaining many of the benefits of writing documentation prior to the actual code.

8. **Try "pair programming"**

Uh-oh. I fear I've just lost many of you. This is one of those counter-intuitive ideas that has to be tried out to be evaluated. Pair programming is two coders working on the same code at the same computer with a single keyboard/mouse.

I can tell you that my own experience with pair programming has taught me to value it greatly. A nat-

Sitehosting.net

www.sitehosting.net

ural rhythm develops in which one member of the pair buckles down to the immediate task while the other is thinking more broadly and strategically, exploring how this code may affect other code, possibly not yet written. These functions aren't permanent. When one member gets stuck – or just tired of typing – the places can be switched.

Most pointy-haired bosses aren't overjoyed by the idea that one expensive programmer working on a single piece of code has been replaced by two expensive coders working...well, you get the point. Again, I can only suggest that you try it.

9. **Separate form from function.**

This isn't an easy task nor are there any sure, easy answers. Still, we can gain from "refactoring" our code – looking for areas where functions and data can be abstracted into non-visual objects, complete with methods that can be accessed by other application pages. Without this, our code tends to get "smeared out" among many pages, making maintenance

difficult and reuse almost impossible.

You might also think about adopting a more formal methodology that supports the distinction between the model and the view, such as the one presented by Ralph Fiol in his fine work at www.cfobjects.com. Like a consistent coding standard, there's no right way, but applying the general principle of separation of form and function usually results in better code reuse and maintainability.

10. **Strive for simplicity.**

An old Shaker hymn tells us, "'tis a gift to be simple; 'tis a gift to be free." It may be a gift, but it certainly isn't easy to come by! Complexity and obscurity seem to achieve a life of their own, slipping uninvited into our code.

For example, I recently looked at some convoluted code that was troubling a developer who was trying to maintain it.

```
<cfloop from="1"to= "#ListLen(Trim
(ValueList(myQ.Name)))#" index="i">
```

```
#ListGetAt(ValueList(myQ.Name),i)#<br>
</cfloop>
```

I suggested that future coders called on to work on that particular code file would thank her if she rewrote it like this:

```
<cfloop query="myQ">
  #name#<br>
</cfloop>
```

To put it simply, as my old boss used to say, "Sometimes, Hal, clever code is bad code."

We developers can have an enormous influence on the world around us. The Internet – and the Internet economy – would never have developed had the pointy-haired managers been running the show. I hope that by examining my candidates for tipping points, you'll find your own way to leverage your own power – and maybe revolutionize the world again.



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

Ektron

www.ektron.com

Getting Focus()ed – and a Quick JavaScript Lesson



BY
CHARLES
AREHART

Tips and tricks relating to setting the form field focus using JavaScript

You visit a Web site that offers a form inviting you to enter some input. No big deal – we see them all the time, right?

But do you move your cursor to that first data entry area using the mouse? Or are you, like me, a keyboard maven – in which case you find that often you have to hit the tab key several times before you can enter data?

I've visited sites with navigational toolbars across the top and/or left side of the form – forcing me to tab dozens and dozens of times. It's an annoyance, especially if visitors to your site know the site designer could have easily prevented the problem.

