

COLD FUSION Developer's Journal

ColdFusionJournal.com

April 2002 Volume: 4 Issue: 4

web services **EDGE**
world tour 2002

NEW YORK, NY.....**APRIL 19**

SAN FRANCISCO, CA.....**MAY 13**

Page 18 Register for Web Services Edge
2002 East Gold Passport,
Attend the World Tour FREE!

Editorial **Going North!**

Robert Diamond page 5

Q&A

Ask the Training Staff
Bruce Van Horn page 42

CFDJ News
page 50

Full Conference Program
INSIDE page 19

THE LARGEST WEB SERVICES CONFERENCE & EXPO IN THE WORLD

WORLD SERVICES **EDGE** **JD/EDGE** **XML/EDGE**

Conference: June 24-27 Expo: June 25-27

SAVE \$200

Call Today!

The Future of Enterprise Computing

JACOB JAVITS

Register Today!

SYS-CON
MEDIA

Toward Better CF Server Administration

by
Charles Arehart
page 34

CFDJ Feature: Robust Session Management Part 1

Surviving the slings and arrows of outrageous users

```
<cfif defined("cookie.CFID")>
```

Philip Chalmers

Foundations: In Praise of Simple Tools

What's the point of change for the sake of change?



Hal Helms

CF Techniques: How to Sidestep Locking

Should you lock? You don't really have to...

Sandy Clark & Michael Smith

CFDJ Feature: Back to Basics: When sites run into trouble, it's usually the database that's the problem, not CF

Tom Peer

Debugging Tips: A More Thorough Debugging

Some often overlooked techniques to improve your apps

Eric Brancaccio

CF Tips: Session States Without the Session

Why on earth would you want to try such a thing?

```
<!-- Build the "session" structure -->  
<!-- Set a default number of variables or pass it in as "numVars" -->  
<!-->
```

Tsara Borsting

CF Community: Tales from the List

CFDJ discussion threads...and their creators



Simon Horwith

EMPIRIX

www.empirix.com/double/cfm

INTERLAND

www.interland.com

ACTIVEPDF
www.activepdf.com

international advisory board

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

department editors

editor-in-chief
 Robert Diamond robert@sys-con.com
vice president, production
 Jim Morgan jim@sys-con.com
executive editor
 M'lou Pinkham mpinkham@sys-con.com
managing editor
 Cheryl Van Sise cheryl@sys-con.com
editor
 Nancy Valentine nancy@sys-con.com
associate editor
 Jamie Matusow jamie@sys-con.com
associate editor
 Gail Schultz gail@sys-con.com
associate editor
 Jean Cassidy jean@sys-con.com
product review editor
 Tom Taulli
tips & techniques editor
 Matt Newberry

writers in this issue

Charles Arehart, Tsara Borsting, Eric Brancaccio,
 Philip Chalmers, Sandy Clark, Robert Diamond,
 Hal Helms, Simon Horwith, Tom Peer,
 Michael Smith, Bruce Van Horn,

subscriptions:

For subscription requests please call
 1 800 513-7111 or go to: www.sys-con.com
 cover price \$8.99/issue
 domestic \$89.99/yr. (12 issues)
 canada/mexico \$99.99/yr
 overseas \$129.99/yr
 back issues \$12 U.S. \$15 all other

editorial offices: SYS-CON MEDIA, INC.

135 Chestnut Ridge Rd., Montvale, NJ 07645
 Telephone: 201 802-3000 Fax: 201 782-9600
 COLD FUSION DEVELOPER'S JOURNAL (ISSN #1523-9101)
 is published monthly (12 times a year)
 by SYS-CON Publications, Inc.

postmaster: send address changes to:

COLD FUSION DEVELOPER'S JOURNAL
 SYS-CON MEDIA
 135 Chestnut Ridge Rd., Montvale, NJ 07645

copyright © 2002 by SYS-CON MEDIA

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:
 Carrie Gebert carrieg@sys-con.com

SYS-CON PUBLICATIONS, INC., reserves the right to revise, republish, and authorize its readers to use the articles submitted for publication.

distribution in USA:

by International Periodical Distributors
 674 Via De La Valle, Suite 204
 Solana Beach, CA 92075
 Phone: 619 481-5928

All brand and product names used on these pages are trade names, service marks, or trademarks of their respective companies.



Going North!

BY ROBERT DIAMOND



On May 3 I'll be packing my bags and trekking north from New York to Toronto to participate in CF North, the only Canadian developer's conference for ColdFusion. A large chunk of the ColdFusion world will be heading up that way with me, both those in the area and those elsewhere looking to network between now and the Macromedia DevCon this fall. It's a worthwhile event – enough for me to leave my new dog at home. (Say Hi to

Max, the world's cutest pug and the first dog determined to learn CFML.)

Taking place at the Toronto Eaton Centre Marriott Hotel on May 4–5, the event will consist of 40 high-quality CF sessions on a wide variety of topics. The conference has been smartly divided into the following five tracks to make scheduling easier throughout the intense two-day period: ColdFusion Bootcamp, Best Practices, Geek Out: Advanced Technologies, Flash and ColdFusion, and Fusebox Methodology. Classes will be taught by many experts in the CF world, including many whose articles you see here in *CFDJ* every month. For more information on the sessions offered, and for registration details, visit www.cfnorth.com.

A series of keynote addresses by such industry luminaries as Jeremy Allaire and Ben Forta will be among the highlights of the conference. A special keynote is planned by Bruce Scott, president and CEO of Pointbase, Inc., mobile database leaders in the Java industry. If you aren't familiar with Bruce, who cofounded Oracle, maybe you're familiar with *scott/tiger*, the original user login/password combo for Oracle. He is also the coarchitect/author of the first three versions of Oracle.

One other keynote, a panel discussion led by yours truly and members of the *CFDJ* Editorial Board, will also be on the schedule (small, shameless self-promotion, but worthwhile). We'll be covering and discussing the hottest issues of the day, as well as taking questions from the audience in an interactive format.

Even if you can't make it to the conference, you can still be a part of it. We'll be accepting advance questions via e-mail, so if you have questions about ColdFusion, about the magazine, about the industry as a whole, we're assembling a good group of folks who are right in the thick of it. Drop me a line at the e-mail address at the bottom right of this page, and we'll try to fit you in. We'll be covering the panel keynote and the rest of the conference in future issues of *ColdFusion Developer's Journal*.

• • •

Back to the magazine. This month we're introducing something brand-new. Simon Horwith, a senior developer and Macromedia-certified ColdFusion instructor at Fig Leaf Software, contributes *CF Community*: "Tales from the List." The column, which will appear monthly, is devoted to the latest happenings on the *CFDJ*List. Simon is the new manager for the list, one of the most active ColdFusion discussion communities on the Net. For more information on that, jump on over to www.sys-con.com/coldfusion/list.cfm.

Also in this issue: Hal Helms writes "In Praise of Simple Tools"; Bruce Van Horn explains some techniques for querying e-mail addresses and porting applications; Galen Winchip writes on "Real-Time Database Validation of Form Inputs"; and Eric Brancaccio covers some often overlooked techniques in "A More Thorough Debugging."

On the heels of last year's successful *CFDJ* Readers' Choice Awards, the upcoming month will have news regarding this year's nominations.

Hope to see some of you at CF North in Toronto, and the rest here and online.



ABOUT THE AUTHOR

Robert Diamond is editor-in-chief of *ColdFusion Developer's Journal* as well as *Wireless Business & Technology*. In addition, he is the director of technology at SYS-CON Media. Named one of the "Top thirty magazine industry executives under the age of 30" in *Folio* magazine's November 2000 issue, Robert holds a BS degree in information management and technology from the School of Information Studies at Syracuse University.

Robert Diamond

@ ROBERT@SYS-CON.COM

Robust CF Session Management



ColdFusion Feature

By PHILIP CHALMERS

**Surviving the slings
and arrows of
outrageous users**

Part 1 of 2

Browsers and what users can do with them create a minefield for Web app developers. Users can:

- Disable all cookies. This creates unpleasant problems for CF's session management.
- Do something else for half an hour and then try to resume where they left off. Now the session data has probably timed out, but your app gets a page request that assumes the session data is still available.
- Trample all over the dialog logic by using Back, Forward, and Refresh, or by cloning browser windows. This may create a major security hazard.
- Enter the site at an inner page via a search engine result, a bookmark, or a URL e-mailed to them.
- Hit the Submit button several times, especially if the Web or your server is running slow. They may think it's a harmless way of letting off steam.
- Use proxy servers that serve as much as possible from their own caches so different users get the same versions of pages – sometimes including the same CF session-management cookies – so the users are sharing a session!
- Open your app in different browsers (e.g., IE and Netscape) so that CF creates separate sessions.

What can you do to make your app less vulnerable to the slings and arrows of outrageous users? And how should you keep the user informed of what's going on? (If you don't, he or she is likely to make mistakes and eventually avoid your site if the consequences of the mistakes are serious.)

My suggestions don't depend on the use of JavaScript; about 12% of users disable JavaScript or use non-JavaScript browsers (see www.upsdell.com/BrowserNews/res_design.htm).

Fusebox designers can easily adapt these ideas, for example, by substituting "fuseaction" for many occurrences of "page".

This article ignores the additional issues raised by clustered servers that make it virtually impossible to use session or updatable application variables. If your site uses or may use clustered servers, Marc Funaro's June 2000 article, "So You Want to Manage a Session on Load-Balanced Servers?" (*CFDJ*, Vol. 2, issue 6), explores the design and conversion issues thoroughly.

I'm assuming here that session-level data will be stored in session variables. I don't assume that client variables will be stored in any particular way.

I also don't cover how to combat skilled and well-equipped hackers who, for example, can detect and read cookies passed over a communications channel. If you're that concerned about security, you probably need to run your entire site through SSL and then add measures to combat the weaknesses in SSL.

This article is in two parts. Part 1 summarizes my recommendations and discusses why you need to use cookies for session management, as well as the detection and handling of timeouts. Highlights of Part 2, which will appear in a forthcoming issue of *CFDJ*, are listed at the end of this article.

Summary in Advance

In case you know most of this already, or want to use this article as a reference, here's a summary:

- Use cookies to identify your session. Passing CFID and CFTOKEN as URL parameters creates a security hazard for your app and for the user.
- Make all cookies per-session, including CFID and CFTOKEN. This will make it easier to persuade users to accept your cookies, and further reduces privacy and security risks to both sides.
- If you use CFLOCATION, always code ADDTOKEN="No" to avoid showing CFID and CFTOKEN as URL parameters.
- Each app should use its own session management cookie containing a list of name-value pairs delimited by nondisplay characters.
- Each page except the point of entry to your app must check whether cookies are enabled, and each page, including the point of entry, must set a "cookies enabled" name-value pair in the session management cookie for use in this check.
- If cookies are disabled, explain to the user why per-session cookies are the most secure way to manage a conversation.
- The "please enable cookies" page should always link to the start page rather than the page the user requested.
- Check for session timeouts. If a timeout occurs, tell the user what has been lost and what has not – for example, "Order number 12456 has been recorded in the database".
- To do this, your session management cookie needs a name-value pair for every logical group of data items. Set the name-value pair when the logical group is populated and delete it if the logical group is cleared.
- You should generally save a "last transaction was ..." name-value pair in the cookie so that your "session has timed out" message can help the user to avoid reentering transactions that were completed.
- When you display a "session has timed out" message, delete the relevant items from your session management cookie to prevent the dialog from looping.
- If you store updated application-level data, application timeouts are an issue. In this case you probably need to save the updated application data periodically in a database and run a scheduled task to purge out-of-date entries from the database.
- CFLOCK reads and writes of session data and of updatable application data.
- To avoid trouble with Back, Forward, multiple Submits, and cloned browser windows, maintain a session-level page counter and include the counter value as a HIDDEN INPUT in every form. Then check that the returned value matches the expected value every time your app is asked to update the database or important session data.
- Avoid requesting updates via links, as you'd then have to include the page counter value as a URL parameter and the user could edit it.
- To avoid trouble with proxy servers, add a random number as a URL parameter in all links and FORM ACTIONS.
- If you need to prevent or control the use of multiple sessions with the same user, require users to log in and maintain an application-level structure keyed on userid and containing the date and time of the last page request. Schedule a process to delete entries older than the session timeout interval.
- You can't force the user to log out – he or she can close the browser window.

Cookies

Web Apps Need Cookies

CF relies on getting CFID and CFTOKEN from the browser to identify each session. Without this information it treats each page as a new session. This is not a CF limitation, it's a consequence of the statelessness of the Web.

There are two possible ways of passing CFID and CFTOKEN between the browser and CF: (1) use cookies, or (2) pass CFID and CFTOKEN as URL parameters in every page request. The second method creates a security hazard because malicious or simply curious users may edit the URL parameters. Such hacking has cost some companies real money and might also cost users real money.

So you need the user to accept cookies. But some users don't like cookies for reasons of security and privacy. Other users (including me) accept per-session cookies but reject persistent cookies. Your chances of persuading the user to accept cookies will improve a lot if you promise to use only per-session cookies, that is, cookies that aren't saved to disk and are scrapped when the browser is closed.

Using only per-session cookies also improves the security of your app because they aren't stored when the browser is closed and hence the user can't edit them in order to hack a later session.

CF automatically creates cookies for CFID and CFTOKEN unless your CFAPPLICATION tag says SETCLIENTCOOKIES="NO". By default CF makes the CFID and CFTOKEN cookies persistent, that is, they're saved to disk. This code will make them per-session cookies:

```
<cfif IsDefined("Cookie.CFID") AND
IsDefined("Cookie.CFTOKEN")>
    <cfset cfid_local = Cookie.CFID>
    <cfset cftoken_local =
Cookie.CFTOKEN>
    <cfcookie name="CFID"
value="#cfid_local#">
    <cfcookie name="CFTOKEN"
value="#cftoken_local#">
</cfif>
```

onRequestEnd.cfm looks like an obvious place to do this, but onRequestEnd.cfm isn't processed if your app issues a CFABORT or CFEXIT, for example, in an error handler, or if there's no Application.cfm in the same directory as onRequestEnd.cfm.

So I prefer to make the CFID and CFTOKEN cookies per-session as early as possible, preferably in a CF template called by all output pages – for example, in a template that creates the app's logo or navigation bar.

Check That Per-Session Cookies Are Enabled

You can't read a cookie back in the page request that creates it because the cookie hasn't been sent to the browser yet. It will always appear as though the browser has accepted the cookie because CFCOOKIE automatically creates the corresponding cookie variable.

In general, all pages (action as well as display) must check that the expected cookies exist, except for your startup page, which can't check for cookies the first time it's opened. If your startup page is reentered later – if it's a search page, for example – you could make it check for cookies. But this is tricky because you'd need a session variable to count entries to the start page, and the session variable could time out.

All display pages must create a test cookie in order to test for cookies. The cookie must have the same name throughout the app. Don't use Cookie.CFID and Cookie.CFTOKEN for this check, because CF will automatically create these cookie variables before they're sent to the browser unless your CFAPPLICATION tag says SETCLIENTCOOKIES="NO".

All other pages (action as well as display) must check for the test cookie's "cookies enabled" value *before* creating the test cookie. If the check returns "undefined", they should display a "please enable cookies" page that links to your startup page. But these pages have no way of knowing whether the check returned "undefined" because cookies are disabled or because the user has entered the site by the back door (e.g., via a bookmark on an inner page) and hence the test cookie hasn't yet been sent to the browser. So the wording of your "please enable cookies" page must also allow for the possibility of entry by the back door.

The "please enable cookies" page should always link to the start page rather than the page the user requested. It would be user-friendly but dangerous to make the "please enable per-session cookies" page link to the page the user requested (including any URL parameters because:

1. The requested page may create session-level data that won't be available to later pages if the user didn't actually enable per-session cookies before clicking the link, because CFID and CFTOKEN are needed in order for CF to find session data. In addition, the requested page may need previously created session-level data that won't be available because the "please enable per-session cookies" page generally won't send the correct values of CFID and CFTOKEN even after the user enables cookies. CF keeps creating new values of CFID and CFTOKEN

since it's not getting them from cookies sent by the browser.

2. In the worst case, if the user has been playing with the browser's cookie settings, CF may receive old values of CFID and CFTOKEN and your app will either find that the corresponding old set of session variables has timed out or (much more dangerous) will use the old session values.
3. The user may have tried to enter the site by the back door, bypassing your security and dialog management.

Beware of CFLOCATION

CFLOCATION has two dangerous features: (1) it discards all previously built output, *including* cookies (in other words, it starts a new session!); and (2) its default is to set AddToken="Yes", so CFID and CFTOKEN are appended to the URL!

I wish I could say, "Don't use CFLOCATION," but the safest way to display a custom error page is to CFLOCATION to a static HTML page.

So always code ADDTOKEN="NO".

Checking for Timeouts

Timeouts on application and session variables are:

- Necessary to prevent servers' memory from being swamped.
- A degree of security for the user. They reduce the risk that someone else will use your application while the user's away from his or her desk.
- A dangerous nuisance when they happen, both for the user and for the developer. In the worst case you could perform a database update outside a valid context.

You Can't Avoid Session-Level Timeouts by Using Cookies

It would be nice to avoid session-level timeouts by storing session-level data in cookie variables. But browsers place tight limits on the number and size of cookie files (see Netscape's specification for cookies at http://home.netscape.com/newsref/std/cookie_spec.html):

- A maximum of 300 cookies in total; if a browser hits this limit, it discards the least recently used cookie
- No more than 20 cookies per site
- Maximum size is 4KB per cookie (including headers, delimiters, etc.)

