SERVER MIGRATION MADE EASY

U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

ColdFusionJournal.com

January 2002 Volume: 4 Issue: 1

SYS-CON MEDIA

ColdFusion Developer's Journal

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
COLDFUSION DEVELOPER'S JOURNAL (ISSN #1523-9101)
is published monthly (12 times a year)
for $89.99 by **SYS-CON Publications, Inc.,**
135 Chestnut Ridge Rd., Montvale, NJ 07645

postmaster: send address changes to:
COLDFUSION DEVELOPER'S JOURNAL
SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645

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

Looking Back…and Forward…

BY **ROBERT DIAMOND**

Well, we're at the start of a new year, and before we look forward, I thought it might be an appropriate time to look back and reflect on <CF_TimeGoneBy>. With this January issue we enter our fourth year of publication, proudly covering the world of ColdFusion since January of 1999.

Back when ***CFDJ*** launched, the technology world was very different. If you were running ColdFusion, you were probably using version 3.0 and waiting for the first set of 4.0 patches on your path to upgrading. JRun was a lovely Java server owned by LiveSoftware. The Allaire/Macromedia merger was nary a glimpse in anyone's eye. And the word Neo that we now hear all the time would mean absolutely nothing! Oh, the horror. Worse than all that, you'd be getting a new issue of ***ColdFusion Developer's Journal*** only every other month! Absolute worst of all – for the first few issues of *CFDJ*, you didn't have me as editor-in-chief. It's a wonder we all survived. ;-)

Now in the modern world of ColdFusion, a copy of *CFDJ* arrives at your door every month, you're running ColdFusion Server 5, and eagerly awaiting Neo. Speaking of good things, we have a great issue this month as well…

We have two articles from Charlie Arehart – so how could this not be a great issue? His cover story is about better CF Server administration, and if you're operating your own ColdFusion Server, you'll certainly want to take a look. His **Journeyman** column – one of my personal favorites – discusses adding in new Help topics to CF Studio. For those of us who don't program everyday, or aren't Ben Forta, these Help topics can provide an invaluable reference to the exact format and functionality of ColdFusion tags. With Charlie's help, you can get the same great wealth of knowledge about JavaScript, SQP, and DHTML.

Ben Forta discusses search engines and ColdFusion, a topic of interest to just about every Web site proprietor. Steven Lewis writes on protecting images to prevent people from hot linking to your site and using your images without permission. These are helpful tips if there's a spike in your image traffic – but not on your pages. Keep those bandwidth stealers away.

Hal Helm's Fusebox Tutorial is the first in a series of articles that will walk you through building a Fusebox application. Susan Matteson reviews *Core ColdFusion 5* by Eben Hewitt, which comes in at a hefty 969 pages. Anton Prakash has written a fantastic feature on migrating CF from an NT platform to Linux or Unix. Since he recently completed that task himself, he wanted to present his experiences.

To close out this month's editorial, from time to time (usually from a member of his family) a reader will wonder what happened to Chad Sitler, the man who helped launched *CFDJ*. He now works in Charlotte, NC, for Knight Ridder Digital. He looks back on his time at *CFDJ* fondly – "Working with all the folks associated with *CFDJ*, from SYS-CON staff to writers to readers, was a great growth opportunity for me. It was truly a blast!"