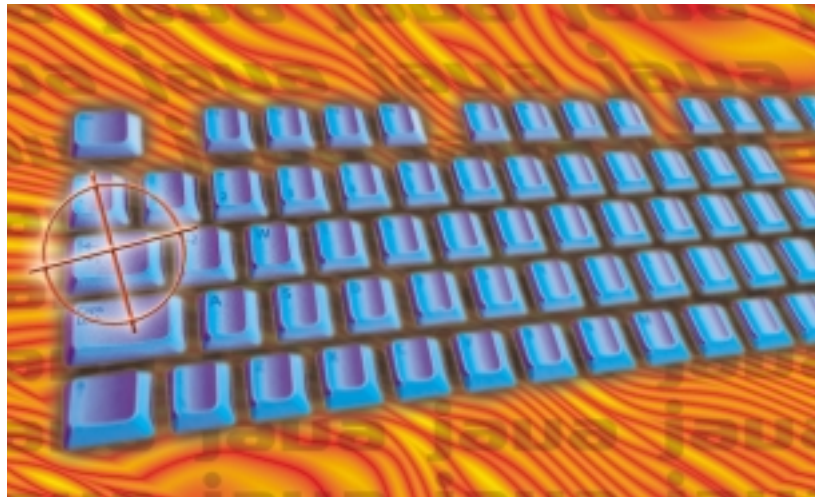
There's an incredibly simple solution. Just the tiniest bit of JavaScript. A single line of code, really. This article will show you how...and will also lay the most basic foundation of using JavaScript if you're new to it.

Laying the Foundation

The technique we're talking about here involves putting the "keyboard focus" on whatever form input field – a text, password or textarea box or even a radio, checkbox or select field – you want the user to enter data into first. The solution is the JavaScript "focus" method.

The JavaScript language is available to any Web page designer. (Microsoft must refer to their version of the language, made popular by the Netscape browser, as *JScript*. They're nearly the same, especially for the purposes of this article.) It's a scripting language that can be used to add features to your Web page that aren't otherwise provided by HTML.

This article will show you the simplest application of this focus method. It's pretty straightforward and should work just fine in most instances. JavaScript is widely supported by most browsers now and this



particular feature is also supported pretty much identically in both Netscape's and Microsoft's versions of the language, so there's no need to worry about it breaking on different browsers.

This built-in focus "method" is pretty much the same sort of thing as a function in ColdFusion – like now(). In fact, we'll even use it in a similar fashion, referring to it as "focus()" in our code.

Gimme the Goods

We need to put this method in some JavaScript code inside our page. If you haven't coded JavaScript on your pages, you need to know that there are several ways to cause JavaScript to be coded on a page. You can embed the code in a pair of `<SCRIPT>` tags (and even then you have choices about where those tags are placed) or you can place it in special attributes of HTML tags (such as `ONLOAD` within a `<BODY>` tag).

Let's take a simple example in just the first of those two forms. We can use the JavaScript `alert()` method to

cause a message to be displayed to the user in a popup box. To do that, we could use the following code:

```
<SCRIPT LANGUAGE="JavaScript"
TYPE="text/javascript">
  alert('Hello World');
</SCRIPT>
```

Put that code just about anywhere in an HTML page, and when you view that page in a browser you'll see a message pop up: "Hello World." Try it. We can use the same approach for specifying JavaScript code to indicate that we want to have the focus placed on a particular field. We're almost ready!

Whither the Form Field?

The `focus()` method operates on form fields. One of the many powers of JavaScript is that not only can we refer to all the elements of our page (all the forms, form fields, indeed every tag and its contents and attributes), we can manipulate them as well.

How do we specify that we intend

to work with some part of our page, such as a particular form field? Most of you will know that you can give a name to a form field, and in CF that will become the name of a form field on the form's action page. In our JavaScript we can use that same name to refer to the form field in order to give it focus. So if we have an input field for a "UserId" that should be entered, the HTML would be:

```
<FORM ...>
...
Enter UserId: <INPUT TYPE="text"
Name="UserId">
...
</FORM>
```

Of course, the <FORM> tag would be specified completely and there would be other form field information provided both before and after this input field. But it's important to know that this does indeed (and must) occur within a <FORM> tag.

One reason that's important (besides being the way HTML is coded!) is that when we want to refer to a particular element within a Web page from within JavaScript, we need to refer to it in the form of its relationship to the entire page. This input form field, "UserId", occurs inside of a form. So on a simple level, we would refer to it as form.UserId. This isn't technically correct, however, since it's possible to have more than one form in a page. We need instead to indicate the specific form in which the field occurs.

What's in a Name?