CF makes the size limitation a bit more stringent. It encodes cookie data with a URL encoding scheme that expands alphanumeric data by about 30% and nonalphanumeric data by a factor of 3. It also throws an error if you try to store more than 4,000

Quill Design

www.quilldesign.com

president and ceo

Fuat Kircaali fuat@sys-con.com

vp, business development

Grisha Davida grisha@sys-con.com

advertising

senior vp, sales & marketing

Carmen Gonzalez carmen@sys-con.com

vp, sales & marketing

Miles Silverman miles@sys-con.com

advertising director

Robyn Forma robyn@sys-con.com

advertising account manager

Megan Ring megan@sys-con.com

associate sales managers

Carrie Gebert carrie@sys-con.com

Alisa Catalano alisa@sys-con.com

Kristen Kuhnle kristin@sys-con.com

Leah Hiltman leah@sys-con.com

editorial

executive editor

M'lou Pinkham mpinkham@sys-con.com

editor

Nancy Valentine nancy@sys-con.com

managing editor

Cheryl Van Sise cheryl@sys-con.com

associate editors

Jamie Matusow jamie@sys-con.com

Gail Schultz gail@sys-con.com

Jean Cassidy jean@sys-con.com

production

vp, production & design

Jim Morgan jim@sys-con.com

art director

Alex Botero alex@sys-con.com

assistant art directors

Cathryn Burak cathyb@sys-con.com

Louis F. Cuffari louis@sys-con.com

Richard Silverberg richards@sys-con.com

graphic designer

Aarathi Venkataraman aarathi@sys-con.com

Tami Beatty tami@sys-con.com

sys-con events

vice president, events

Cathy Walters cathyw@sys-con.com

conference manager

Michael Lynch mike@sys-con.com

sales executives, exhibits

Richard Anderson richard@sys-con.com

Michael Pesick michael@sys-con.com

customer relations/JDJ store

manager, customer relations /JDJ store

Anthony D. Spitzer tony@sys-con.com

web services

web designers

Stephen Kilmurray stephen@sys-con.com

Christopher Croce chris@sys-con.com

content editor

Lin Goetz lin@sys-con.com

accounting

chief financial officer

Bruce Kanner bruce@sys-con.com

assistant controller

Judith Calnan judith@sys-con.com

accounts payable

Joan LaRose joan@sys-con.com

accounts receivable

Jan Braidech jan@sys-con.com

accounting clerk

Betty White betty@sys-con.com

encoded bytes of data in a cookie. And it automatically uses separate cookies for CFID, CFTOKEN, and (if your CFAPPLICATION has CLIENTSTORAGE="COOKIE") CFGLOBALS (this contains client variables URLToken, HitCount, TimeCreated, and LastVisit). That's two or three of your twenty used already.

So most apps won't be able to store all their session-level data in cookie variables. Some session-level data shouldn't be stored in cookies in any case, as you'll see later.

Timeout-Handling Requirements Vary

We have to deal with a variety of situations. In some cases you need to check for timeouts only when the user requests a page that needs some session data. For example, in a shopping cart the customer's login is often required only for the checkout process and for updating the customer's registration details.

In other cases you may need to check constantly to see whether session data has timed out. For example, in a shopping cart you should warn the user immediately if the cart's contents have timed out, even if the page the user requested doesn't need the cart. Otherwise, when the user asks the app to display the cart or go to the checkout, he or she will find items are missing from the cart and will lose confidence in your app. Even worse, the user may not notice and so place an incomplete order, discovering the omissions only when it's delivered.

In general you need a separate check on the status of each logical group of session variables. *Logical group* is a rather fuzzy term, but I suggest logical groupings are largely based on the intended life cycle of the information. For example, login data will ideally persist until the user closes the browser (or logs out or in as someone else, depending on the system), but shopping cart data should be cleared immediately after an order is placed.

Logical groupings must also take account of whether you need to check the data on every page request or only when the user requests a page that needs the data.

Using Cookies to Detect and Handle Session Timeouts

A timeout is often detected when you get a request for a page that requires the data and it no longer exists. But if you're checking some session variables on every page request, you need to distinguish between a variable or a data structure that has timed out and one that just hasn't been populated yet (or has been cleared, e.g., a shopping cart when an order has been processed).

For each logical group of session data you need a "session data created" indicator that isn't invalidated by a session timeout. You should set this indicator when information is added to the logical group, rather than when an empty logical group is created (by

CFPARAM, for example). And you should clear or delete the indicator if the logical group is emptied – for example, when a shopping cart is cleared at the end of the checkout process.

Here are the options I'm aware of for storing these indicators:

1. In a per-session cookie that's unique to your app, a cookie with a maximum size of 4,000 bytes (including headers and delimiters), should be enough to store a few hundred "session data created" indicators.
2. You could maintain an application-level structure keyed on Session.SessionID and containing a "session data created" indicator for each logical group (probably as an array). This approach has the major weakness that application data can also time out. If application timeout intervals are long, you also need to write and schedule housekeeping processes to remove entries for expired sessions.
3. You can use a database in the same way. This costs additional disk accesses and processing time, but avoids the application timeout problem and conserves memory in heavily loaded servers, especially if application timeout intervals are long.

You'd still need to write and schedule housekeeping processes, though, and the database must be one that is good at reusing space freed by deleted records (MS Access is not).

For most apps per-session cookies are the simplest and most robust approach. To minimize the risk of too many cookies or of a session management cookie being overwritten, you should use one session management cookie per app, with a name that's unique to the app. It will then be read and rewritten by every page request, so the risk of its being discarded is very small.

You should store the app's session management information in this cookie as name-value pairs delimited by a character you won't use in a name or a value. I prefer delimiters that have no display values or effects – for example, from CHR(1) to CHR(6) CF automatically URL-encodes them when sending and URL-decodes them on receipt. This cookie should also contain your "cookies enabled" name-value pair.

Your session management cookie will be a list of lists, and CF's list-handling functions make it easy to manage this type of structure. In particular, you can easily unpack it into a session-scoped structure in order to add, change, and remove name-value pairs, and then reassemble the cookie before sending the next page.

I recommend a session-scoped structure because it has one set of values per session and you can CFLOCK updates to it, increasing your app's protection against simultaneous updates from cloned windows.



When you display a “session has timed out” message, delete the relevant item from your session management cookie to prevent the dialog from looping. This is particularly important if you check some logical data groups on every page request. For example, if you check the shopping cart’s status on every page request and don’t clear the “cart expected” indicator for the cookie, each page request will show the session timeout and the user will never get a chance to put anything in the cart.

If you have an enormous app with too many “session data created” indicators to fit in a cookie, you may need to use a database table as described above.

Which Pages Check for Which Session Variables?

Obviously, any page that needs a session variable to have a real value (not just a default) should check it. Otherwise it depends on the app. I’ve already suggested that every page in a shopping cart app should check that the cart’s contents have not been lost by a timeout.

What Should the Timeout Warning Page Say?

If the user can enter several transactions in a session, the simple message:

Sorry, your session has timed out. All the information you entered has been lost

could mislead the user into reentering transactions that have already been successfully completed. It would also be misleading if some session variables have timed out but others are okay. For example, if the user:

- Logs in to place an order
- Doesn’t use the app for longer than the timeout interval
- Places more items in the shopping cart for a second order
- Goes to the checkout


the login data will have timed out, but the shopping cart’s contents will be okay.

In this type of situation you need to tell the user explicitly what has been lost and what is okay, and reassure him or her that the last completed transaction (if any) has been recorded on the database, preferably with a link to a confirmation page.

To achieve this your session management cookie should keep a note of completed transactions by including, for example, a name-value pair “LastOrder : 123456”. You can then use this to display the last order number in the timeout warning page and as a URL parameter in the link to the confirmation page.

Checking for Application-Level Timeouts

There’s no problem if your application-level data is constant. It will always be

refreshed from Application.cfm or equivalent. But if it’s updated to accumulate statistics, for example, it’s not enough just to detect a timeout, you need to forestall it. One way to do this is to run a scheduled task at intervals of 5 minutes less than the application timeout to record the accumulated data in a database and reset the application-level counters. The scheduled task must also reset the application timeout clock so it would have to use the same application name.. 

In Part 2 of this article we’ll look at:

- Back, Forward, Refresh, cloned windows, and multiple Submits
- Proxy servers
- Users entering your site via an inner page
- Multiple sessions with different browsers
- Closing a session
- References and acknowledgments

About the Author

Philip Chalmers has been working in information technology since the early 1970s. He’s relatively new to ColdFusion, but has specified, designed, and developed systems on platforms ranging from mainframes to PCs in about a dozen languages. His first technical publication appeared in 1979, when he wrote several sections of the Adabas Design Guide.

 PHILIPCHALMERS@BLUEYONDER.CO.UK

CFDYNAMICS

www.cfdynamics.com

In Praise of Simple Tools

BY
HAL
HELMS



My first impressions of going to a professional baseball game are still vivid.

What's the point of change for the sake of change?

I was 11 years old and I went to see the greatest team in the world – the New York Yankees – play in Yankee Stadium.

I grew up listening to my brothers tell tales of watching Mickey Mantle, the head of the Yankee pantheon, and of the unfortunate Roger Maris, who, though great, never won the hearts of Yankee fans. Unfortunately, my life as a fan began during a hiatus between periods of Yankee dominance. Still, these *were* the New York Yankees, this *was* Yankee Stadium, and I was a huge fan.

Recollections are odd things. You remember bits and snatches: the smell of new-mown grass, the patterns in the playing field made by careful manicuring, the cry of ven-

dors barking, “Programs...get your programs here...can't tell the players without your programs!” But I knew all the players' names and numbers, knew their stats, and knew where they'd grown up. (This was before players were transferred annually, so an investment in knowing your team made sense: you'd see these same players year after year.) I had no need for a program.

Today, though, I'd have to pore over the program to make any sense of the players. The baseball world has changed faster than my ability to keep up. In apparent obedience to some lunar cycle, players change teams with dizzying regularity. Who wouldn't need a program?

Of course, such a rate of change appears absolutely stodgy in the world I now work in, where technology changes so rapidly that virtually any book is out of date by the time it's published. The latest technology to appear is Web services.

Web services, we're told, will change everything. Unfortunately, we were told this about last week's

favored technology. In fact, the phrase “This will change everything” has become a cliché, even a joke, wryly used to mean the exact opposite. What about Web services? Are they doomed to be just the latest in a succession of great ideas that flashed onto the scene and then were not so much rejected as forgotten in the hoopla of the next great thing's publicity?

Unfortunately, it's impossible to predict with any accuracy the fate of Web services. Certainly their goals are noble enough. Wouldn't we all benefit from sharing of information and collaboration of work? But

these goals aren't new. They've had predecessors – CORBA being perhaps the best known – and they have failed.

The reasons for failure or success are never as simple as the reasons usually given. Years ago the writer of the Biblical book of *Ecclesiastes* reflected on this: “I returned and saw under the sun, that the race is not to the swift, nor the battle to the strong, neither yet bread to the wise, nor yet riches to men of understanding, nor yet favour to men of skill; but time and chance happeneth to them all.”

There are powerful, if little discussed, counterforces that are working against Web services. More than one promising technology has withered from lack of interest because the politics of the situation undermined it. The idea of you sharing with me is more appealing than its necessary converse, and whether the dreaded “not invented here” syndrome will defeat Web services is anyone's guess. Soothsaying in all aspects of life is bad business.

If Web services do succeed, I think they will do so because they take a very different approach than that of previous attempts to solve the same problem. Web services have the startling, unusual virtue of simplicity. It's startling because so many technologies plunge into complexity without ever looking back. Why? Why this love of complexity? Despite numerous failures where complex solutions are attempted, our industry seems drawn to complexity like a moth to the flame – and often with the same results.

The movers and shakers of technology love gadgets. I suspect most of us do – complicated gadgets.



MACROMEDIA

www.macromedia.com/go/usergroups

They're fun and they engage our attention. We're told that this newest tool or method or model or technology will solve problems that were too tough for simpler means. The new product's abilities are touted as far superior to marginally older, simpler tools.

Yet after a time a pattern emerges: a technology is introduced...we scramble to learn it...hopefully we master it. Or we at least try. The truth is, we're probably a bit behind the curve, still struggling to recover from several product cycles back when the latest, greatest technology demanded our attention. But who can argue that newer is better? That more complex is better? Surely we don't wish to be Luddites, opposing promising new technologies...

But the promoters of the new technology are often the ones who, just a short time ago, were touting the benefits of what they now wish us to set aside, leaving us on a path of faster and faster change. Is change always for the better? What's really going on?

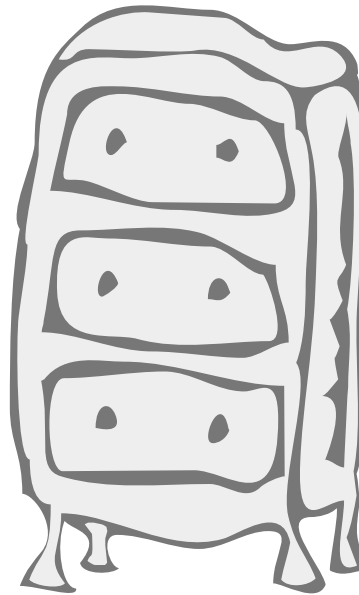
The first, most obvious, answer is profit. If I sell you product A and you like it, find it useful, and master it, that's good for you. But enough about you; what about me? How can I stave off my competition, to say nothing of meeting Wall Street analysts' famously short profit expectations, by *having* sold you a product? I need to sell you a new one to keep the cash flowing. In order to sell it, it needs more features, more capabilities, and, of course, it must be 100% buzzword compliant.

So I introduce – hopefully to great fanfare – product B. Product A, which replaced an older, now forgotten product, is *passé*; product B is what all the most fashionable developers are using. I don't use the word *fashionable*, of course. I prefer words like *competitive* or *cutting edge* or *a breakthrough in technology*. We're all used to this cycle. We accept it. Sometimes we embrace it. And we almost always accept the idea that the more complex a technology is, the better it is. It's a notion that has surprisingly little evidence to support it.

New technologies have learning curves associated with them. The more complicated, the longer the

learning curve. The time required to learn a new technology takes time away from actually producing useful items with it, a calculation we're seldom encouraged to make.

That may explain why people want to sell a newer, more complicated technology every week. Why do we so often accept their questionable proposition? I think there are at least four reasons that lead us to accept added complexity with only the flimsiest of justifications.



First is the misplaced notion that we're failures if we want to get off the technological merry-go-round. If we perceive that we are "real developers" only in proportion to our ability to play this high-tech version of "Chicken," we will forever be at the mercy of vendors' desires to sell us newer versions of their wares. Our significance as developers lies in our ability to fashion solutions to clients' real-world problems. We are developers, not mere technology consumers.

Second is simple boredom. The more complicated things are, the more there is to learn, to do. A new technology can be just the antidote to more of the same. Actually, I'm quite sympathetic to this. Few of us want to do the same thing over and over, but a better solution is to go deeper into the craft of programming, to learn how to make code simpler and cleaner, rather than acquiring a new tool to play with.

Third may be the misperception that simple tools won't help us achieve the rich, complex work we want to do. This myth isn't unique to technology. Years ago George Orwell lamented the trend to set aside simple, English words in areas of business, academia, and government in favor for more complex words and phrases, often Latin in origin.

Commenting on the example from *Ecclesiastes* I quoted above, Orwell says that the majestic cadences and simple prose of the original translation would today be rendered thus: "Objective considerations of contemporary phenomena compel the conclusion that success or failure in competitive activities exhibits no tendency to be commensurate with innate capacity, but that a considerable element of the unpredictable must invariably be taken into account."

Such a confusion of complexity and sophistication is a stage of understanding many of us go through and a few get stuck in. Years before personal computers came on the scene, I was a cabinetmaker. I worked alongside craftsmen who were highly accomplished and who had a bevy of tools and machines to help them work their wonders. I wanted to be just like them. They took a rough piece of wood and fed it through a (power) surface planer to make it smooth. They produced a wonderfully straight edge by using a (power) jointer, then turned that wood to a (power) table saw to cut it to the dimensions they needed.

But the owner of the shop was a crusty Cape Codder who had some very peculiar notions. For the first year of my apprenticeship I wasn't allowed to use a power tool, to say nothing of the wondrous machines the shop was filled with. When I needed to fashion a piece of wood, I took a rough piece of wood and a hand plane.

If you've never tried this, I can only assure you that it is a very difficult task. After hours of assaulting the wood and thinking that I – finally – had the knack of it, I'd find that I had planed one end of the wood thinner than the other. Or it was a different thickness across its width. If I was working with a piece of hard

Despite numerous failures where complex solutions are attempted, our industry seems drawn to complexity like a moth to a flame – and often with the same results”

wood – oak, say, or cherry – the plane blade dulled quickly and had to be sharpened often.

What I was learning, though I had no idea of it at the time, was that a tool, properly used, was an extension of my hands. Over time I developed the ability to sharpen and hone a plane to the point where I could shave with it. I learned how to read the grain in a piece of wood to see how to best work it. Later, when I “graduated” to using power tools and machines, I marveled at the wisdom of my boss, who was willing to invest so much to teach me the meaning of craftsmanship. I found

that often the best tool for the job was a simple hand tool.

I also noted that people who began with complicated tools often never developed real mastery of their craft. They were so busy fiddling with new router bits or shaper blades (the woodworking world, too, has its share of “kewl” stuff) that they never developed a sense for the work itself. Their knowledge began and ended with what the tool could do, and their work was less the product of their creativity than of the limitations of the tool. They had fallen for the fallacy that simple tools were good only for simple work.

The scope of their solutions to problems was limited to what the tool could do, not what they could do.

When I moved to the high-tech world, I was surprised to find that many technology workers were also defined by their tools and technology. I was marvelously lucky to have someone who insisted that I learn the craft of woodworking so that I could use my tools and not be defined by their bounds.

I was equally fortunate to begin my programming career working with experienced developers who believed that programming is not primarily a matter of arrays and data types. It is more fundamental. Programming lets us build self-consistent worlds of our own fashioning. It is an act of great creativity. Viewed thus, programming can be a source of deep satisfaction and enjoyment. We have a rich path to follow, one that is defined by our selves and not our tools.



@HAL.HELMS@TEAMALLAIRE.COM

ABOUT THE AUTHOR

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

HOSTMYSITE.COM

www.hostmysite.com

How to Sidestep Locking

BY
SANDY CLARK
AND
MICHAEL SMITH



Everywhere we turn there's another article, another e-mail, and another best practices – all of which entail locking.

According to Macromedia: "If you are not locking your shared scoped variables, your application *will* fail under load." We're not going

'They' say you should lock, but you don't really have to...

We all know we should lock our shared scoped variables – that is, application, session, and server variables that can be shared between multiple ColdFusion threads.

to get into why you should lock; there are enough excellent articles that cover that aspect. Rather, we're going to give you some quick and easy ways to get around it.

Locking Required

Get around locking? How can that be? We *have* to lock our variables. Macromedia and a lot of other people say we do. Well, not entirely – the basic premise of locking shared scoped variables still holds, so the answer is: "Don't use shared scoped variables."

This article takes a look at the problems with locking shared scoped variables and explains how you never have to explicitly lock them again. It also explains how to make session variables truly clusterable without pain.

Application Variables

We've all used application variables for what we call *global constants*, constants that are set once in Application.cfm and are then used throughout our program. Application.datasource or Application.dsn come to mind. When we consider that not only do we have to lock all the writes to this scope, but all the reads as well, having every <CFQUERY> use the Application.datasource variable suddenly doesn't look very good.