There are two ways to do this: we can refer to the form either by name or by indicating the relative location of the form on the page. Which will be appropriate for you depends on your situation. If you've specified a NAME attribute on the <FORM> tag, then you'd use that name.

If we had specified, for instance, NAME="Login" on the <FORM> tag, then we would refer to the form field as Login.UserId. But we're not done yet. The form itself occurs within the Web page, and while it mightn't seem necessary, the contents of the page are considered to be within the "document" on the page. (If you've heard of the Document Object Model, what we've been describing is part of that.)

We finally have the complete means by which to refer to the field: document.Login.UserId.

Before leaving this subject, let's clarify that if you haven't named your form, you can still refer to it by indicating the relative location of the form within the document. There is a special "forms" array in every Web page document, so if you have only one form in yours, you would indicate it as the first form.

Unfortunately, even this can trip you up because in JavaScript (as in many languages) you start counting lists of elements at 0 rather than 1. So the way to refer to your form field (assuming it's within the only form on the page) would be: document.forms[0].UserId.

Notice that this isn't used as "form[0]" but as "forms[0]". It's a subtle point, but again: if it's not specified correctly, you'll receive an error.

We're on the Case

It's critically important to remember that JavaScript is case-sensitive. The case you choose when you name something yourself must be the case you use when referring to it later with JavaScript.

If we were to refer to our form field above as Document.Forms[0].UserId, for example, we'd get an error. JavaScript expects us to refer to both the "document" and "forms" elements as all lowercase. You can't use uppercase, not even for the first character.

Similarly, since we named the form field UserId, we must refer to it exactly that way. If we referred to it as document.forms[0].userid, or even document.forms[0].Userid, we'd receive an error either way.

Forcing the Focus()

In order to set the focus, we use the focus method, which is specified with parentheses after it, like a function in CF – though JavaScript methods are specified by appending them to the end of the name of the object they'll work on. To set the focus for our form field called "UserId", we might specify it as:

```
document.forms[0].UserId.focus()
```

That's about it. There's nothing to be specified within the parentheses. (Notice that the word *focus* is also all lowercase.) This statement, when executed,

will cause the named field in the named form to receive the focus when the page is displayed to the user.

Our final challenge is to decide where and how to specify this statement. Of course, it's JavaScript and must be specified to be executed as such. We mentioned earlier that this could be done by way of the <SCRIPT> tag. The following code will cause the focus to be placed on our UserId field:

```
<SCRIPT LANGUAGE="JavaScript"
TYPE="text/javascript">
<!--
document.forms[0].UserId.focus()
//-->
</SCRIPT>
```

If only it were this easy. Well, it is – almost! You just need to make sure the statement is executed after the form has been loaded. In JavaScript, as in ColdFusion, it's inappropriate to refer to a variable (or form field, as it were) before it's been defined. In this case, with this approach to executing the statement, we just need to be sure not to put this code on our page prior to the form itself.

There are multiple ways to specify JavaScript within a page. Besides the <SCRIPT> tag, we can also use event-handler attributes of other tags. For instance, we could also execute the JavaScript only after the entire page has loaded. Some may know that there's an attribute for the BODY tag called ONLOAD; it does just what we need, causing whatever statements it executes to be run only after the page has loaded. And since the form is within the page, we have another solution to our problem.

Fortunately, we can very easily specify our JavaScript statement (the statement itself, not the <SCRIPT> tags) by placing it in the BODY tag's ONLOAD attribute, as in:

```
<BODY
ONLOAD="javascript:document.forms[0].
UserId.focus()">
```

Pretty nifty! (The "javascript:" directive preceding the statement is not usually required, but it's the formal method of specifying JavaScript within the ONLOAD attribute.)

Experienced JavaScript developers



FIGURE 1 Allaire Developer's Exchange Front Page - Secondary Form

may prefer yet another approach: creating a function instead and calling that function in the ONLOAD.