So stop! Throw out all application variables...well, don't throw them away, but change them to a Request-App structure. Using this method of coding, Request.App becomes a global constant, which is always available. You can also use other request structures for other types of global constants: Request.Site for site mappings and variables for paths,

Request.Page for page constants. You get the picture. When you place every global constant into the Request-App structure within your Application.cfm, you still have a very visual way of noting what your global constants are, but you've also totally eliminated the need for locking.

Server Variables

To be honest, most people don't use server variables, which are like application variables, but are available to all applications named on a particular server. Some people suggest storing query recordsets, such as all states, into a server-scoped variable. But again, that requires a lock to read as well. If you're using server scope to store queries, cache the queries instead, using the cached-within attribute of the <CFQUERY> tag. One good use for server variables we saw recently was for caching things like CFDIRECTORY and CFPOP.

Session Variables

Now it starts to get interesting. The biggest problem with shared scope and locking we developers experience is when using session scoped variables. What would you say if we told you that, using one variable, your developers would never have to write to the session or client scope again, and that at the same time you could switch your applications from session-based to client-based state management? Interested? Keep reading.

Scenario One: Lazy Developers

We've all worked with "Joe" (the name has been changed to protect the innocent). Joe's a good guy, and a fairly good developer, but he's lazy.



REQUEST SCOPE

The request scope was introduced in ColdFusion 4.5. It is the same as the regular page variables scope, but can also be seen inside custom tags called from that page.

ABOUT THE AUTHORS

Sandy Clark, an advanced Macromedia-certified ColdFusion developer, SCCFUG speaker, and author, is president of Shayna Productions (www.shayna.com), a Maryland-based ColdFusion consulting company.

Michael Smith is president of TeraTech (www.teratech.com), a Rockville, MD-based consulting company specializing in ColdFusion, database, and Visual Basic development. Michael runs the MDCFUG and organized the CFUN-2K conference.



Programmers who use this technique rapidly forget there was ever such a thing as client or session variables"

Someone needs to look at Joe's code a lot since Joe hasn't been fully convinced yet of the necessity of locking his session variables. While we can (and certainly should in a development environment) turn on the CF Admin option that locks all scopes so it always throws errors, we want some better way to eliminate worry about this.

Scenario Two: Clustering Complex Session Variables

You've spent months working on an application that went live last month. You've used a lot of complex structures in your session variables and the app is smoking. Suddenly a lot of users access your application and your boss wants to cluster it.

The problem? Since you've written the application using session variables, you have only two options:

1. Implementing sticky sessions on your load balancer (which not only will keep all your users on their original server, but is exactly what your boss doesn't want).
2. Going into your code and replacing all session variables with client variables. Keep in mind that you can't really just run a search-and-replace since client variables can't store complex data structures except through WDDX.

The situation described is exactly what we came up against on an application. We had a junior developer who wouldn't or couldn't understand about locking variables; three quarters of the way through the project the powers that be decided we had to make them clusterable. We needed a solution that would solve both problems.

The solution was a custom tag, `<CF_REQUEST>`, that takes one attribute, `Type`, which can be set to "read" or "write." It also takes a request scope flag (`Request.StorageType`) that is set and used in `Application.cfm` when you use the `<CFApplication>` tag (see Listing 1).

By setting the `Request.StorageType` variable, we could literally pick and choose whether to run a session-state or a client-state management application. If client scope is required, set up `Request.StorageType` to equal `Client`, which will in turn run the `<CFApplication>` tag that sets up the client-state management.

So much for worrying whether the application needed to use session or client variables in the app. By using one variable, the problem was solved.

Making Request Scope "Permanent"

Then it was time for the more interesting problem. We knew we had to use the request scope for all the reads and writes within the application, but also knew that request scope holds variables only for the page calls and can't store the information. So we had to write the request scope variables to either client or session scope depending on the `Request.StorageType` flag.

As stated above, `<CF_REQUEST>` takes one attribute, "Type," which can be set to either read or write. Depending on the attribute and the `Request.StorageType` variable set in `Application.cfm`, the request scope can be used in all application development and the program will determine the best way to save the variables between page calls (see Listing 2).

SAVE 30% off the annual newsstand rate

JAVA DEVELOPER'S JOURNAL

Offer subject to change without notice

ANNUAL NEWSSTAND RATE	
\$71.88	
YOU PAY	
\$49.99	
YOU SAVE	
30%	Off the Newsstand Rate

DON'T MISS AN ISSUE!

Receive 12 issues of *Java Developer's Journal* for only \$49.99! That's a savings of \$21.89 off the cover price. Sign up online at www.sys-con.com or call 1 800 513-7111 and subscribe today!

In April JDJ:

Pervasive Computing: The Next Generation of Consumer Applications

Now that broadband connections are becoming commonplace and wireless technology is becoming a reality, developers should begin positioning their applications to leverage the new capabilities.

J2EE Application, Module and Component Packaging

The different types of J2EE modules and how they fit into the J2EE architecture.

JDiff - What Really Changed?

JDiff is an open source Java tool, based on Javadoc and developed by the author, that produces HTML documentation describing the precise API changes between two versions of a product.



A switch statement based on the Request.StorageType variable reads either the client scope or the session scope into a temporary variable. The tag then loops through the temporary variable structure and makes sure that the structure being copied doesn't belong to a global constant or a read-only variable (after all, these are always available and don't need to be saved between page calls). At the end, the temporary structure is deleted and there is now a copy of either the client structure or the session structure in the request scope (see Listing 3).

The write portion of CF_Request works in much the same way, except that instead of reading to the request scope, it writes from request scope to either session scope or client scope. The write also makes sure that global constants don't get written to the session or client scope by comparing the structure against the nocopy list.

The calls to the custom tag need to come in three places. In Application.cfm, right after the implementation of the CFApplication tag, the following call should be inserted:

```
<cf_request type="read">
```

The write portion (<cf_request type="write"> has to be called twice. It needs to go into your onRequestEnd.cfm so that all variables called and written during a page call can be placed in the proper scope. It also needs to go just before any call to <CFLOCATION> – this is because any call to CFLOCATION won't run onRequestEnd. For simplicity's sake, we used the custom tag <CF_Location> (found on the fusebox.org site) and added the call into this tag just before the actual location took place.

• • •

Are there any problems with this system? We haven't found any that

relate technically. Reading and writing to the various scopes happens very quickly. The only problem we found is that programmers who use this technique rapidly forget there was ever such a thing as client or session variables. In retrospect, that hasn't been so bad...

Resources

- *Locking in CFML*: www.depress-edpress.com/DepressedPress/Content/ColdFusion/Guides/Locking/Index.cfm
- *ColdFusion Locking Best Practices*: www.macromedia.com/v1/handlers/index.cfm?ID=17318&Method=Full&Cache=Off
- *<cf_request>*: www.shayna.com/index.cfm?fuseaction=coldfusion

@_SLCLARK@SHAYNA.COM

@_MICHAEL@TERATECH.COM

Listing 1: Application.cfm

```
<cfparam name="Request.StorageType" default="Session">

<cfswitch expression="#Request.StorageType#">
  <cfcase value="Client">
    <cfapplication name="DemoApp"
      clientmanagement="Yes"
      sessionmanagement="No"
      setclientcookies="Yes"
      clientstorage="StorageDB">
```

```
</cfcase>
<cfcase value="Session">
  <cfapplication name="DemoApp"
    clientmanagement="No"
    sessionmanagement="Yes"
    setclientcookies="Yes"

    sessiontimeout=#CreateTimeSpan(0,0,20,0)#>
  </cfcase>
</cfswitch>
<CF_Request type="read">
```

RECEIVE \$150
DISCOUNT OFF FULL CONFERENCE
WEB SERVICES EDGE REGISTRATION

web services **EDGE**
world tour 2002

**Learn How to Develop
SOAP Web Services NOW!**
at a One-Day Seminar...Coming to a City Near You!


```
<cfparam attributes.type="read">
```

```
<cfparam attributes.type="read">

<!--- Set up list of structures that are global constants
and don't need to be written --->
<cfset nocopy="page,app,site,cfid,cftoken,sessionid,urlto-
ken,checklastvisit,StorageType">

<cfswitch expression="#attributes.type#">
  <cfcase value="read">
    <!--- Now choose whether to read from Client or Session
Variables --->
    <cfswitch expression="#Request.StorageType#">
      <!--- If we are using Client Variables. --->
      <cfcase value="Client">
        <!--- Read the structure from the Client Variables
using WDDX --->
        <cfwddx action="WDDX2CFML" input="#Client.Request#"
output="Temp">
      </cfcase>

      <!--- If we are using Session Variables. --->
      <cfcase value="Session">
        <cflock scope="SESSION" type="ReadOnly" timeout="30">
          <cfset Temp=Duplicate(Session.Request)>
        </cflock>
      </cfcase>
    </cfswitch>

    <cfif isDefined('Temp')>
      <!--- Now copy temporary variable to Request Scope --->
      <cfset keystostruct=structkeyarray(temp)>
      <cfloop index="i" from="1"
to="#ArrayLen(keysToStruct)#">
        <!--- As long as the temporary variable is not one of
the no copy structures. --->
        <cfif listfindnocase(nocopy, keystostruct[i])>
          <cfset "Request.#keysToStruct[i]#" =
#duplicate(temp[keystostruct[i]])#>
        </cfif>
      </cfloop>

      <!--- Clear Temp Structure --->
      <cfset Temp="">
    </cfif>
  </cfcase>
</cfswitch>
```

```
<!-- Set up list of structures
```

```
<!-- Set up list of structures that are global constants  
and don't need to be written --->  
<cfset nocopy="page,app,site,cfid,cftoken,sessionid,urlto-  
ken,checklastvisit,StorageType">  
  
<cfswitch expression="#attributes.type#">  
  <cfcase value="write">  
    <!-- Grab the keys to the request structure. --->  
    <cfset keystostruct=structkeyarray(request)>  
  
    <!-- Set up a temporary structure. --->  
    <cfset Temp=StructNew()>  
  
    <!-- Loop through the Request Structure. --->  
    <cfloop index="i" from="1" to="#ArrayLen(keysToStruct)#">  
      <!-- As long as the request variable is not one of the  
no copy structures. --->  
      <cfif listfindnocase(nocopy, keystostruct[i])>  
        <!-- Copy that particular structure to the temporary  
structure using the duplicate() function --->  
        <cfset "Temp.#keysToStruct[i]#" =  
#duplicate(request[keystostruct[i]])#>  
      </cfif>  
    </cfloop>  
  
    <!-- Now choose how to save the structure --->  
    <cfswitch expression="#Request.StorageType#">  
      <cfcase value="Client">  
        <!-- Write the structure to Client Variables using  
WDDX --->  
        <cfwddx action="CFML2WDDX" input="#Temp#"  
output="Client.Request"> --->  
      </cfcase>  
  
      <!-- If we are using Session Variables. --->  
      <cfcase value="Session">  
        <cflock scope="SESSION" type="Exclusive" timeout="30">  
          <cfset Session.Request=Duplicate(Temp)>  
        </cflock>  
      </cfcase>  
    </cfswitch>  
  
</cfcase>  
</cfswitch>
```

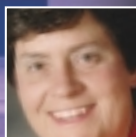
COD
LISTING
■■■■■■■■■■■■■■■■■■■■

The code listing for

**Jump-start your Web Services knowledge.
Get ready for Web Services Edge East and West!**

AIMED AT THE JAVA DEVELOPER COMMUNITY AND DESIGNED TO EQUIP ATTENDEES WITH ALL THE TOOLS AND INFORMATION TO BEGIN IMMEDIATELY CREATING, DEPLOYING, AND USING WEB SERVICES.

EXPERT PRACTITIONERS TAKING AN APPLIED APPROACH WILL PRESENT TOPICS INCLUDING BASE TECHNOLOGIES SUCH AS SOAP, WSDL, UDDI, AND XML, AND MORE ADVANCED ISSUES SUCH AS SECURITY, EXPOSING LEGACY SYSTEMS, AND REMOTE REFERENCES.



PRESENTERS...

Anne Thomas Manes, Systinet CTO, is a widely recognized industry expert who has published extensively on Web Services and service-based computing. She is a participant on standards development efforts at JCP, W3C, and UDDI, and was recently listed among the Power 100 IT Leaders by Enterprise Systems, which praised her "uncanny ability to apply technology to create new solutions."

Zdenek Svoboda is a **Lead Architect** for Systinet's WASP Web Services platform and has worked for various companies designing and developing Java and XML-based products.

EXCLUSIVELY SPONSORED BY



systinet

BOSTON, MA (Boston Marriott Newton) . **SOLD OUT!** . **JANUARY 29**

WASHINGTON, DC (Tysons Corner Metro) **SOLD OUT!** **FEBRUARY 26**

NEW YORK, NY (Doubletree Guest Suites).... **NEW DATE!****APRIL 19**

SAN FRANCISCO, CA (Marriott San Francisco) **MAY 13**

REGISTER WITH A COLLEAGUE AND SAVE 15% OFF THE \$495 REGISTRATION FEE.

Register at www.sys-con.com or Call 201 802-3069



WebServices
JOURNAL

XML JOURNAL

THE FIRST & ONLY
WEB SERVICES
RESOURCE CD

WEB SERVICES RESOURCE CD

THE SECRETS OF THE WEB SERVICES MASTERS

INCLUDES EXCLUSIVE .NET ARTICLES

MORE THAN

400
EXCLUSIVE

WEB SERVICES
& XML
ARTICLES



EDITED BY
SEAN RHODY

\$119

CD

VALUE

FROM
WEB SERVICES
JOURNAL

Web
services
conference & expo

EDGE

\$100

coupon inside

EVERY ISSUE OF WSJ & XML-J EVER PUBLISHED

THE MOST COMPLETE LIBRARY OF EXCLUSIVE WSJ & XML-J ARTICLES ON ONE CD!

"The Secrets of the Web Services Masters"

CD is edited by well-known WSJ Editor-in-Chief
Sean Rhody and organized into more than 40 chapters
containing more than 400 exclusive WSJ & XML-J articles.

Easy-to-navigate HTML format!

Bonus:

Full .PDF versions of every WSJ & XML-J published
since the first issue

XML in Transit
XML B2B
Java & XML
The XML Files
XML & WML
Voice XML
SYS-CON Radio
XML & XSLT
XML & XSL
XML & XHTML
2B or Not 2B
XML Script

XML Industry
Insider
<e-BizML>
XML & Business
XML Demystified
XML &
E-Commerce
XML Middleware
XML Modeling
CORBA & XML
Data Transition
XML @ Work

XML &
Databases
Electronic Data
Interchange
Ubiquitous
Computing
Information
Management
Objects & XML
XML Pros & Cons
Java Servlets
XML Filter

UML
Integration
WSDL
Beginning
Web Services
Web Services
Tips & Techniques
Frameworks
Management
Security
UDDI
.NET

3
YEARS
25
ISSUES
400
ARTICLES
ONE CD



Special Limited Time Price

Now
Shipping

\$79

+ S&H

ONLINE
ORDER AT
JDJSTORE.COM
SAVE
\$40

WWW.JDJSTORE.com

OFFER EXPIRES JUNE 30, 2002

Back to Basics:

When sites run into trouble, it's usually the database that's the problem, not ColdFusion

[Ed. Note: We've run articles on this in the past, but – we can never say it enough!]

Cutting ColdFusion code is fun, and therein lies ColdFusion's biggest problem. It's too easy to get carried away writing code to do things that should be happening in the database itself – or often shouldn't be happening at all. It's also just too easy to use. You can get up and running far too quickly, and get into trouble even quicker. Sometimes experienced developers get carried away trying to do too much; beginners often start to run before they can walk.

When I came across a site where a simple query of news stories was taking a quarter of a second, I wasn't sure that writing their own object-caching mechanism in ColdFusion was entirely the right approach. I called a timeout – a morning to go over the basics. One simple run of the Query Analyzer diagnosed the problem: a date field was being used in a sort clause without an index.

The query time was reduced to around 15ms – around 6ms database time plus the ODBC overhead. If you use SQL Server, the Query Analyzer is a tool you should learn to love. As a basic rule, any query that does more than select a single record using a primary as the criterion will probably benefit from some sort of index.

Using it couldn't be simpler – simply open a window, enter your SQL query text and then press Ctrl + I to run an index analysis (see Figure 1). In SQL Server versions 7 and up it's become an invaluable tool. You can also take a look at the path analysis and turn on the "Show stats time" option in the Query->Current Connection Options dialog.

It's useful, but in a publishing application there's only so much you can do to tune database performance. You can get too worried about optimization – you're pushing about large text strings and that takes time. Hard-core optimization is more relevant in financial and engineering applications where the database is summarizing and calculating vast numbers of small records.

You're more likely to achieve better results if you concentrate your efforts on your ColdFusion code. ColdFusion offers two powerful ways of caching queries: the "cached within" feature and caching in a shared scope. Many developers aren't aware that you can cache any ColdFusion object in the application scope; structs and queries can just as easily be cached as scalar variables. You can cache queries that are unlikely to change in application.cfm:

```
<cfif NOT
IsDefined("application.courses")>
    <cfquery name="courses" data-
source="#application.dsn#">
        SELECT * FROM COURSES
        ORDER BY COURSE
    </cfquery>

    <cflock scope="application"
type="exclusive" timeout="10">
        <cfset application.courses =
courses>
    </cflock>
</cfif>
```

(The locking after the query is vitally important; see article 20370 in the ColdFusion knowledgebase.)

By putting the actual query and assignment above into a separate include file and calling it whenever the table is affected, you can create your own full-fledged cache server. Alternatively, you can convert the query to a struct keyed by the primary key of the database table and put the struct into the application scope. There are custom tags in the developer's exchange to do this, but it's almost quicker to do it yourself.

Don't be scared of using the application scope. Memory's cheap and downtime isn't. If your entire database is only half a gigabyte and you have 2GB in your server, why not load it into the application scope?

Usually it's best to mix this approach with the cached within approach. Say you have a database of two thousand suppliers, listed by region, together with the services they offer. You want to use the "startrow, endrow" parameters of <CFOUTPUT> to show fifty at a time, but this means fetching the entire listing in the query, which takes nearly 150ms.

In this case you could put the fields you need for the listing in an application-level query and then, on a separate "view" page, select the main details in a separate query that you apply the cached within parameter to. This parameter of <CFQUERY> is one of ColdFusion's best features. It can send your sysadmin into a blind panic when he or she sees CF taking up 800MB ("It's got a leak!"), but it's so simple to use compared to rival offerings.

One technique I use all the time, caching formatted output, is a little more contentious. Say, for instance, you have fields in your supplier database for TEL, FAX, EMAIL, WEB – any of which may be blank or need formatting. In a listing page of fifty records this can take time. On one of my pages it was taking too much time just to do a simple list. My solution? Rather than copy the whole query into the application scope, I created a new query (see the Query-New() function) with fields for the primary key and formatted display.

Even more contentious: sometimes I put the formatted output back into the database. I do this all the time where the database is storing XML and I need to convert to HTML. Converting a page on the fly takes around 400ms (assuming XSL doesn't get stuck in an endless loop) and displaying a cached query takes less than 1ms.

This may sound like a lash-up (I less than impressed an old college friend who followed a slightly more formal career), but this is how the really expensive application servers work. Sure, they won't store large chunks of HTML in the database; they'll store serialized Java objects, but at least the HTML will still work when somebody changes the code. (Application servers store objects in a format that depends on the object definition [your code]. When you change that definition, the entire object store can be rendered unusable. It's very problematic, and I've seen plenty of implementations of expensive object servers where almost all the data is stored in plain SQL tables.)

Don't be scared of using the application scope. Memory's cheap and downtime isn't. If your entire database is only half a gigabyte and you have 2GB in your server, why not load it into the application scope?"

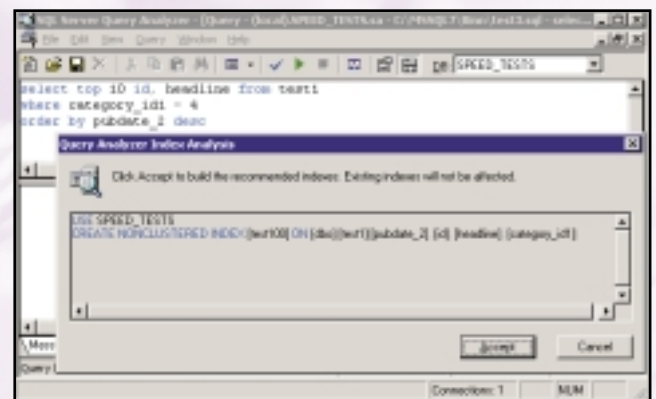


FIGURE 1: SQL Server Query Analyzer recommending an index

Note: With reference to Ben Forta's February **CFDJ** article ("Faster and Safer Database Queries," Vol. 4, issue 2), you can't cache a query that uses a QUERY-PARAM tag. You can, though, achieve the same security against malicious URLs by

ensuring that all your tables have a numeric primary key and then validating any variables used in SQL queries with an "Is-Numeric()" test. It's one more reason why all database tables should have a numeric primary key.

No matter how cheap memory gets, you do still need to apply a certain amount of discipline to your coding. One of the most common causes of database problems

is fetching too much data from the database. Using derived tables and other advanced SQL constructs, you should be able to avoid fetching vast amounts of query data and looping over it.

A typical derived table query looks something like this:

```
SELECT TOP 1 A.NEWS_ID, A.HEADLINE,
A.STORY FROM NEWS AS A,
(SELECT NEWS_ID FROM NEWS_CATEGORIES_JOIN
WHERE CATEGORY_ID IN (#cat_list#)) AS B
WHERE A.NEWS_ID = B.NEWS_ID
ORDER BY A.PUBDATE DESC
```

This will select a record from a correctly built many-to-many join (of which much more later) when a GROUP BY can't be used (i.e., with long text fields).

This is just a simple example (a subselect could be used equally well here and run as quickly); derived tables are most effective when some sort of calculation needs to be performed, perhaps a summation or a count. If you find the idea of working with tables that don't exist a little frightening, you can often achieve the same affect by using a view, a stored query that

can be queried and joined just like a standard table (see Figure 2).

Often underused, views are a great tool for making both your SQL and your CF code as simple as possible. They're second nature to anyone who's come to SQL Server from Access (where they're known as *queries*), but experienced programmers who have been using old-style databases often prefer to do everything in one query. Using views is usually quicker than performing joins or summaries at runtime. I saw a job advertisement recently that said "must be able to write join queries without using visual tools" and I thought: "Why?" There's no benefit, save a few techy kudos.

If you do need to loop over entire tables, try to avoid fetching the data into ColdFusion. Database cursors (described by Ian Rutherford in "Using MS-SQL Stored Procedures with ColdFusion," **CFDJ** Vol. 4, issue 2) will run much faster than dragging data into ColdFusion, processing it, and then running an update. It also won't take up a ColdFusion thread or valuable network capacity.

Speed Isn't Everything

As important as optimization is, sometimes there's too much focus on saving a millisecond here or cutting down white space there at the expense of properly structured code and databases. The key to software engineering, as in all engineering, is to keep it simple, and the main aim in a ColdFusion development should be no different from any other project: to keep the code to a minimum.

It's this discipline of keeping things simple that's the most important part of building reliable and scalable sites. It may seem strange, but sometimes code that runs slower can make for better Web sites. Take, for example, what for me is the litmus test of good database design: the use of many-to-many joins. A typical database application might be a listing of schools and the courses they offer. The original data, as always, is in Excel, looking something like the example in Figure 3. The experienced developer will recognize the need for a separate table of courses and a third "join" table indicating which schools have which courses.

The temptation, though, is simply to import it into the database and use it as is, relying on ColdFusion's list handling to determine which courses are offered by which schools. That would be bad, but what would be even worse would be some attempt to code the courses offered as separate database fields (see Figure 4).

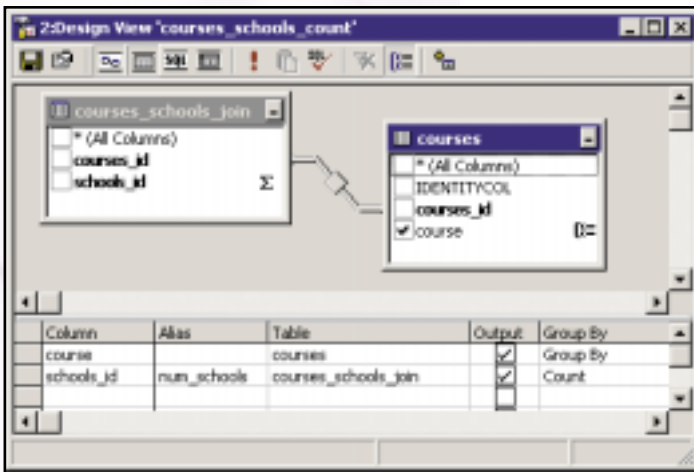


FIGURE 2: Creating a view in the Enterprise administrator

No matter how cheap memory gets, you do still need to apply a certain amount of discipline to your coding"

	A	B	C	D
1	SCHOOL	ADDRESS	WEB	COURSES
2	AAA Flying	Biggin Hill	www.aaaflying.com	PPL/JR/NT, VTOL
3	AAB Flying	Gatwick	www.aabflying.com	PPL/JR
4	AAC Flying	Heathrow	www.aacflying.com	NT

FIGURE 3: Typical raw data

FIGURE 4: "Hard-coded" data used as field names

Why you shouldn't do this is hard to explain, although it's simple: in a SELECT query it'll run two to three times as fast as using a many-to-many join, and it's easy to program using ColdFusion Studio. You just point the wizards at the right table and five minutes later you're done. But by doing something like this, you have hard-coded "data" into the "structure" of your system. To build truly scalable systems, you need to keep a very definite division between the structure and the data. The Golden Rule is this: if your ColdFusion code is completely separate from your database, then your system will scale. If you've hard-coded data as fieldnames – you just can't do this – your data is inextricably tied up with your application.

Using ColdFusion, it isn't always possible to adhere to the strict three-tier model (database, application, presentation), but creating a division between the structure and the data is still a must. You should be able to edit the data directly without using your ColdFusion application and still see the changes reflected on your site.

So, for instance, were you to add a new course to the table of courses, your site should display that course wherever relevant. Often this doesn't happen; data fields are hard-coded and the layout of the pages is dependent on the data structure.

Sometimes you might use the ColdFusion application to enforce certain database rules – for instance, that a salesperson can have a maximum of only 10 customers. This is fine. In fact, database theorists call such a rule an *application* rule for the very reason that it doesn't affect the underlying data. If a salesperson had 11 clients it wouldn't affect the data in any way. But if the only way you could add or remove clients was through the ColdFusion-based Web interface, then something would be amiss.

A simple example like the one shown in Figure 4 doesn't really illustrate the necessity of relational design. Once you're bug-free, the example will run faster than a relational design, and it will scale. It's simple and, above all, that's what matters.

You wouldn't have to get much more complicated before a flat file system like this would start to break

down, though. It's not just raw performance that's the issue – the extra complexity of the code and the difficulties in making changes will all contribute in different ways to a lack of scalability. If a system relies on a single overworked developer to make simple changes, then it won't be scalable no matter how optimized the code.

As with any rule, the one about keeping the data separate from the code is easily broken. It's quite possible to make such a convoluted mess of triggers, derived tables, cursor loops, and other database constructs that no amount of processing power will save you. Keep it simple, though, and it will scale.

Easy in Theory

Although it makes things simpler in the database system, relational design can, if you're not careful, make things more complicated in the Web application. Using separate tables for many-to-many joins requires extra INSERT queries, and SELECT functions often have to employ grouping, subselects, or derived tables.

Perhaps the principal reason many-to-many joins don't get used is that most database editing systems don't have a standard component for implementing them. Access doesn't (you have to create a subform or use some VB), and if you're using HTML forms, you unfortunately can't use the "query" attribute of <CFSELECT> as it doesn't support multiple values for "selected". You can, however, use a simple loop:

```
<cfselect name="course_id" multiple="Yes">
    <cfoutput query="application.courses">
        <option
#iif(ListFind("#current_vals#",
courses_id),
    DE("selected"), DE(""))#
value="#courses_id#">#course#
    </cfoutput>
</cfselect>
```

Alternatively, you can use checkboxes; another effective device for editing many-to-many joins is a swap box. It's especially useful when you have many possible options to choose from, and it's more intuitive than multiple select boxes where the user has to control-click (see Figure 5).

SAVE 16% off
the annual cover price

ColdFusion Developer's Journal

ANNUAL NEWSSTAND RATE

~~\$107.88~~

YOU PAY

\$89.99

YOU SAVE

\$17.89 Off the Annual
Newsstand Rate

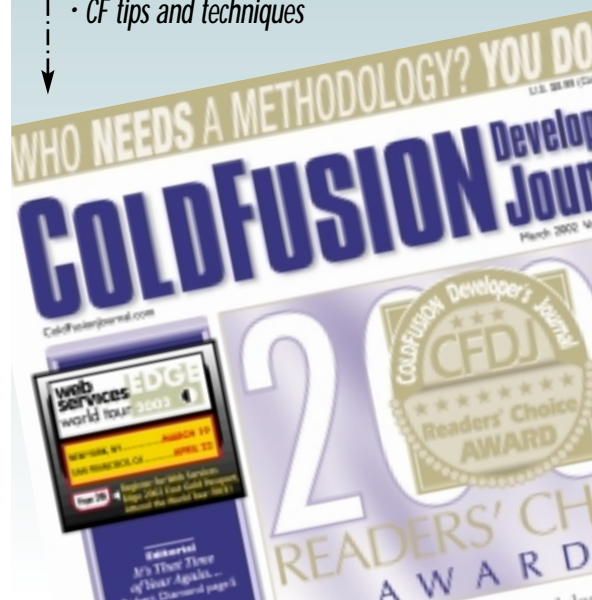
Receive 12 issues of
ColdFusion Developer's Journal
for only \$89.99! That's a savings of
\$17.89 off the annual newsstand rate.

Visit our site at www.sys-con.com

or call 1-800-513-7111
and subscribe today!

Here's what you'll find in every issue of ColdFusion Developer's Journal

- Exclusive feature articles
- Interviews with the hottest names in ColdFusion
- Latest CFDJ product reviews
- Code examples you can use in your applications
- CF tips and techniques



SUBSCRIBE AND SAVE

XML JOURNAL

Offer subject to change without notice

ANNUAL NEWSSTAND RATE	
\$83.88	
YOU PAY	
\$77.99	
YOU SAVE	
\$5.89	Off the Newsstand Rate

DON'T MISS AN ISSUE!

Receive 12 issues of **XML-Journal** for only **\$77.99!** That's a savings of **\$5.89** off the annual newsstand rate.

Sign up online at www.sys-con.com or call 1 800 513-7111 and subscribe today!

In April **XML-J:**

Deploying Web Services on WebSphere

A real-life example, using the tools and services provided

ADO.NET & ASP.NET...

...How to use them to build XML applications

.NET Web Services: The 'Three I' Monster

Web services...middleware for the masses...XML miracle tools...instant integration: just add SOAP

Middle-Tier Data Management

The middle tiers in the XDBMS architecture can be confusing

A Pragmatic Convergence of ebXML & Web Services

Is this a second chance for the industry?

XML - Beyond Transport

Persistence for Web services



There are several custom tags to do this in the developer's exchange, and my own `<CF_SWAPBOX>` is available from my site (see "Resources" section). It has almost identical syntax to `CFSELECT`, and also supports multiple values for "selected".

One thing you can't do much about is the complexity of inserting and updating the "join" tables (the third table used for many-to-many joins). You can cut down the number of queries needed by using an "insert into" with a subselect, for example:

```
INSERT INTO COURSES_SCHOOLS_JOIN
(COURSES_ID, SCHOOLS_ID) SELECT
COURSES_ID, #form.SCHOOLS_ID#
FROM COURSES WHERE COURSES_ID IN
(#form.courses_id#)
```

but this won't run any faster than using multiple inserts in the same query. Nor can you run this as a stored procedure (you can't pass a list of integer primary keys as a `VAR-CHAR` and then use it in a subselect), nor can you use the `<CFQUERY-PARAM>` tag. It's best just to run a separate insert for each row of the join table.

Perhaps, though, the biggest obstacle to correct database design isn't a technical one, but the very familiar problem of client demands. Before you can build a sound database, you need to convince the client that that's what's needed. The temptation for clients is to think of their Web site like a magazine: "I want to put that over there, that over there." Convincing clients they need to "think data" is the hardest part of any project.

In the most difficult project I ever worked on, the editor of the site was so incensed at no longer being able to work on the HTML pages himself that he got hold of the administrator password and replaced all the ColdFusion code with HTML pages he'd "saved as" from his browser. All the dynamic functions duly broke, but because the site still looked the same as it did before, the client never really understood what had happened. As far as he was concerned, my code didn't work. The editor was there to edit pages and that was what he was doing.


The concept of changing a page not by editing the page itself, but by



FIGURE 5: A swapbox control

editing a separate data system, is a surprisingly difficult conceptual hurdle for many people. It's such a simple concept that it's easy to forget how revolutionary it is. Those who have come from a print media background aren't going to buy into the database idea until they've had that hallelujah moment and seen just what an impact information systems can have. Until that time they're going to find using a database frustrating and too inflexible.

If your client isn't of the database mind-set when you start a project, it's vital that you help him or her see the light before you start programming. Otherwise you'll end up putting in hooks and hacks to try and satisfy client demands ("sometimes we might want to do something different"). They'll come back to haunt you.

I've found that the most important ingredient of a successful site isn't the design or my code, but getting the client used to the idea of editing a database. A client who comes to you not with ideas for "new pages" but a "new section" and a sketch of the underlying data is worth more than any amount of milliseconds saved from a database query. 

Resources

- To download `<CF_SWAPBOX>`: www.articlemanager.com/download.cfm
- A good guide to relational database design: Hernandez, M.J. (1997). *Database Design for Mere Mortals*. Addison-Wesley.

About the Author

Tom Peer has been in electronic publishing of one sort or another for eight years, including a stint as manager of New Scientist Online (www.newscientist.com). He specializes in taking printed publications online and has recently completed the online edition of The World Handbook of Stock Exchanges (www.exchange-handbook.com).

[@TOM@TOMPEERCONSULTING.COM](mailto:TOM@TOMPEERCONSULTING.COM)

MACROMEDIA

www.macromedia.com/go/mastering

A More Thorough Debugging

BY
ERIC
BRANCACCIO



Some often overlooked techniques to improve your applications

When an application is functioning exactly as expected, does that mean it's bug-free? Not necessarily.

ColdFusion provides developers with many useful and powerful debugging techniques. If I find that part of an application isn't functioning the way I expect, I can usually use one or more of them to target the problem.

I consider debugging to mean much more than targeting obvious malfunctions in an application. It also involves taking into account performance and security, and making sure that execution time of all pages is optimal and that no part of the application is vulnerable to malicious attacks.

Obviously, you could write an entire book on performance and security checks, but I hope to share some of the ones I feel are most important and most often overlooked. An application could seem to be functioning fine, but it could still be inefficient and unsafe once it's opened up to the outside world. I go through several distinct procedures to make sure my code is bug-free.

First I configure my development environment to give me as much feedback as possible about the pages being processed. This includes enabling all debug output to be appended to each processed page. From the security/performance standpoint this tells me a lot about execution time, and it lets me see all form, URL, and browser variables, which should always be dealt with carefully. This article discusses some of the practices I use, once this is in place, to keep my code fast and safe.

Parameterized Queries and Bind Variables

I believe that database interaction is the area in which most developers can improve their coding techniques. Many performance and security issues arise when commu-

nicating with a database. I develop the majority of my applications using an Oracle 8.1.6 database. The <CF-QUERYPARAM> tag provides tremendous performance and security benefits when used with this database because it enables the use of bind variables in queries.

Before ColdFusion 4.5.1 the only way to implement bind variables was to write a stored procedure. For example, if I want to insert a large block of text into my database, I'd have to write a pl/sql block or a stored procedure to convert a literal to a bind variable like this:

```
DECLARE Long_Text LONG;
BEGIN
  Long_Text :=
  '#LongTextVariable#';
```

The introduction of the <CF-QUERYPARAM> tag made the process much more manageable. It's possible to code an entire application without using a bind variable, but I've found that using them within my queries has resulted in performance increases of up to 700%. The reason for this performance increase lies in Oracle's execution path.

Let's look at the following query:

```
<CFQUERY DATASOURCE = "dsn">
SELECT survey_name
FROM survey
WHERE survey_id = #session.survey_id#
</CFQUERY>
```

If the survey table contains 50,000 records, depending on the session variable, the SQL statement generated by ColdFusion Server will be something like this:

```
SELECT survey_name FROM survey
WHERE survey_id = 34791
```

When the Oracle database receives this statement, it must decide on an execution path for retrieving the data. The path it finds is the optimal way for it to retrieve the data from the table. For example, it would be more optimal to do an index read than a full table scan.

The execution path that Oracle generates can be represented as a tree structure in which each node represents an operation. This path can be analyzed by using Oracle's EXPLAIN PLAN command. To minimize the time it takes to find this optimal execution path, Oracle keeps a cache of queries and their optimal execution paths. Each value for the session variable above will result in a different cache entry, however, so not much time will be saved.