But Will It Fail?

You should have no trouble if you specify the JavaScript statement using one of the two appropriate approaches, refer to the form field using the naming references indicated, remember to specify the proper case for the naming reference and ensure that the statement occurs only after the form has been loaded.

There are a couple of other edge cases to be aware of, however. First, some folks use a single page to serve as both the form and action page (loaded from within the same CF template, for instance). That's fine, and can be a powerful way to reuse lots of code, but be careful: if you leave JavaScript code referring to the form field on the page, you'll get an error when it's acting as the form's action page – because the form itself won't usually have been sent on that page. This is particularly risky when using the ONLOAD approach, because you may not think of it when coding the page to serve both the form and action page.

In a similar vein, you don't want to include the JavaScript statement on the page if the form field to which it's referring isn't included in the page. With a tool like ColdFusion dynamically building the page, this isn't so strange as it might sound. Forewarned is forearmed.

What if a user's browser simply doesn't support JavaScript? With the <SCRIPT> tag approach described

above, you may notice that just inside the opening and closing <SCRIPT> tags there are HTML comment tags. This is a standard JavaScript approach that guarantees that if the browser doesn't support the <SCRIPT> tag (which it will simply ignore), it will also ignore the JavaScript statements inside the script and comment tags. But browsers that support JavaScript still do execute the statements inside the comment tags. As musician Bruce Hornsby sang, "That's just the way it is."

With the ONLOAD approach, browsers that ignore JavaScript will also ignore attributes they don't understand. The ONLOAD attribute is purely and simply for executing JavaScript and hence is ignored by non-JS browsers. By association, the value specified for that attribute (our JavaScript statement in the second method above) is also ignored.

Some Caveats

Why not add the focus method to the first input field to the forms in *all* your pages? Well, first it may not be appropriate to the interface. On some pages, the form and its input fields may not really be the focus of the page. Let's take a look for example at the front page of the Allaire Developer's Exchange, at www.allaire.com/developer/gallery.cfm (see Figure 1).

If you look closely, there is indeed a form on this page but it's definitely not the focus of the page – it appears in the lower-right corner of the screen. Were we to put the focus there by default, it would be fine for someone about to enter his or her e-mail address; the signup form, however, clearly isn't the focus of this page, so giving it the focus would be inappropriate.

There are other forms at the Allaire site, such as the Knowledge Base search form www.allaire.com/Support/KnowledgeBase/Search-Form.cfm, which have as their sole purpose the entry of search criteria for their respective pages. So putting the focus on the input field there would indeed be appropriate.

Another issue is that the e-mail signup form is also located quite far down the page, so even if we did choose to give it the focus, an unintended result would be that the screen would scroll down when displayed to the user so that the focus

(that form input field) was visible. This may cause the top of the page to be hidden from the user. Remember too that users often have different (and lower resolution) monitors than developers, increasing the chances of this problem developing. It can also happen if the users don't maximize their screen when displaying your page.

Afterword: Using SCRIPTJ in Studio

One last comment. This article has been as much an introduction to using JavaScript as it has been about the specifics of setting the form field focus. As such, there's one last really useful trick you should know about. It's a feature of ColdFusion Studio and its sister product HomeSite.

Remembering again the <SCRIPT> tags offered above, with the HTML comment characters specified within them, I mentioned that these are a standard set of tags to be used for all JavaScript entered within a page (other than that entered within other tag attributes like ONLOAD).

Well, you could hope to remember to enter that properly formatted set of basic tags...

```
<SCRIPT LANGUAGE="JavaScript"
TYPE="text/javascript">
<!--

//-->
</SCRIPT>
```

...or you could take advantage of a really nifty shortcut in Studio: simply type SCRIPTJ, press CTRL-J (the control key and J) and watch as Studio converts that into the set of tags offered above. The cursor is even sitting within the paired tags waiting for you to type in your JavaScript. (Hey, now there's a great use of cursor focus!)