The key to taking advantage of Oracle's caching of queries and their execution paths is use of the <CF-QUERYPARAM> tag, which will pass the session variable to the database as a bind variable. Optimizing the query above, it becomes:

```
<CFQUERY DATASOURCE = "dsn">
SELECT survey_name
FROM survey
WHERE survey_id = <CFQUERYPARAM
VALUE = "#session.survey_id#"
CFSQLTYPE = "CF_SQL_NUMERIC"
MAXLENGTH = "5">
</CFQUERY>
```

The query that's sent to the database is now:

```
SELECT survey_name FROM survey
WHERE survey_id =: 1
```

The ": 1" that's substituted in the query is a parameter reference, so now there will be only one cache entry and corresponding execution plan for every possible value of the

session variable. When dealing with large tables, the SQL cache will fill up very fast if bind variables aren't used. Performance will be lost since the CPU will be busy parsing so many different execution plans.

Using the <CFQUERYPARAM> tag also takes care of handling potentially unsafe data, primarily FORM, URL, COOKIE, and browser variables, which can be especially threatening when we use them to interact with our database. Many applications include queries that contain FORM or URL variables. I once came across an application that would log in users with a query like:

```
<CFQUERY DATASOURCE = "dsn"
NAME = "checkusers">
SELECT *
FROM users
WHERE user_name =
'#FORM.user_name#'
AND password = '#FORM.password#'
</CFQUERY>
<CFIF CHECKUSERS.RECORDCOUNT IS 1>
<!-- Login is valid -->
</CFIF>
```

Clearly this query would be very vulnerable to a malicious attack and is a very insecure login method. If someone were to modify the contents of the FORM.password variable to something like:

```
"(SELECT password FROM users
WHERE user_name = 'jdoe')"
```

they could log in as any user whose username they happen to know. A simple solution to this problem would be to use a parameterized query:

```
<CFQUERY DATASOURCE = "dsn">
SELECT user_name
FROM users
WHERE user_name = <CFQUERYPARAM
VALUE = " #FORM.user_name#"
CFSQLTYPE = "CF_SQL_VARCHAR"
MAXLENGTH = "10">
AND password = <CFQUERYPARAM
VALUE = " #FORM.password#"
CFSQLTYPE = "CF_SQL_VARCHAR"
MAXLENGTH = "10">
</CFQUERY>
```

Now if a string longer than length 10 is passed in, an error will be thrown and the query won't be executed.

There are situations in which parameterized queries aren't the right choice. In the previous example the session variable will change often depending on which of the 50,000 surveys is being accessed. Let's consider a situation in which we are writing a query based on a variable that can have only a few values:

```
<CFQUERY DATASOURCE = "dsn"
CACHEDWITHIN = "#CREATETIMES
PAN(0,1,0,0)#">
SELECT state_name
FROM states
WHERE region_id = #region_id#
</CFQUERY>
```

In this situation it's most efficient to cache the query.

There are a few final things I want to touch on concerning database interaction. Make sure to pay close attention to the query execution time in the debug output. Sometimes simply reorganizing the order of the SQL clauses can give a performance gain, so all possible structuring of a SQL statement should be tested. Also, when you expect a large amount of rows to be returned, make sure to specify a block-factor. This will reduce unnecessary trips to the database and can cut down execution time substantially.

GetTickCount()

The GetTickCount() function is a valuable tool for analyzing code performance. ColdFusion Server allows a developer to get the execution time of entire templates, but often we need to get the execution time of specific pieces of code within a template. I use the GetTickCount() function along with the IsDebugMode() function throughout my applications so I can get feedback on every bit of processing that ColdFusion Server will perform. I surround blocks of code that I'm analyzing like this:

```
<CFIF IsDebugMode( ) >
<CFSET GetTickCount( ) = t1>
</CFIF>
<!-- Code Block Here -->
<CFIF IsDebugMode( ) >
<CFSET TotalTime =
GetTickCount( ) - t1>
<CFOUTPUT>#TotalTime#</CFOUTPUT>
</CFIF>
```

SUBSCRIBE AND SAVE

WebServices JOURNAL

Offer subject to change without notice

ANNUAL NEWSSTAND RATE	
\$93.88	
YOU PAY	
\$69.99	
YOU SAVE	
\$13.89	Off the Newsstand Rate

DON'T MISS AN ISSUE!

Receive 12 issues of **Web Services Journal** for only **\$69.99!** That's a savings of **\$13.89** off the annual newsstand rate.

Sign up online at www.sys-con.com or call 1 800 513-7111 and subscribe today!

In April **WSJ**:

Using Web Services with J2EE
Moving into a technology-independent world

Asynchronous Web Services
Deployment based on JMS that extends SOAP-over-HTTP

Dynamically Converting Existing Java Code to a Web Service
A robust and comprehensive framework for delivering Web services

Web Services Over P2P Networks
The technology is ready for an interesting intersection

Knowing the Score - Web Services and Business Processes
The promise of a new era in e-commerce



The `IsDebug()` function is used to test whether debugging has been enabled. If this function returns false, the variable "TotalTime" won't be displayed. This practice is very useful, especially on pages when a lot of server-side processing is being done. It tells a developer the most efficient coding techniques to use to accomplish certain tasks. It's a good idea to write a custom tag to time code blocks so you can analyze blocks of code like this (see Listings 1 and 2):

```
<CF_TIMETEST>
<!-- Code Block Here -->
</CF_TIMETEST>
```

Server-Side Validation

If security is an issue, all data validation should be performed on the server side. Client-side validation is functionally fine, but it can easily be bypassed. Here are some server-side validation techniques I use to make sure the data I use is safe. First I make sure every request is coming from my server:

```
<cfif not Find( "myDomainName",
CGI.HTTP_REFERER)>
<!-- This has not come from
my server -->
```

This check alone is still not good enough because the `http_referrer` is a client variable and can be modified. A developer must look at every variable being handled and ask: "How could this variable have unwanted effects on my application?" It's important to check for the existence of all variables using the `IsDefined()` function. Then I use the `<CFPARAM>` tag, not to assign a default value if one doesn't exist, but to validate the data type:

```
<CFPARAM NAME = "user_id" TYPE
= "numeric">
```

Not only does specifying a data type speed up processing, but also, if the `user_id` variable isn't numeric, ColdFusion will throw an error.

I believe it's important to become familiar with Regular Expressions. Often when I handle string data I need to put some complex validation rules in place, and the use of Regular Expressions facilitates this. For exam-



ple, any variable that will be outputted on a page should be checked to make sure it doesn't contain any HTML tags other than `` or `<I>`. If a `</td>` or a `</div>` tag is outputted on a page, it can ruin the display of the page. I have come across so many message board applications in which a user can post some text with a hyperlink that starts an infinite loop of `window.open()` calls and crashes the user's browser. A simple fix for problems like these, using Regular Expressions, is:

```
<CFSET TextToBeDisplayed =
REReplaceNoCase(FORM.TextToBeDis
played,"(<[/]?[^\BI]>)|(<[/]?([^\>
]+)>)|(<[/]?([^\>]{2,})>)", "", "ALL")>
```

For numeric data I always check it first with the `Val()` function. This returns either a number or zero (if the string passed in can't be converted to a number). This is useful because some values that ColdFusion considers to be numeric could have undesired results if handed off to a database. The `Val()` function converts these values to a number:

```
<CFSET MyNum = "5d3">
<CFPARAM NAME = "MyNum" TYPE =
"numeric">
<CFSET MyNum = Val(MyNum)>
<CFOUTPUT> #MyNum# </CFOUTPUT>
<!-- 500 will be printed -->
```

I then make sure the length and range of the number are consistent with what I'm expecting.

Clearly, data validation is something that must be done; however, if it's performed on the client side, it's basically worthless protection against people who are purposely trying to disrupt your site.

Encrypting and Encoding

I believe in encrypting all URL and FORM and form variables. This makes it much more difficult for someone to tamper with an application. For example:

```
<input type = "hidden" name =
"user_id" value = "25"> or
http://MyDomain/viewdocument.cfm
?doc_id=2352
```

These would give a user much more information than:

```
<input type = "hidden" name =
"user_id" value =
"04J039M9384N"> or
http://MyDomain/viewdocument.cfm
?doc_id=098HT9BNF893
```

ColdFusion provides the `Encrypt()` and `Decrypt()` functions, but I recommend developing your own tag, which should include some sort of check to make sure the data hasn't been changed between the encryption and decryption process.

Not only should URL variables be encrypted, but they should be encoded as well, using the `URL-EncodedFormat()`. This will ensure all characters that could cause problems will be escaped.

A stronger form of encryption can be achieved through the `Hash()` function. This can be used only to compare values since there's no way to decrypt the string. I use this occasionally as a way to check whether certain values have been changed.

• • •

A thorough debugging is a vital part of the development process. In many instances an application that my team developed ran perfectly in the test environment. Once we opened it up to the world, however, various problems would arise – primarily database performance problems due to the CPU on our Oracle box parsing so many execution plans. Even the load testing we did while testing failed to reveal this problem.

To improve our security I began logging all queries to better understand the attacks on our systems. This taught me all sorts of new methods for protecting my applications. Choosing the code-checking techniques I thought were most important and at the same time most infrequently used by most developers was a difficult task. Hopefully these techniques will improve the overall quality of your applications.



@E_BRANCACCIO@ACS.ORG

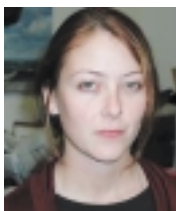
ABOUT THE AUTHOR

Eric Brancaccio, president of EMB Consulting and a certified ColdFusion developer, uses ColdFusion to provide powerful, low-cost solutions primarily to small and nonprofit institutions. He is also a senior systems engineer for the American Chemical Society in Washington, DC.


```
<!-- Example showing use of custom tag to test performance
-->
```


Session States Without the Session

BY
TSARA
BORSTING



Why on earth would you want to try such a thing?

Discussions have centered around clusters and shared sessions: How do you overcome the problem of being unable to share session variables across clustered servers? Why else would you build a sessionless site?

The answers may surprise you. This article documents a technique that allows session states on clustered servers, increases user security, and provides greater development flexibility. However, like all ColdFusion applications, this is only one way to create a “sessionless session,” and it can be modified infinitely.

The first answer I came up with when presented with this problem was: “Simple. Use client variables.” But how do you create client variables to house data that should expire after a certain time? Also, do users want your session information stored on their hard drive? I don’t think so. Besides, it’s a huge security risk to store a user’s login and session on a computer on the other side of the planet.

The answer? All the data you need is already being set on their machine: CFID and CFTOKEN. For those of you who don’t know, whenever session management is initiated, ColdFusion automatically sets the CFID and CFTOKEN cookies on users’ computers to track their sessions. This solution does the same thing, just without the session...

Pro’s and Con’s

The advantages to the CFID/CFTOKEN approach are endless. By using the CFID and CFTOKEN cookies to reference a user’s data, I can store vast amounts of information about a user’s session state in an easily maintainable datasource on my server (and no one is the wiser). Here’s where CFLOCK also comes into play: you can lock all of your client and session state variables in one area without going to every page that uses them. This reduces the amount of CFLOCK-ing in gen-

eral (a time-saver) and the odds of variables being overwritten by other users (a hair-saver).

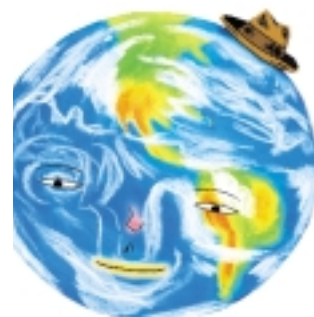
Naturally, there are disadvantages to this approach as well. What if your user has cookies turned off? You may as well ask what if your site is red and green and your user is color-blind. You can’t control this and have no way of knowing about it. However, you can avoid this potential problem by turning the ClientManagement attribute of your CFApplication tag to “no” and meticulously appending the CFID and CFTOKEN variables to every URL, form action, and hyperlink.

The ColdFusion documentation agrees that this is unnecessary: “In ColdFusion, client state management is explicitly designed to work with cookies, the standard tool for identifying clients. Using client state management without cookies requires careful programming to ensure that the URLToken is always passed between applications.”

This technique also requires you to perform a small amount of footwork. You must call the appropriate modules each time you want to use a “session” variable. The advantages greatly outweigh this overhead aspect and are easily streamlined. Read on to see how.

Data Storage

Now that you’re using the CFID/CFTOKEN approach, the next question is: How will you keep track of all this data? Technically, you accomplish this by setting client variables, but these variables aren’t stored on the client’s machine. By setting the “ClientStorage” attribute of the CFAPPLICATION tag to your specified datasource, you can collect all client data in two tables set up by Cold-



Fusion. When you specify a datasource, ColdFusion automatically stores all standard client variables (LastVisit, HitCount, TimeCreated) in the CGlobal table and all user-specified client variables in the CData table.

So how do you store the data? Didn’t I just answer that? Yes and no. The data is stored in the CData table. However, you have the option of:

- Storing each variable as its own row, indexed off the user’s CFID and CFTOKEN
- Wrapping it all into a bundle and storing the data as a BLOB in one row

This is a matter of choice; both are perfectly viable options. I prefer the latter approach for two reasons. First, there’s a smaller chance of “losing” users’ data by having multiple rows associated with their ID. Second, you may never know all of what’s being stored in the client state or how it’s named. By storing your information in an often forgotten ColdFusion feature – a structure

– you can read and write all of the data without knowing any of the variable names (see Listing 1). To read structure values without knowing the key names, use the StructKeyList function and the CFLOOP tag, as follows:

```
<cfloop List =
"#StructKeyList(SessionInfo)#"
index="ThisValue">
    <cfset "#ThisValue#" =
Evaluate("StructFind(SessionInfo
,'#ThisValue#')")>
    #ThisValue#<BR>
</cfloop>
```

Unfortunately, ColdFusion doesn't allow you to set a structure as a client variable. To overcome this obstacle, I used another often overlooked aspect of ColdFusion: WDDX. Simply serialize the structure into a WDDX packet and set the packet as a client variable. When you need to call out your client variables, deserialize the packet and set its name-value pairs as local variables for the calling page:

```
<cfif
IsDefined("Client.SessionInfo")>
    <cfwddx action="WDDX2CFML"
input="#Client.SessionInfo#"
output="SessionInfo">
</cfif>
```

I created this feature as a custom tag that's called on any page that sets or reads client variables. If you have a modularized site, you can easily integrate this into your "header" module. With this method you can call any of the stored client variables as a local variable (instead of client dot variable name). If you aren't using modules for your site (you should be!), call the tag each time you want to use the "session" variables. *Note:* Be sure to use the appropriate attributes and caller scopes if you do implement this as a module.

The Timeout Issue

Finally I tackled the timeout issue: how to automatically log out of a user's session after a given period of inactivity. I built this feature into the same module that deserializes the user's stored data. I simply compared the users' "LastVisit" time with the time at which they logged in. I set the difference to be compared between the login and LastVisit times as an application variable for more flexibility. Since the LastVisit variable is updated with every page visit, the time is accurate. If the LastVisit time is greater than the set logout period, the client WDDX packet is deleted from the datasource and the user is logged out.

Obviously, this isn't the only way to log out inactive users, just the technique I prefer. Other ideas include a constant time check in the header module or Application.cfm. I chose my method because my goal was to decrease the amount of footwork for the other developers on my team and to combine as many attributes into one module call (see Listing 2).

Is This Right for You?

Everyone's site is different. Some sites may not need such an elegant session module when one is already built into ColdFusion. If you plan to scale your site to a clustered environment in the future and don't want to rewrite every page, starting in this style is a safe choice. This technique is ultimately geared toward sites on clustered servers, but can just as easily be used on a single server.

This method of coding will help boost your site's security and give you greater control over your users' session information. Just think, no more "shared sessions," CFLOCK mayhem, or lost sessions due to clustered environments.



@ TSARAB@YAHOO.COM

ABOUT THE AUTHOR

Tsara Borsting, a Macromedia-certified ColdFusion 5.0 developer, has been programming for five years in Santa Cruz, CA.

Listing 1

```
<!--- Build the "session" structure --->
<cfset SessionInfo = StructNew()>

<!--- set a default number of variables or pass it in --->
<cfloop From="1" to="#NumberOfVariables#" index="i">
<!--- set the name value pairs as local variables for ease
of use --->
    <cfset q_KeyName      =
Evaluate("attributes.q_SessionName#i#")>
    <cfset q_KeyValue     =
Evaluate("attributes.q_SessionValue#i#")>

<!--- insert the name value pairs into the structure --->
    <cfset temp =
StructInsert(SessionInfo,"#q_KeyName#", "#q_KeyValue#")>
</cfloop>
```

Listing 2

```
<!--- check the difference between the user's login time
and their last visit --->
```

```
<cfif DateDiff("n", "#LoginTime#", "#LastVisit#") GT
"#Application.SessionTimeout#">
<!--- if the user has been inactive for long enough, delete
their structure --->

    <cfif IsDefined("Client.SessionInfo")>
        <cfscript>
            DeleteClientVariable("SessionInfo");
        </cfscript>
    </cfif>

<!--- go to the login page if desired --->
    <cflocation url="login.cfm">
    <cfabort>
</cfif>
```

CODE LISTING



The code listing for this article is also located at

www.sys-con.com/coldfusion/sourcec.cfm

Toward Better CF Server Administration

As an administrator, or a developer interested in how your server is run, are you getting the most out of your CF Server configuration, especially with regard to security-related settings in the Administrator? Are you fully aware of the opportunities and challenges presented by those settings?

In this second article of a series (Part 1 appeared in the January **CFDJ**, Vol. 4, issue 1), we continue the discussion of some perhaps less obvious aspects of managing CF Server. Part 1 focused on developer- and performance-oriented settings and tasks; here we'll look at security-related settings and tasks. The original plan was to cover some miscellaneous features as well, but it turns out there's more to both topics than the original two-part format could hold. One more part is planned.

As I said in Part 1, there's no way to cover everything in a single article (or even three). If you take the time to read the available documentation outlined in Part 1, you'll benefit tremendously. And the available online help in CF5's Administrator is another great improvement.

Still, for those who don't do the reading or may have a hard time determining what to pay attention to, I'll highlight a couple of aspects that you might otherwise miss or find confusing. There's more to CF's security features than you might be aware of. Although each discussion is brief, I hope to motivate you to look into these topics on your own.



Basic and Advanced: Insider Tips

CF offers two forms of security out of the box: basic and advanced. If you fire up the CF Administrator (as discussed in Part 1), you'll find them under *Security* at the top left of the Administrator, as seen in Figure 1.

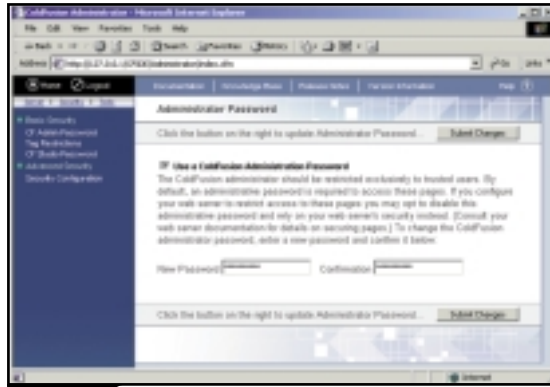


FIGURE 1: CF Administrator basic security settings for Admin password

Note that if “Advanced Security” hasn’t been installed (it’s optional), it won’t show up in that navigational toolbar. Prior to release 5, the Basic Security option showed up left of the main Administrator toolbar under the Server heading (again, Advanced Security shows up under Basic if installed).

The simple designation of “basic” and “advanced” security really doesn’t do justice to the differences between the two. There are three forms of control that these two approaches share: access to the Administrator, access to tags within the CF environment, and access to resources by way of ColdFusion Studio’s RDS feature (supporting remote developers connecting to the server over the Internet).

There are other aspects of security that the Basic approach doesn’t provide, and many developers don’t realize that ColdFusion can help them with Advanced Security’s built-in functionality. First, there’s the simple matter of enabling security within your application, providing a login process to control which users are allowed access to the application and/or specific pages within the app. This is referred to as *authentication*.

Most people “roll their own” form of it, looking up users in a database and using session variables to track a user’s logged-in state. This works, but Advanced Security provides a mechanism to handle these tasks in a more standard way. It allows for authentication against not only a database but also, optionally, an LDAP server or a Windows NT domain.

Identifying who may access your site is just one part of the application security equation: once users are logged in, you may want to limit what they’re allowed to do. This is referred to as *authorization*. Not only can Advanced Security provide for multiple levels of security, with roles and varying rules per individual (or group), but it can even limit access to what specific CF tags, functions, data sources, collections, files, and other resources a user may access. We’ll come back to these aspects of application security later.

This article can’t discuss all these security matters in depth, but I do want to put some things in context.

CF’s Basic Security Option

Using “basic” security, which is implemented by default, the only things that CF secures are:

- Who can access the CF Administrator itself (by way of a password defined at installation or changed within the Administrator)
- Who can access the CF Server resources (files and data sources) by way of Studio’s RDS feature (also controlled by a single, but optionally different, password that is set at installation and can be changed in the Administrator)
- Whether developers can execute any of several potentially dangerous tags

These choices are found under the Basic Security link in Figure 1. Even if you’re familiar with these three settings, you may not have considered some of the aspects.

Understanding and Guarding Access to the Administrator

The CF Administrator is a Web-based application, which means that it’s accessible over the Web if you know the URL for it, which defaults to <http://<yourserver>/CFIDE/Administrator/index.cfm>.

Maximize the administration of your CF Server

Part 2 of 3

That remote accessibility is a benefit for many, but there's also the risk that someone gaining unauthorized access might compromise it. Obviously, those who know the Administrator password will be able to change any aspect of the CF Administrator settings, so the password should be guarded. Sadly, there are no mechanisms forcing you to change it regularly, so it would be a good idea to do that on your own.

You can take extra steps to secure the Administrator Web pages at <webroot>\CFIDE\Administrator\ by using Web server security to tighten control further. Just be careful that you don't lock the entire CFIDE directory one level up, because there are more things in there than just the Administrator. Such measures are discussed at www.macromedia.com/v1/Handlers/index.cfm?ID=10954&Method=Full.

Another idea is to remove the CFIDE/Administrator/ directory if you're not using the Administrator at all (which is unlikely, but possible), or simply to move it so that visitors to your server can't easily guess its location. Just be aware that there is a setting in the Administrator – the Unsecured Tags Directory – which by default points to that Administrator directory. If you move the directory, be sure to change that as well or the Administrator won't function. The Unsecured Tags Directory was discussed in detail in my February *CFDJ* article (Vol. 4, issue 2). More on that in a moment.

Understanding and Guarding Basic Studio/RDS Security

As for the aforementioned RDS password, for those not familiar with it, it's used when setting up Studio's RDS mappings feature to allow a remote Studio user to see the files and data sources on the server, as well as perform debugging against that server. For more information on this, see the Macromedia manual "Using ColdFusion Studio," which, like all CF manuals, is available online on your server (if the docs were installed). If you own Studio, you'll find them in Studio's Help feature. Anyone can view all the CF (and Studio and other) docs online at <http://live.docs.macromedia.com>, as was described in more detail in Part 1.

One problem with the RDS password feature in basic security is that it's an all-or-nothing proposition. Anyone who knows the password and connects to your server via Studio can obtain any resources that the CF Server has access to. The next section shows you how Advanced Security offers an alternative that can provide more granular control over who can access what via RDS. Just know, too, that you can do a couple of things even with simple basic security enabled.

First, you should almost *never* uncheck the "Use a ColdFusion Studio Password" option in the Administrator, which means that no password is required at all. This is similar to the checkbox for the admin password, as shown in Figure 1 (the same admonition obviously applies to the admin password). It may seem obvious that the Use Studio Password checkbox should be checked, but some administrators who don't understand the Studio/RDS password – or misunderstand it, thinking that it might disable RDS – simply turn off the checkbox, not realizing what could happen. Any Studio user who connects to the server could now access any resources under CF's control, unless you take one of the additional steps below.

Indeed, if you know you're not going to support RDS access to your server at all (not a bad idea in a production environment), you can disable the ColdFusion RDS Service. Like the ColdFusion Application Server and ColdFusion Executive services, it's set by default at installation to start when your server is started.

Another thing you could do, as with the CF Administrator itself, is apply additional Web server security to the RDS connection. When a Studio RDS session is connected, Studio actually runs something like a Web service request to the server (using

WDDX under the covers), interacting with a program in the CFIDE\Main\ called *ide.cfm* (ever wonder where the CFIDE directory got its name?). You could apply additional Web server security to that program or directory as another level of protection. Just be sure to test things out within Studio, since some changes you make may not allow Studio RDS connections to work at all. Again, we're discussing authentication here – who's allowed to use the RDS feature – rather than authorization – what an authenticated user is allowed to access within the server once connected via RDS. That kind of security, enabled by Advanced Security, will be discussed later.

A final way to limit what RDS users can see when they connect is to have the ColdFusion Server service start under an account other than the default, System. There are many possible ramifications of that choice, so I recommend you research it carefully before considering it.

Understanding and Fully Leveraging Basic Tag Restriction Security

The third and final aspect of basic security is optional control to restrict access to a set of potentially abused CF tags. If you follow the link for Tag Restrictions, as listed in Figure 1, you'll be presented with a screen like that in Figure 2. If any of the tags listed here isn't checked, no developer on the server can execute that tag. It's a brute force approach, but in the name of security it works.

As with RDS security, this aspect of Basic Security is an all-or-nothing proposition. If a tag is restricted, the intention is that no one on the server can use it at all. A developer may be unable to perform some needed action, like a CFFILE file upload or a CFCONTENT download. For many that's too severe, and they may choose instead to leave the tags unrestricted, which exposes any applications on the server to potential abuse by less scrupulous developers on that same server. There are two alternatives to consider.

First, in my February article, "Unlocking Restricted Use of CFFILE, CFCONTENT, and More," I wrote about using the Unsecured Tags Directory option, which is also offered at the bottom of Figure 2. With that you can still choose to restrict the tags, but any code placed in the named directory can indeed execute any of the otherwise restricted tags (whether by CFINCLUDE or a template there or a call to it as a custom tag, as with CFMODULE). That's a cool alternative, if you didn't know about it.

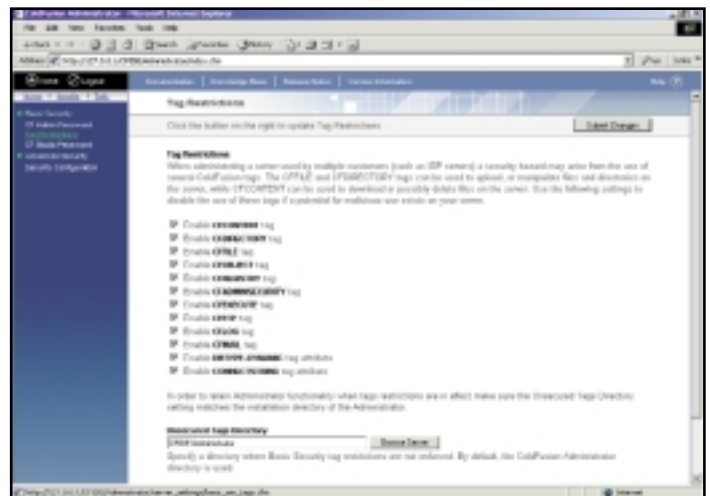
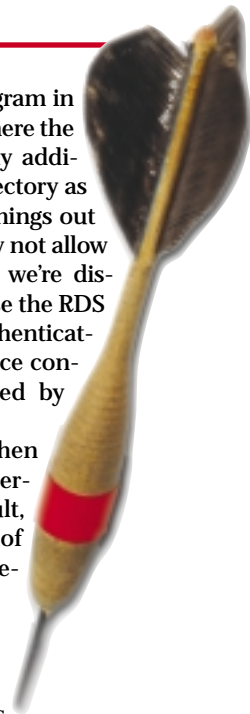


FIGURE 2: CF Administrator Basic Security settings for tag restrictions



**COLDFUSION
DEVELOPER'S CONFERENCE
www.cfnorth.com**

But there's another solution, which on the surface would seem absolutely ideal for this challenge – and many others – and it's the subject of the remainder of this article: Advanced Security, the second part of CF's built-in access-control mechanisms.

CF's Advanced Security Option

The Basic Security mechanism is nearly as old as CF itself, but in release 4 the CF team recognized that a finer level of control was needed beyond the previously available all-or-nothing aspects of securing the Administrator, RDS access, and the tag restrictions. The Advanced Security mechanisms were created to give finer-grained control over those things and much more.

Advanced Security (in CF releases 4 and 5) is enabled by the (transparent) bundling of a third-party product called SiteMinder from Netegrity. It's an option available at installation time and can be added after installation. The only changes it makes to CF are the addition of new mechanisms in the Administrator for controlling the setup of Advanced Security and a couple of tags and functions for the application-level security that it can enable.

Briefly, some of the things Advanced security can enable and/or control are:

- *Authentication* against an ODBC database, LDAP server, or Windows NT security domain
- *Granular control* of RDS access to resources on the server (who can access what)
- *Login/authentication functionality* for your applications (through the CFAUTHENTICATE and CFIMPERSONATE tags and the IsAuthenticated, AuthenticatedUser, and AuthenticatedContext functions)
- *Granular control* of what resources a user may access within an application (tags, custom tags, functions, directories, files, databases, verity collections, etc.) and *authorization functionality* to test for such access in your programs through the IsAuthorized function
- *Permission* for specified developers to access the Administrator to manage their own data sources (a feature that was removed in ColdFusion 5. See KB 21965)
- *Sandbox security*, which may be most useful in a hosting environment, as it restricts at runtime the available directories and files that the users of a given application can access (not related at all to RDS security, and different from the authorization abilities described above)

We don't have room for more detail, but there is ample coverage of it in both the "Advanced ColdFusion Administration" manual (renamed and improved in release 5) and the "Application Security" chapter of the ColdFusion user's guide, named "Developing ColdFusion Applications" as of release 5. Again, if you don't have the printed manuals, these docs are available online at <http://livedocs.macromedia.com> (and are available for purchase as a complete set as well).

It may be worth noting that the ability to authenticate against a database was introduced in version 4.01 of ColdFusion. Initially, many administrators or developers may have dismissed using the authentication features because they didn't want to deal with the complexity of an NT domain or LDAP server, but they may not have heard about the fix in 4.01 that solved that problem. (Indeed, there were many small but important changes in 4.01, and now, two years later, I'd still recommend my February 2000 article, "Hidden Gems in 4.0.1" (Vol. 2, issue 2). There's so much that people missed and still don't know, or worse, spread as misinformation.)

Advanced Security Setup

Briefly, the key is the initial process of creating:

- One or more user directories indicating what database, LDAP server, or NT domain will be used for authentication for an application or for RDS access (you can have different user directories for different purposes)
- One or more resources describing the actual resource (file, directory, tag, collection, etc.) that you want to control and the type of access to it that should be allowed
- One or more policies mapping what users should be allowed access to what resources
- One or more contexts serving as a logical grouping of related user directory, resource, and policy specifications

A better discussion of the details can be found in the Advanced Administration book, specifically the section "Advanced Security Basics," at http://livedocs.macromedia.com/cf50docs/Advanced_ColdFusion_Administration/AdvSecurity3.jsp.

While you're debating whether to use Advanced Security, remember that, even if installed, it doesn't take effect unless the administrator chooses to set it up and enable it. That's done by way of a checkbox called "Use Advanced Security" at the top of the Advanced Security page. Furthermore, simply checking the box won't have any impact unless and until you configure it further.



There's talk of substantial changes to Advanced Security coming in the next release of ColdFusion"

Another thing to keep in mind is that it can be enabled to secure one aspect of the server (for instance, controlling RDS access) while not impacting existing applications at all. In fact, Advanced Security won't affect existing applications unless they're modified to leverage the new features (tags and functions) that it enables. So there's no harm in trying it out. As usual, it's best to experiment with any new approach in a development or testing environment before implementing it on a production server.

It's worth mentioning here that a free, single-user developer edition of ColdFusion is available for such testing. You no longer have to get it from CF Studio alone. Just download the fully functioning, non-user-limited trial version of CF that, after 30 days, will revert to a single-user version. It will remain fully functioning in every other respect. Both it and the downloadable trial version are Enterprise versions of ColdFusion. Some aspects of Advanced Security, such as Sandbox Security, are only in the Enterprise version.

Challenges in Implementation

Advanced Security offers quite a few useful improvements. Unfortunately, the SiteMinder integration underlying it has proved challenging for several reasons, and most developers have thus never benefited from its existence. There are also other reasons why it hasn't been that popular.

It's an option at installation, so many don't even install it, not knowing any better. Furthermore, since it requires setup, many admins have never bothered to take that next step, or they may fear that enabling it without the complete setup will harm their environment (it won't).

THE LARGEST INTERNATIONAL

XML

CONFERENCE & EXPO IN THE WORLD!

**WIN A
\$35,000
LUXURY CAR!**



ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING
TO WIN AND RIDE OFF IN A \$35,000 LUXURY CAR!

XML-NEXT G OF ENTERPRISE DEPLOYMENT

REGISTER ONLINE TODAY

**FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!**

VISIT WWW.SYS-CON.COM

Focus on XML

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

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

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



A Sampling of XML-Focused Sessions

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



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

Featuring...

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

Who Should Attend...

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

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



JEAN-FRANCOIS ABRAMATIC
SENIOR VP R&D, ILOG
FORMER CHAIRMAN, W3C



TYLER JEWELL
PRINCIPAL TECH
EVANGELIST • BEA



DAVID LITWAK
CEO
SILVERSTREAM



ANNE THOMAS MANES
CFO
SYSTEMS



BARRY MORRIS
CEO
IONA



RICK ROSS
FOUNDER
JAVALOBBY



PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

For Exhibit Information

CONTACT: RICHARD ANDERSON
115 CHESTNUT RIDGE RD.
MONTVILLE, NJ 08055
201 842-3054
RAND@SYS-CON.COM

XMLEDGE
conference & expo

JUNE 24-27

JACOB JAVITS
CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3

SAN JOSE
CONVENTION CENTER
SAN JOSE, CA

SPONSORED BY:

IONA | END 2 ANYWHERE™

ADOS

bea

TogetherSoft

SilverStream

Actional

MERANT

Altaworks

PolarLake
Enterprise Strength XML

ALTOVA sm3 app

MEDIA SPONSORS

Federal Computer Week

WebServices.org

XML TIMES.com

Java Skyline

CE Advisor

WebServicesMall

WIRE

XML.org

WROX

XML.org

SDTimes

JAVA

wireless

XML

WebLogic

WebServices

WebSphere

COLDFUSION

OWNED AND PRODUCED BY

SYS-CON MEDIA SYS-CON EVENTS

Since the benefits aren't obvious to admins and developers unless they read the manuals, many developers have never learned of it nor have they demanded its support by their administrators.

There are some more substantial problems, however – even for those who have tried to use it. For example, the Administrator interface for setting it up can be quite difficult to understand and use. It's been improved in both 4.5 and 5, but it's still daunting. Also, since there are so many ramifications to what it enables, it's easy to misconfigure.

In addition, the default installation uses an Access database as the underlying SiteMinder repository, which of course does not perform well. Even in a low-volume site, the highly interactive nature of the Advanced Security authentication capabilities is such that it can bog (or break) down rather quickly. The simple solution is to use an alternate database, and there are knowledge-base articles and documentation (quite improved in CF5) about doing just that, but the bottom line is that many have just not bothered to install or use it at all.

All this is too bad, because there are quite a few positives about Advanced Security. While there's talk of substantial changes coming in the next release of ColdFusion, code-named Neo, at the time of this writing (early March) it's still in beta. It will be interesting to see if the response of the Neo team is toward making what Advanced Security tried to offer more effective, or toward removing features that people never bothered to use. That would be unfortunate, because it's not that the features it enabled weren't worthwhile, but that the initial implementation, marketing, and education appeared lackluster, and most influential users who tried it got a bad taste.

The docs are better, and there are several KB articles on Advanced Security. Just visit the KB Search form at www.macromedia.com/v1/support/knowledgebase/searchform.cfm and search for "Advanced Security." You might want to try it. We can look forward to what awaits us in the next release, but at least now you know what's possible in the meantime.

Other Aspects of Security in the Administrator

While I've focused on the distinctions between Basic and Advanced Security in ColdFusion, there are still a few other aspects of security worth mentioning (and worth your exploring).

Perhaps the simplest to appreciate is security of your data sources by the use of

database usernames and passwords that will be controlled by the database engine. If you use such security, you can choose to store the userid and password in the Administrator's data source definition itself, or you can specify it on the CFQUERY (or any other database-related tag) itself.

An Aside for Users of Microsoft Access

Many of you never bother to secure your database at all, but it's certainly worth doing. One simple way is to create a password for the entire database using Tools>Security>Set Database Password within the Access interface, and then specify that in the password field for "Default Login" (the username defaults to "Admin"), or you can specify that username and the newly set password within a CFQUERY itself. That way, if someone steals the database, it's not that easy to open. And if you want to create individual user logins for the database, look into the Workgroup Information File feature in Access. With that sort of security in place, you can specify that file as the System Database when setting up the data source configuration in the CF Administrator and then use usernames and passwords within your application, as with any other database.