Summary

Using the focus method is a win-win that should be used in nearly all Web pages. Plus it really is just a single statement of JavaScript code in nearly all cases. So now do you see why I find it so frustrating when a site doesn't use it?



ABOUT THE AUTHOR

Charles Arehart is an Allaire certified trainer and CTO of SysManage, an Allaire partner. He contributes to several CF resources and is a frequent speaker at user groups throughout the country.

Career Opportunities

ADVERTISERS INDEX

ADVERTISER	URL	PH	PG
ABLECOMMERCE	WWW.ABLECOMMERCE.COM	360.253.4142	2,4
ADHOST	WWW.ADHOST.COM	888 ADHOST-1	37
ALLAIRE	WWW.ALLAIRE.COM	888.939.2545	11,27,45
CAREER OPPORTUNITIES			53
CATOUZER	WWW.CATOUZER.COM	604.662.7551	59
COMPUTERJOBS.COM	WWW.COMPUTERJOBS.COM		3
COMPUTERWORK.COM	WWW.COMPUTERWORK.COM		25
CORDA TECHNOLOGIES	WWW.CORDA.COM	888.763.0517	31
DEVELOPERSNETWORK	WWW.DEVELOPERSNETWORK.COM	416.203.3690	15
DIGITALNATION	WWW.DEDICATEDSERVER.COM	703.642.2800	9
EKTRON	WWW.EKTRON.COM	603.594.0249	29
INFOBOARD	WWW.INFOBOARD.COM	800.514.2297	41
INTELIANT	WWW.INTELIANT.COM	800.815.5541	47
INTERMEDIA.NET	WWW.INTERMEDIA.NET	650.424.9935	60
JAVACON 2000	WWW.JAVACON2000.COM		42-43
JAVAONE	WWW.JAVA.SUN.COM/JAVAONE/		33
JDJ STORE	WWW.JDJSTORE.COM	888.303.JAVA	54
ON-LINE DATA SOLUTIONS	WWW.COOLFUSION.COM	516.737.4668	30
PAPERTHIN INC.	WWW.PAPERTHIN.COM	800.940-3087	39
RSW SOFTWARE	WWW.RSWSOFTWARE.COM	508.435-8000	13
SAISOFT	WWW.SAISOFTONLINE.COM	860.793.6681	30
SHIFT4 CORPORATION	WWW.SHIFT4.COM	800.265.5795	21
SITEHOSTING.NET	WWW.SITEHOSTING.NET	888.463.6168	41
SYS-CON MEDIA	WWW.SYS-CON.COM	800.513.7111	51
VIRTUALSCAPE	WWW.VIRTUALSCAPE.COM	212.460.8406	16
WATCHFIRE	WWW.WATCHFIRE.COM	613.599.3888	17
WINMILL SOFTWARE	WWW.WINMILL.COM	888.711.MILL	49
XML DEV CON 2000	WWW.XMLDEVCON2000.COM		22-23

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

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. 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

ComputerWork.com

ComputerWork.com is a premiere technical job site for computer professionals seeking employment in the IT/IS industry. ComputerWork.com will match your technical skills and career ambitions to our many employers looking to fill their jobs with specialists in computer related fields. You can submit your resume to a specific position on our job board or you can choose to submit your resume to our resume bank, which is accessed by nearly 400 companies nationwide. ComputerWork.com is the FASTEST way to your ideal career!

6620 Southpoint Drive South, Suite 600 Jacksonville, FL 32216
www.computerwork.com • 904-296-1993

Corda Technologies

Corda Technologies offers tools to build your own charts and graphs for internal reports, reports on your intranet and Internet sites and for many other applications where fast, high-quality graphs and charts are desirable. Corda also offers an Application Service Provider through PopChart.com which works with high-volume sports web sites to display sports statistics with exciting, interactive charts and graphs. PopChart!...an EXPLOSION of Possibilities!