Many don't notice that as another measure of security you can set which SQL statements are or are not allowed via a set of checkboxes in the CF Settings for a data source. (Internal database security can limit that as well, of course, but this is just another possible level of control.) You can even have multiple data sources for the same database with different controls (of SQL statement access or even username/password combinations, if that makes sense for you).

Moving from databases, still another source of security is control over resources on the server and associated servers by way of the Web server and/or operating system, which might be used instead of or in addition to CF's Advanced Security.

Yet another aspect of security involves the ColdFusion Administrator Debugging settings. These are discussed at length in

the CF Manual "Installing and Configuring ColdFusion Server" in a section available online at http://livedocs.macromedia.com/cf50docs/Installing_and_Configuring_ColdFusion_Server/basicconfig13.jsp. You should investigate whether to enable debugging at all, as well as which users should see the debugging output and what potentially sensitive information should be shown or restricted, such as whether to show SQL and data source information and whether to show the full template path of a template in error. I discuss those latter two aspects in my October 2000 article "Toward Better Error Handling (CFDJ, Vol. 2, issue 10).

Another aspect of Administration security that may be worth noting is a change as of release 5. CF5 Enterprise now tracks all the configuration changes you've made to the ColdFusion Administrator, providing options for both a setting summary (a snapshot in time) and change log (an audit trail).

In these days of increasing instances of hacking and occurrences of worms and viruses, administering the server also requires paying attention to security, not only of the ColdFusion Server itself, but also the Web server and physical server environment. To that end, there is a security zone at Macromedia (www.macromedia.com/v1/developer/SecurityZone/) with information that should be understood by anyone setting up a ColdFusion Server (it covers issues related to some common Web server and database security problems as well). It includes more Macromedia products than just ColdFusion, of course, but there are many CF-specific sources there, such as security bulletins, best practices, and white papers. There's even a notification service to receive security bulletins by e-mail as soon as they're released.

• • •

I hope this quick tour of features, possibilities, tips, tricks, and traps will make you at least a little more familiar with what's available to make your server more secure and have it running more effectively – not just for administrators, but for developers and end users as well.



About the Author

Charlie Arehart is a certified Macromedia trainer/developer and CTO of SysteManage. He contributes to several CF resources, is a frequent speaker at user groups throughout the country, and provides training, coaching, and consultation services. Charlie is also a columnist for *Java Developer's Journal*.

CAREHART@SYSTEMANAGE.COM

THE LARGEST INTERNATIONAL

WEB SERVICES CONFERENCE & EXPO IN THE WORLD!

**WIN A
\$35,000
LUXURY CAR!**



ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A \$35,000 LUXURY CAR!

WEB SERVICES SKILLS, STRATEGY, AND VISION

REGISTER ONLINE TODAY

**FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!**

VISIT WWW.SYS-CON.COM

Focus on Web Services

Web Services, the next generation technology that will enable the Internet to work for you and your business, and finally provide that ROI you have been after, will be put under a microscope at Web Services Edge East 2002.

Information-packed sessions, exhibits, and tutorials will examine Web Services from every angle and will provide cutting-edge solutions and a glimpse at current and future implementations. Hear from the innovators and thought leaders in Web Services. Enjoy a highly interactive CEO Keynote panel that will debate and discuss the realities and promise of Web Services.



A Sampling of Web Services-Focused Sessions

- PRACTICAL EXPERIENCES WITH WEB SERVICES AND J2EE
- STATE OF THE WEB SERVICES INDUSTRY
- THE ODD COUPLE: MAKING .NET AND J2EE WORK TOGETHER
- EXPLORING THE .NET MY SERVICES INITIATIVE
- STANDARDS WATCH
- GUARDING THE CASTLE: SECURITY & WEB SERVICES



SEAN PADDY
CONFERENCE TECH CHAIR
WEB SERVICES TRACK CHAIR
EDITOR-IN-CHIEF
WEB SERVICES JOURNAL

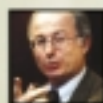
Featuring...

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

Who Should Attend...

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

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



JEAN-FRANCOIS ABRAMATIC
SENIOR VP R&D, ILOG
FORMER CHAIRMAN, W3C



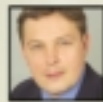
DAVE CHAPPELL
IP CHIEF TECHNOLOGY EVANGELIST
SONIC SOFTWARE



GREG KRESSLING
CEO
SITRAXA



ANNE THOMAS MANES
CEO
SYSTEMET



BARRY MORRIS
CEO
IONA



ERIC RUDDER
SENIOR VP DEVELOPER
AND PLATFORM EVANGELISM
MICROSOFT



PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

For Exhibit Information

CONTACT: MICHAEL PESICK
115 CHESTNUT RIDGE RD.
MONTVILLE, NJ 07045
201 882-3057
MICHAEL@SYS-CON.COM

web services **EDGE**
conference & expo

JUNE 24-27

JACOB JAVITS
CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3

SAN JOSE
CONVENTION CENTER
SAN JOSE, CA

SPONSORED BY:

IONA | END 2 ANYWHERE™

ADOS

bea

TogetherSoft

SilverStream

Actional

MERANT

Altaworks

PolarLake
End-to-end Strength XML

ALTOVA

MEDIA SPONSORS

Federal Computer Week

WebServices.org

XML TIMES.com

Java Skyline

CE Advisor

WebServicesMail

WIRE

XML.org

WROX

iCert

SDTimes

BASIS

wireless

JAVA

WebLogic

XML

WebSphere

WebServices

WebSphere

COLDFUSION

OWNED AND PRODUCED BY

SYS-CON MEDIA | SYS-CON EVENTS

Ask the Training Staff

A source for your CF-related questions

BY
BRUCE
VAN HORN



Many times, we look at a string or other piece of data and don't even realize that what we're looking at is actually a list. Once we realize that something can be treated as a list, CF gives us many ways of using or manipulating that data. Here are two perfect examples:

Q: *We have an e-mail newsletter that we send out to those who subscribed at our Web site. All they have to do is give us their e-mail address. I want to find out how many unique domains are represented in our subscription list. In other words, for "john.smith@somecompany.com", how can I parse out the domain ("somecompany.com") from the user ("john.smith") in an e-mail address?*

A: There are several easy ways to do this. The easiest way, perhaps, is to treat the e-mail address as a list that's delimited by the "@" character. Most people think of lists as being delimited only by commas or tabs and usually containing more than two data elements. CF, however, allows you to specify any character as the delimiter, so virtually any data string can become a list in the eyes of CF. I'll show you how to count the number of unique domains quickly, but I'll also encourage you to consider changing the way you store your e-mail address in your database to make this easier in the future.

First (see Listing 1), run a query to retrieve all your e-mail addresses. Next, create an empty list variable to hold the unique domain names. Next, loop over the query, treat each e-mail address as a list, and pull out the second element (the domain) in the list. If that domain isn't already in the list, append it to the list. Last, output the count of items (using `ListLen()`) in the domain list.

Now for my other recommendation: most of us simply create a column in our database called "E-mail" into which we stick our e-mail addresses. However, most of us also want some kind of reporting, as you've mentioned, so we know how many domains are represented in our list or, more likely, how many people there are from each domain (you'll quickly see how many of your subscribers are from AOL!). For this reason, you might want to consider breaking your one "E-mail" column into two database columns: "E-mail_Name" and "E-mail_Domain". You don't have to change anything on your Web form – let users enter their e-mail address in a single field. It's still very easy to insert into the database (see Listing 2), and it also makes it easy to pull your report using a query (see Listing 3).

Q: *My application needs to add data to another company's application (it's written in PHP). I understand how to post the data to their site (I use the CFHTTP tag to post the data), but the site gives me back return codes that are one long string with the variables separated by the "&" character and I'm not sure how to get the data I need out of the string since it often comes back in a different order for each post. Their tech support people haven't been very helpful. Any suggestions for how I can get this data?*

A: Yes. This is a common way of giving return codes back to a remote application, and getting the data can be very easy. What you want to do is break the single string down into the individual variable/value pairs that make up the string. Then you want to turn those pairs into CF variables



that you can use in the logic of your application. Your return string may look something like this: "id=28734 &status=success&message=record was added successfully&add_date=3/1/02".

Here are the steps to take (see Listing 4): first, initialize an empty structure to hold the variables/values returned by the remote application. Since you're using CFHTTP to post to the other site, their return string is coming back to CF as a variable named `CFHTTP.FileContent`. As you indicated, that string is a list delimited by the "&" character. Therefore, you need to loop over this list to find each variable/value pair. As you loop, you need to break each pair into its appropriate part (a variable name and its value). This is nothing more than another list delimited by a "=" sign. The value in front of the equal sign (position 1 of the list) is the variable name. The value in position 2 is the actual value of that variable. Set each one into its own variable and then dynamically name keys in your structure with the appropriate name/value. Now, regardless of the order in which they came back to you, you can test for the existence/value of the variables you were expecting.

• • •

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



@BRUCE@NETSITEDYNAMICS.COM

ABOUT THE AUTHOR

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


```
<!-- Retrieve all unique e-mail addresses -->
<cfquery name="getEmail" datasource="#request.dsn#">
  SELECT Email
  FROM newsletter_subscribers
</cfquery>
```

```
<!-- Create an empty list variable -->
<cfset domainlst = ">
```

```
<!-- loop over the query -->
<cfloop query="qEmail">
  <!-- get the domain name from the e-mail address -->
  <cfset domain = ListGetAt(qEmail.Email,2,"@")>
  <!-- if not in the list of domains, add it -->
  <cfif not ListFindNoCase(domainLst,domain)>
    <cfset domainLst = ListAppend(domainLst,domain)>
  </cfif>
</cfloop>
```

```
<cfoutput>
#ListLen(domainLst)#
</cfoutput>
```

Listing 2

```
<!-- Add the e-mail to the database -->
<cfquery datasource="#request.dsn#">
    INSERT INTO newsletter_subscribers (Email_Name,
    Email_Domain)
    VALUES
    ('#ListGetAt(form.Email,1,"@")#', '#ListGetAt(form.Email,2,"
    @")#')
</cfquery>
```

Listing 3

```
<!-- Query shows how many users from each domain -->
<cfquery name="qSubscribers" datasource="#request.dsn#">
  SELECT Email_Domain, COUNT(Email_Name) AS
CountOfSubscribers
  FROM newsletter_subscribers
  GROUP BY Email_Domain
</cfquery>
```

Listing 4

```
<!-- Initial a Structure to hold the return vars -->
<cfset ReturnVars = StructNew(>
```

```
<!-- loop over through the return string to get the
variable/value pairs using "&" as the delimiter -->
<cfloop list="#CFHTTP.FileContent#" index="Item" delimiters="&">

  <cfif ListLen(variables.Item,"=") gt 1>

    <cfset varname = ListGetAt(variables.Item,1,"=")>
    <cfset value = ListGetAt(variables.Item,2,"=")>
    <cfset "ReturnVars.#variables.varname#" =
variables.value>

  </cfif>
</cfloop>
```

```
<cfif IsDefined( "ReturnVars.Status" )>...
```

CODE LISTING

The code listing for this article is also located at

Next Month...

Don't miss the May issue!

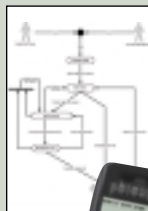
ColdFusion MX – A First Look

Macromedia has been talking about what used to be called Neo for almost two years now. Finally, this next-generation ColdFusion is in beta. **Ben Forta** introduces some of the most exciting new technologies and features.



ColdFusion and WAP

An introduction to WAP/WML development with ColdFusion. **Christian Schneider** explains how to develop a WAP-based e-mail interface.



ColdFusion Session Management

Philip Chalmers continues his series on using ColdFusion to solve some common session management problems in Web applications.



Query Custom Tags in ColdFusion

A simple way to write and debug your code using query custom tags, as demonstrated by **Reuben Poon**.

ColdFusion Developer's Journal

PACIFIC ONLINE

www.paconline.net

April 2002

What's Online

www.sys-con.com/coldfusion

CFDJ Online

ColdFusion Developer's Journal, the leading resource in the ColdFusion market, proudly brings you www.sys-con.com/coldfusion. Check out our Web site for up-to-the-minute news, events, and developments. Whether you consider yourself an IT professional or are just starting out, our Web site will spark your interest and keep you coming back for more.

Web Services Edge 2002 East

Register online for Web Services Edge 2002 East International Conference & Expo today and be a part of the largest Web services event in the world. This four-day conference, featuring the leading information technology professionals in the industry, will be held at the Jacob Javits Convention Center in New York City, June 24-27. Don't be left out of this major event!

Search ColdFusion Jobs

ColdFusion Developer's Journal is proud to offer an employment portal for IT professionals. Get direct access to the best companies in the nation. Find out about the "hidden job market" and how you can find it. As an IT professional curious about the market, this is the site you've been looking for.

Simply type in the keyword, job title, and location and get instant results. You can search by salary, company, or industry.

Need more help? Our experts can assist you with retirement planning, putting together a résumé, immigration issues, and more.

Answers Are a Click Away

Are you blocked by your blocking factor? Are your synchronization objects out of sync? Visit the **CFDJ Technical Forum** (www.sys-con.com/fusetalk/categories.cfm?catid=6) and see what other developers have to say. You can ask a question, share a tip, improve your code, find a way to solve a glitch, and network with ColdFusion professionals. Recent postings have generated online discussions on security issues, data importing, dialog boxes, submit buttons, image info tags, ColdFusion forms, and more.

The Latest in CF at the CFBuyersGuide.com

The most-read CF resource on the Internet, CFBuyersGuide.com (www.sys-con.com/coldfusion/wbg/index.cfm), is essential for those who want to learn about the newest books, software, tools, and services related to ColdFusion. Web hosting, testing tools, books, e-business software, custom tags, Web development tools, consulting services, education and training, and much more are featured at this comprehensive site. Why waste time searching the Net for your CF needs? Discover the convenience of CFBuyersGuide.com.



<dot.com>

- buyer's guide
- coldfusion forums
- mailing list
- coldfusion jobs
- coldfusion store

<magazine>

- advertise
- authors
- customer service
- editorial board
- subscribe

<content>

- archives
- digital edition
- editorial
- features
- interviews
- product reviews

<conferences>

- web services edge 2002
june 24-27
2002
- web services edge 2002
oct 1-3
2002

<search cfdj>

Search

<cfdj specials>

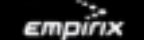


<bestsellers>

1. IBM WebSphere Application Server 2.4 Standard Edition. \$699.99
2. CFJ and CF Advisor: The Complete Works. \$229.99
3. AllWeb Application Platform Release 2.4. \$3,499.99
4. ColdFusion 5.0 Server Enterprise. \$4,999.99
5. ColdFusion 5.0 Server Professional. \$3,299.99

QUICK POLL

- What is Your Favorite New Feature of ColdFusion 5.0?
- ☐ User-Defined Functions
 - ☐ Query of Queries
 - ☐ Improvements
 - ☐ Integrated Charting
 - ☐ Performance Improvements
- Vote Results



Special online offers **SAVE UP TO \$100**
FOR MULTIPLE SUBSCRIPTIONS

Pick **4** or **5** and Subscribe
 for one **special low price**

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTION



WWW.SYS-CON.COM/2001/SUBOFFER.CFM
 Offer subject to change without notice

Once you're in it...

- Wireless Business & Technology
- Java Developer's Journal
- XML-Journal
- ColdFusion Developer's Journal
- PowerBuilder Developer's Journal

reprint it...



Contact Carrie Gebert
 201 802-3026
 carrieg@syz-con.com

Re Prints

SYS-CON MEDIA

Now in More than 5,000 bookstores worldwide

FOR FAST DELIVERY subscribe **Now!**

The World's Leading Independent WebLogic Developer Resource

Go Online and Subscribe Today!

WebLogic Developers Journal .com
 SYS-CON Media, the world's leading publisher of I-technology magazines for developers, software architects, and e-commerce professionals, brings you the most comprehensive coverage of WebLogic.

Helping you enable inter-company collaboration on a global scale

- Product Reviews
- Case Studies
- Tips, Tricks and more!

SPECIAL INTRODUCTORY OFFER
SAVE \$31*
 HURRY, DON'T DELAY! OFFER EXPIRES APRIL 30, 2002

***Only \$149 for 1 year (12 issues) regular price \$180.**

SYS-CON MEDIA

Tales from the List

CFDJ discussion threads...and their creators



BY SIMON HORWITH

ABOUT THE AUTHOR

Simon Horwith, a senior developer and Macromedia-certified ColdFusion instructor at Fig Leaf Software in Washington, DC, has been using ColdFusion since version 1.5. He is a contributing author to Professional ColdFusion 5.0 (WROX) as well as technical editor of The ColdFusion 5.0 Certification Study Guide (Syngress). This spring Simon plans to launch www.how2cf.com, a ColdFusion "how-to" site hosted by cfdynamics.

Welcome to the first installment of "Tales from the List," a monthly column devoted to the discussion threads (and their creators) on the **ColdFusion Developer's Journal** Mail List (CFDJList).

Each installment will examine popular discussion thread(s) from the list from the previous month. On occasion we'll profile an active member of the list, view list statistics, and that sort of thing. As this is the first article, I'll give a brief overview of the list and discuss how you, the reader, can play an active role.

SYS-CON Media's CFDJList began three years ago to support the growing community of ColdFusion developers. From its humble beginnings the list has grown to a current enrollment of over 600 active members. Although the overwhelming majority of the members are from the United States, many other nationalities are represented, including The United Kingdom, Germany, Saudi Arabia, France, China, Canada, and India. There is no cost for membership; simply send a blank e-mail to join-CFDJList@mail-box.sys-con.com, or visit www.sys-con.com/coldfusion/list.cfm and follow the instructions there.

List members exchange ideas and post questions and answers on a wide range of topics related to tasks commonly performed (and challenges encountered) by ColdFusion developers in today's workplace. On average, between fifty and a hundred e-mail messages are sent to the list on any given weekday; fewer are posted on the weekends (yes, there is life beyond programming).

Relevant topics include JavaScript, SQL, Web server configuration, ColdFusion Application Server administration, and, of course, CFML application development. Likewise, the level of expertise of list contributors ranges from newbie developers to some of the most seasoned Web developers in the business. Recently, the list archives section of the CFDJWeb site was overhauled. Old software was discarded, and the list now has a (FuseTalk-powered) Web-based forums interface, which allows nonlist subscribers to read and post questions. The archive also serves to make past and present list threads fully searchable.


For quite some time now, my involvement with the CFDJList has been that of a frequent contributor. When I began working with ColdFusion back in version 1.5, there really weren't many resources available to help developers new to the CFML language hone their skills. I began contributing to the list as a way to

offer the sort of advice to CF newcomers that I wish had been available when I began my Web development career. Over time, ColdFusion and its development community have grown in size and popularity (as did list membership). At the request of Robert Diamond, **CFDJs** editor-in-chief, I have become the official list administrator. He also requested that I use this column to share popular discussions and interesting discoveries contributed by list members.

As a certified Macromedia instructor, I try to leave each finished class with a few important pieces of advice to aid students in the workplace. The most important of which is never to spend more than 15 minutes on any development problem. The odds are that if you don't see the answer in the first five to fifteen minutes, you're probably not going to find the answer anytime soon – so stop banging your head against the wall. When you do encounter a problem (don't worry...it happens to the best of us), the best thing to do is seek a second pair of eyes or get advice from someone who's been down that road before. This help might be in the form of an online knowledge base, a co-worker, a favorite reference book, or maybe the CFDJList.

I recommend that any of you who aren't active members of a mail list now subscribe to one. If you don't find yourself banging your head against the wall very often, great...your insight will be invaluable to members of the list who need advice. If you're just learning, find yourself spending long hours debugging code or doing research, or just have a lot of questions or ideas you'd like to bounce off other developers, then you'll likewise find that you can serve an important role on the CFDJList. We're all part of a community, and it's only through supporting one another that any of us can become better at what we do.

I'd like to take this opportunity to say "thank you" to all of the current members of the list who make being part of the ColdFusion development community so rewarding and so much fun, and "welcome" to those of you who haven't yet joined the CFDJList but plan to in the future.

Look for this column next month when we examine recent popular list discussions and their resolutions. 

@ SHORWITH@FIGLEAF.COM

**SIMPLEX KNOWLEDGE
COMPANY**
www.skcc.com

CF USER GROUPS

Birmingham, AL Jeff Bryant jeff.bryant@mindspring.com	Gainesville, FL Taner Aktas aktas@ufl.edu	Southern Maryland Russ Mahan rmahan@erols.com	Mid-Ohio Valley Dave Hannum hannum@ohio.edu	Northern Virginia Bret Peters novacfug@figleaf.com	Hong Kong Shashank Tripathi shashank.dp.tripathi@hk.pwcglobal.com
Huntsville, AL Brian Yagar brian.yagar@redstone.army.mil	Jacksonville, FL Russ Johnson russ.johnson@cyberfusionsolutions.com	New England Developer's Network Steve Casco steve@ateaz.com	Cleveland, OH Matt Newman mnewman@newchanneltech.com	Richmond, VA Robert Baird bbaird@btgl.net	Hungary Zoltan Sheshler ozoltan@yahoo.com
Phoenix, AZ Sean Tierney sean@pubcrawl.net	Orlando, FL Randy Drisgill rdrisgill@vshift.com	Detroit, MI Matt McDonald matt@stoneage.com	Columbus, OH Angelo McComis Angelo@CompuServe.com	Figleaf CFUG Bret Peters cfug@figleaf.com	India Zahid Java zahidjava@yahoo.com
Tucson, AZ Darcey Spears dmspears@uswest.net	South Florida Kevin Langevin kevin@cfug-sfl.org	Mid-Michigan Group Rick Mason Rick@SeedChoices.com	Oklahoma City, OK Glen Collymore Glen-Collymore@ouhsc.edu	Seattle, WA Shannon Smith president@cfseattle.org	Indonesia Agus Basuni agus@digitaldevelopment.com
Bay Area, CA Nathan Dintenfuss nathan@changemedia.com	Tallahassee, FL Benjamin Bloodworth ben@electronet.net	Twin Cities, MN Dan Chick dan@danchick.com	Tulsa, OK Casey Bourbonnais casey@cfbenterprises.com	Walla Walla, WA Sarah Lewis sarahlewis@alghena-design.com	Ireland Daryl Fullerton daryl@thebiznet.net
Southern California Leon Chalnack lchalnack@advantasolutions.com	Tampa Bay, FL Kenneth Beard kbeard2323@hotmail.com	Kansas City, MO Bryan Laplante directors@kcfusion.org	Portland, OR Jeff Cram jcram@isitedesign.com	Milwaukee, WI Ron Kurtus ron@ronkurtus.com	Dublin, Ireland Justin MacCarthy maccarthy@iol.ie
Southern California - Los Angeles Sid Sheres sheres@gle.net	Atlanta, GA Adam Churvis president@prodenhance.com	SoMoCFUG Jeremiah Andrick jandrick@theinteriorlife.com	Roseburg, OR Sharon Bland sbland@douglassesd.k12.or.us	Wyoming Matt Rodosky mrodosky@wyoming.com	Italy Andrea Veggiani andrea@dinamica.it
Orange County, CA Peter Amiri peter@amiri.net	Atlanta, GA Seth Land gacug@figleaf.com	St. Louis Metro Area Dan Anderson dan@pswebstudio.com	Harrisburg, PA Neil Ross info@codesweeper.com	Nepal, Asia Sudhir Raj Joshi sjoshi@iapex.com	Okinawa CFUG Michael Bradford michael@devapps.com
Sacramento, CA Geoff Lilley geofflilley@hotmail.com	Columbus, GA Rob Seebach rseebach@sfts.com	Helena, MT Tom Marino tmarino@falcons.com	Philadelphia, PA Chris Swansen cswansen@etechsolutions.com	Australia Marc Dimmick marc@baseglobal.com	Korea HyeHyun Seo hhyseo@otozone.com
San Diego, CA Robert Munn rob@illumineez.com	Boise, ID Paul Jenkins pjenkins@dol.state.id.us	Omaha, NE Mark Kruger Mkruger@CFWebtools.com	Pittsburgh, PA Kyle Jenkins kyle@vaporbiz.com	Canberra, Australia Kunal Bhatia khatia3@csc.com.au	Malaysia CFUG Halimi Yasin affroman@affroman.com
Orleans, Ontario Clovis Paquette clovispa@comnet.ca	Chicago, IL Robert Burns bob-burns@mediaone.net	Las Vegas, NV Dave Byers pegarm@lvcn.com	State College, PA Brian Panulla brian@elmwoodmedia.com	New South Wales, Australia Kym Kovan kymk@microset.com.au	Netherlands Ron Pasch ronnieo@dds.nl
Edmonton, Alberta Jeff Acker jeff.acker@topham.ab.ca	East Central Illinois William Steiner WilliamS@hkusa.com	Concord, NH Neil Giarratana NGiarratana@ininet.com	Greenville, SC David Goldfield Dave@goldfield.com	Northern Territory, Australia Fred Lecul flecul@paspailey.com.au	Auckland, New Zealand Amanda Irvine amanda.irvine@delta.co.nz
Lethbridge, Alberta Chris Sorensen chris@csystems.ab.ca	Quad Cities Jamie DeVolder JamieD@4CS.COM	Central New Jersey Hien Nguyen HNgyen7@prius.jnj.com	Knoxville, TN Doug McCaughan doug@colknex.com	Queensland, Australia Barry Moore barry.moore@yahoo.com	Romania Andrei Oprea andrei@oprea.org
Montreal, Canada John Kopanas kopanas@videotron.ca	Springfield, IL Gary Ashbaugh ashbagd@mail.loc.state.il.us	Northern New Jersey Cindy Pomarlen cjp@neotech-associates.com	Memphis, TN Michael Kubicki mkubicki@univ-solutions.com	Southern Australia Brett Hales brett.hales@baesystems.com	Johannesburg, South Africa Steven Ringo stevan@tutuka.com
Ottawa, Canada James Milks jamesm@videotron.ca	Central Indiana Joshua Kay josh@dataquix.com	South Jersey Mark Stewart m-stewart@home.com	Nashville, TN Chris Mosier chris.mosier@e-maginative.com	Tasmania, Australia Mark Heseltine HeseltineM@logica.com	Spain Jorge Pueyo jpueyo@lander.es
Toronto, Canada Kevin Towes KTOWES@PangaeaNewMedia.ca	Northern Indiana Graham Pearson gpearson@nfcug.org	Albuquerque, NM Christopher Jungmann jungmann@swcp.com	Austin, TX Mike Dickinson cranch@texas.net	Victoria, Australia Jordan Thomas jordan@vector7.com.au	Sweden Stefan Gudmundsson stefan@cfug-se.org
Vancouver, Canada Cameron Siguenza cameron@evolutionb.com	Cedar Valley CFUG David Chandler David@ChandlerResearch.com	Albany, NY Thomas McKeon tmckeon@newkirk.com	Dallas, TX Patrick Steil pmsteil@imailbox.com	Belgium Denis Wauthy d.wauthy@switchon.com	Zurich, Switzerland Martin Buerlimann bue@sercon.ch
Colorado Springs, CO Anthony Starkey astarkey@mentiscorp.com	Des Moines, Iowa Kevin Schmidt schmidt@hungrycow.com	Long Island, NY Christopher Collins chrisc@sitespan.com	Houston, TX User Groups usergroups@allaire.com	Rio de Janeiro, Brazil Marello Frutig frutig@astrolabio.com.br	Taiwan Jack Jair jack@leetide.net
Denver, CO Robi Sen robi@rarularity.com	Louisville, KY Tim Newton TNewton@MercuryFusion.com	New York, NY Michael Dinowitz mdinowitz@houseoffusion.com	San Antonio, TX Douglas White doug@dwwhite.ws	Sao Paulo, Brazil Marcantonio Silva marco@channelmedia.com.br	Bangkok, Thailand Paul Hastings paul@tel.or.th
Rocky Mountain Mike Miller mmiller@ewtechnology.com	New Orleans, LA Kathy Hester kathyhes@yahoo.com	Rochester, NY Dave Horan dhoran@fusionproductions.com	Utah State College Jon Nelson jnelson@nstep.net	Central Europe Sven Slazenger slazenger@interlake.net	Turkey Oguz Demirkapi oguz@cftr.net
Hartford, CT Chris Graves graves@rapidcf.com	Annapolis, MD Matt Schuster mattschuster@erols.com	Syracuse, NY Laurie Ferger lferger@twcny.rr.com	Salt Lake City, UT Mike George ueberhund@hotmail.com	China Feng Jian Feng webmaster@cfwindow.com	Dubai Maryam Ebrahimi CFUG@digiba.com
New Haven, CT Alfred DiMarzio cfug@hbgraphics.com	Baltimore, MD Bret Peters bmorecfug@figleaf.com	Charlotte, NC Dave Crawford dcrawford@charlotte-cfug.org	Provo, UT Josh Trefethen josh@exciteworks.com	Finland Tero Pikala tero.pikala@kolumbus.fi	London Niklas Richardson niklas@ukcfug.org
Stamford/Westchester, CT/NY Barry Hyman bah@abodofoto.com	Broadneck High School CFUG Rick Blaha IRICKY@aol.com	Greenville, NC Steven Forehand forehands@mail.ecu.edu	Vermont Sarah Moulton info@mbbytes.com	France Fabien Graille FG500001@exchange.FRANCE.NCR.com	Northern England Stephen Moretti stephen@cfug.org.uk
Delaware John McKown john@delaware.net	Maryland Michael Smith Michael@teratech.com	Raleigh, NC Cheryl Elia ccfug.mgr@listserv.unc.edu	Hampton Roads, VA Raymond Camden morpheus@deathclock.com	Ramstein Air Base, Germany Tom Nunamaker tom@nunamaker.com	Southern UK Stefan Elliott cfug@four-runner.com

THE LARGEST INDEPENDENT

JAVA

DEVELOPER CONFERENCE IN THE WORLD!

**WIN A
\$35,000
LUXURY CAR!**



ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING
TO WIN AND RIDE OFF IN A \$35,000 LUXURY CAR!

JAVA IN JUNE

ESPECIALLY IN NEW YORK

REGISTER ONLINE TODAY

FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

VISIT **WWW.SYS-CON.COM**

Focus on Java

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

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

The Java Fast Facts on June 24, a Java-focused CEO Roundtable Panel, and comprehensive conference sessions will focus you on Java-only information-packed day!



A Sampling of Java-Focused Sessions

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



ALAN WILLIAMSON
JAVA CHAIR • EDITOR-IN-CHIEF
AND DEVELOPER'S JOURNAL

Featuring...

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

Who Should Attend...

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

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



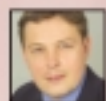
TYLER JEWELL
PRINCIPAL TECH
EVANGELIST • BEA



GREGG KIEBLING
CEO
SITRAKA



DAVID LITWACK
CEO
SILVERSTREAM



BARRY MORRIS
CEO
IDIA



GREGG O'CONNOR
PRESIDENT
SONIC SOFTWARE



RICK ROSS
FOUNDER
JAVALOBBY



JAMES DUNCAN
DAVIDSON
FATHER OF ANT

For Exhibit Information

CONTACT: MICHAEL PESICK
135 CHESTNUT RIDGE RD.
MONTVILLE, NJ 08045
201 842-3057
MICHAEL@SYS-CON.COM

JDJEDGE
conference & expo

JUNE 24-27

JACOB JAVITS
CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3

SAN JOSE
CONVENTION CENTER
SAN JOSE, CA

SPONSORED BY:

IONA | END 2 ANYWHERE™

ADOS

bea

TogetherSoft

SilverStream

Actional

MERANT

Altaworks

PolarLake
Enterprise Strength XML

ALTOVA sm3 app

MEDIA SPONSORS

Federal Computer Week

WebServices.org

XML TIMES.com

Java Skyline
Register for Java Skyline

CE Advisor

WebServicesMail

WIRE

XML.ORG

WROX

port

SDTimes

BASIS

wireless

JAVA

WebLogic

XML

WebSphere

WebServices

COLDFUSION

COLDFUSION

Virtual Dressing Room for Victoria's Secret

(San Francisco) – A new online application by Victoria's Secret, the specialty retailer of intimate apparel and beauty products, is using Macromedia Flash and ColdFusion to deliver a beneficial user experience for the company's new Swim Separates collection.

The collection enables people to mix and match swim tops and bottoms in different styles, colors, and sizes, then purchase their favorite combinations. The Swim Separates application is available at www2.victoriasecret.com/swim/separates/index.cfm.

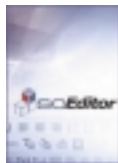
Victoria's Secret uses Macromedia ColdFusion Server to update the Macromedia Flash experience in real time to reflect the

store's real-time pricing and inventory, indicate which alternate colors are available, and manage the purchase process without taking users out of the application.

www.macromedia.com

SiteObjects Releases soEditor 2.1

(Lansing, MI) – SiteObjects, Inc., a leading provider of Web-based development solutions, has released SiteObjects soEditor 2.1, a ColdFusion custom tag that turns an ordinary form field into a WYSIWYG browser-based HTML editor, allowing users who have no HTML knowledge or programming expertise to create, modify, and maintain content in a Web site.



Macromedia Unveils Macromedia Flash MX

(San Francisco) – Macromedia, Inc., has announced Macromedia Flash MX, with



support for video, application components, and accessibility, to facilitate the creation of expressive online content and applications. The product reportedly simplifies the process of visual authoring for traditional Web developers who want to use client technology for their applications. Macromedia Flash supports industry standards and provides users with prebuilt user interface components, accelerating development.

www.macromedia.com

macromedia what the web can be™



SiteObjects soEditor 2.1 can be installed easily by any ColdFusion programmer, and it's available in both "Lite" and "Pro" versions. www.siteobjects.com



Events

CF North Developers Conference

May 4-5, Toronto
www.cfconf.org/CFNorth/

CF Cruise

May 19-26, Caribbean
www.cfconf.org/cf_cruise/

CFDJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
ACTIVE PDF	WWW.ACTIVEPDF.COM	949.582.9002	4
CFDYNAMICS	WWW.CFDYNAMICS.COM	800.422.7957	11
COLD FUSION DEVELOPER'S CONFERENCE	WWW.CFNORTH.COM		37
COLD FUSION DEVELOPER'S JOURNAL	WWW.SYS-CON.COM	800.513.7111	25
EMPIRIX	WWW.EMPIRIX.COM/DOUBLE/CFM	866.228.3781	2
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM	877.215.HOST	15
INTERLAND	WWW.INTERLAND.COM	800.335.1327	3
INTERMEDIA.NET	WWW.INTERMEDIA.NET	800.379.7729	52
JAVA DEVELOPER'S JOURNAL	WWW.SYS-CON.COM	800.513.7111	17
JDJ EDGE CONFERENCE & EXPO	WWW.SYS-CON.COM	201.802.3069	49
JDJ STORE	WWW.JDJSTORE.COM	888.303.JAVA	20, 21
MACROMEDIA	WWW.MACROMEDIA.COM/GO/USERGROUPS	877.460.8679	13
MACROMEDIA	WWW.MACROMEDIA.COM/GO/MASTERING	877.460.8679	27
MACROMEDIA	WWW.MACROMEDIA.COM/DOWNLOADS	877.460.8679	51
PACIFIC ONLINE	WWW.PACONLINE.NET	877.503.9870	43
PAPERTHIN	WWW.PAPERTHIN.COM	800.940.3087	31
QUILL DESIGN	WWW.QUILLDESIGN.COM		9
SIMPLEX KNOWLEDGE COMPANY	WWW.SKC.COM	845.620.3700	47
SYS-CON REPRINTS	WWW.SYS-CON.COM	201.802.3026	45
WEB SERVICES EDGE CONFERENCE & EXPO	WWW.SYS-CON.COM	201.802.3069	18, 19, 41
WEB SERVICES JOURNAL	WWW.SYS-CON.COM	800.513.7111	29
WEBLOGIC DEVELOPERS JOURNAL	WWW.SYS-CON.COM/WEBLOGIC	800.513.7111	45
XML EDGE CONFERENCE & EXPO	WWW.SYS-CON.COM	201.802.3068	39
XML JOURNAL	WWW.SYS-CON.COM	800.513.7111	26

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of ColdFusion Developers Journal. All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in ColdFusion Developers Journal. Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc.



www.wbt2.com

SUBSCRIBE NOW

www.javadevelopersjournal.com

TO THE

www.sys-con.com/xml

FINEST

www.coldfusionjournal.com

TECHNICAL

www.sys-con.com/pbdj

JOURNALS

www.webspheredevlopersjournal.com

IN THE

www.wldj.com

INDUSTRY!

www.wsj2.com



subscribe online www.sys-con.com or call 800 513-7111



wireless | java | xml | coldfusion | powerbuilder | websphere | weblogic | web services

Macromedia

www.macromedia.com/downloads

Intermedia.net
www.intermedia.net