1425 S. 550 East Orem, UT 84097
www.corda.com • 801-802-0800

DevelopersNetwork.com

Developers Network is the essential online business-to-business resource for new media technology and Internet business solutions. Our Resource, Business and Product channels combine elements of helpware and community in a business setting, successfully reaching those buyers developing and managing Internet strategies.

3007 Kingston Road Toronto, Ontario CANADA M1M 1P1
www.developersnetwork.com • 416-203-3610

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 non-technical 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

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

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

Highlight your website with
infoboard
NT and UNIX
Cold Fusion Hosting
Development Consulting
Oracle, Informix, MS SQL, E-Commerce Plug-ins
1-800-514-2297
<allaire> Alliance
Partner
sales@infoboard.com
www.infoboard.com

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

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

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. 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.

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

Shift4 Corporation

Shift4 Corporation is a leading provider of credit card and transaction processing software utilized by merchants and application developers in various industries. Shift4 products facilitate the point-of-sale and accounting functions associated with electronic payment media, including credit cards, debit cards, purchase cards, specialty cards, private label cards and electronic check clearing. More than 2,000 customers worldwide utilize Shift4 software to process over 85 million credit card transactions annually.

8691 W. Sahara Ave. Las Vegas, NV 89117-5830
www.shift4.com • 800 265-5795

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 life-cycle 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

To place an ad in the
ColdFusion Marketplace
contact Robyn Forma at 914 735-0300

Macromedia Launches Dreamweaver UltraDev

(San Francisco, CA) Macromedia, Inc., announces Macromedia Dreamweaver UltraDev, the first authoring software that allows developers, programmers and designers to visually create and edit data-driven Web applications on multiple server platforms.

In addition to specific functionality for building applications, Dreamweaver UltraDev provides all the features of Macromedia Dreamweaver 3.

www.macromedia.com 

WebPerfect Solutions Adds Palm.net Access to Its Wireless Intranet Service

(Vienna, VA) - WebPerfect Solutions, Inc., has added Palm.Net access to its wireless intranet service, providing

Palm VII users with fully integrated access to its intranet service, including e-mail, file sharing, forms management, project management, contact management and scheduling.

www.webperfect.com 



AbleCommerce Adds New CEO to Management Team

(Vancouver, WA) - David R. Henderson has joined AbleCommerce as its new CEO effective immediately. With over 30 years of management experience in a variety of technical and administrative roles, Henderson adds considerable depth to the AbleCommerce executive team. Michael J. Randolph, founder and former CEO of AbleCommerce, will continue as chairman, focusing on research and

development.

Henderson last served as chairman and CEO of Emerald Computers, Inc. He attended the University of Washington and is both a mechanical and an electronic engineer.

www.ablesolutions.com 



ColdFusion Community Sites Share Resources

Fusion Authority and CFNewbie, two sites dedi-

cated to providing information to the ColdFusion community, have joined forces, allowing for a free exchange of information between the sites. Starting immediately, Fusion Authority articles that are written for new ColdFusion developers will be reprinted in CFNewbie and CFNewbie articles will be reprinted in Fusion Authority.

www.fusionauthority.com

www.cfnewbie.com 

Array Software Launches HotQuery 1.0

(Edison, NJ) - Array Software Inc. announces the release of HotQuery, the first 100% browser-based database query engine and reporting tool. HotQuery allows you to view, query and report on Oracle and SQL Server tables and views stored on a remote Web server or LAN directly from any standard Web browser.

"HotQuery is invaluable

in urgent, time-sensitive RAD project development and administration. Most reporting functions for users and administrators are easily handled without time-consuming report generation," said Dale Deffenbaugh, project developer, American Honda Motor Co, Inc. "

A full-featured Web-based demo is available at www.hotquery.com. 

Career Opportunities

Saisoft

www.saisoftonline.com

Catouzer

www.catouzer.com

intermedia.net

www.intermedia.com