ABOUT THE AUTHOR

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

@ ROBERT@SYS-CON.COM

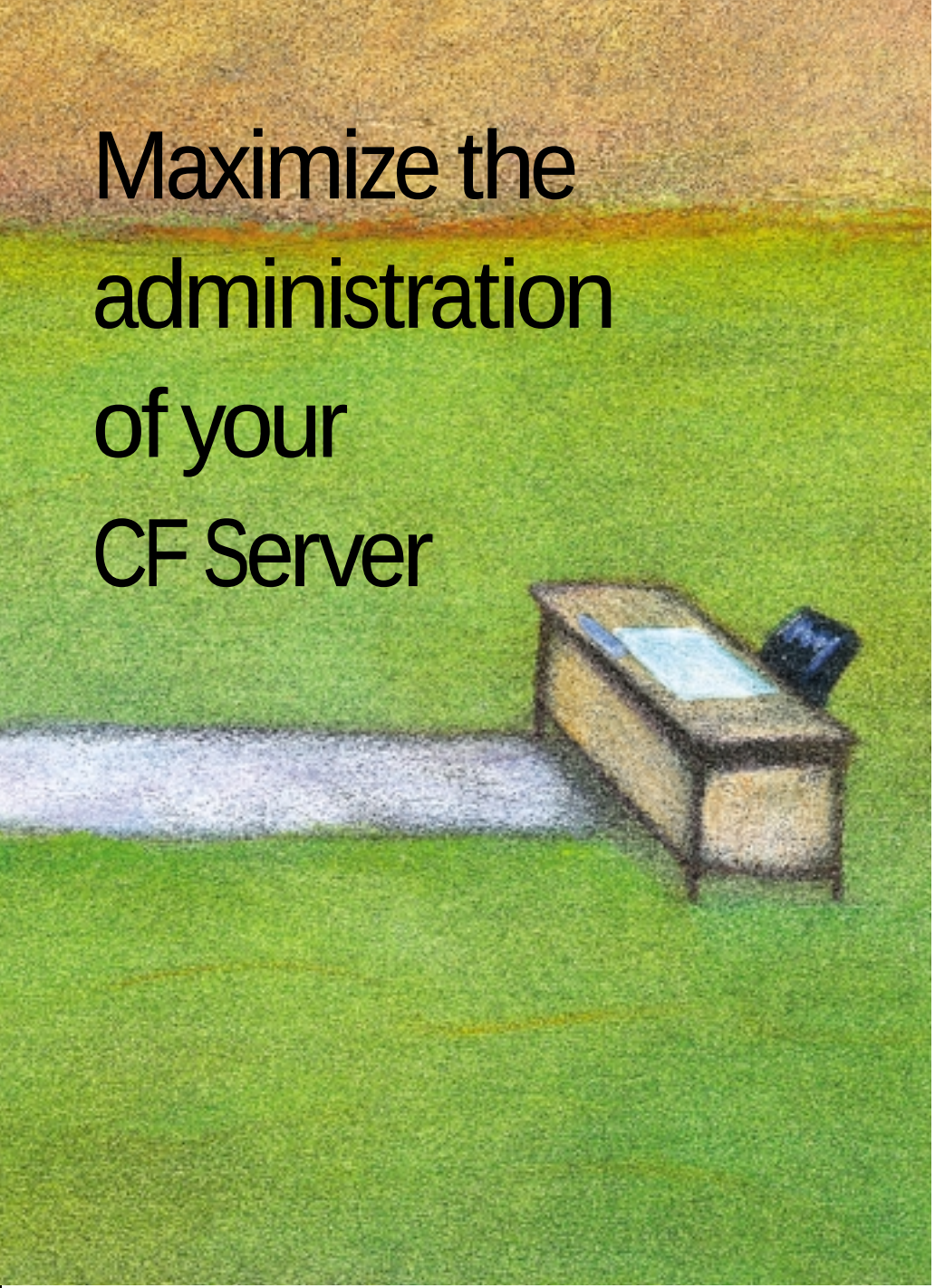
ColdFusion Feature

By Charles Arehart

Part 1 of 2

Toward Better CF Server Administration

Maximize the administration of your CF Server



Are you a CF Server administrator? Or a developer with an interest in knowing how your server is configured and managed? Are you aware of all that should be done to keep the server running well – or all that could be done to make the most of it?

One View of CF Administration

In this two-part article we'll look at some of the things CF administrators should be paying attention to, as well as some of the things they should take advantage of but may not have noticed. We'll discuss aspects of managing the server that extend beyond the CF Administrator interface, including tasks for keeping the server environment healthy and running effectively.

What this will *not* be is a boring review of each page in the CF Administrator, nor will we get bogged down in trivialities like how to create a data source. The Macromedia ColdFusion manuals (and online help in the Administrator) do a fine job of providing that sort of "reference" material. (If you haven't read the manuals in CF5, they've improved and there's now context-sensitive help within the Administrator. More on that later.)

Instead, this article brings up a variety of topics that should interest a typical CF administrator. Some topics will revolve around settings in the Administrator, while others involve tasks outside that interface (and some not specific to ColdFusion). We'll look at all these from the perspective of two broad categories, performance and security.

We'll include some key reminders and hidden tips, with a summary of these presented at the end of Part 2 for easy reference. I'll conclude the article with a discussion of where to learn more about these things and how to stay updated on changes (new features, fixes, bugs, etc.).

No single article could be all encompassing with regard to administering a CF Server. With two administration books in the Macromedia CF 5 documentation and all the online help now supplementing that, there's certainly a lot I could cover. And there's still more to the role that involves tasks not even documented in the manuals. My hope here is to at least raise some key points, perhaps highlighting important aspects you may have missed, leaving you to investigate them further.

If you think I've left out something substantial, please use the comment feature at the online version of this article on www.sys-con.com/coldfusion/. Better still, consider writing an article to expand on any topic of interest.

Different Strokes?

Before exploring the technical aspects of managing a CF Server, let's clarify that different people will have very different perspectives on what the job entails. It can either be a daunting and time-consuming chore or a trivial afterthought, depending on many things. These include the number of applications running and their complexity, the volume of traffic from site visitors, and your skills as an administrator and those of the developers working in the environment.

The nature of the server environment also influences what the role entails. A corporate environment running a single application is quite different from one that supports apps for several departments, which is different again from a hosted environment running a shared server with dozens of apps.

Indeed, there is a wide variety of potential audiences for a discussion of administrator topics. You could be:

- A full-time administrator doing nothing but CF administration
- An administrator with responsibility for many (or all) aspects of the server environment
- Someone who supports CF only as a side role, perhaps reluctantly
- A CF developer who also administers the CF Server
- A CF developer with no admin rights but interested in how the server is configured

That last role opens the potential audience for this topic far beyond just administrators. Indeed, I hope all developers will read on and continue to learn more about administering a CF Server. There really are a lot of configuration settings and tasks to be handled by an administrator that can affect your programming abilities.

Before moving on, let's clarify that if you are responsible for administering a server remotely, you can run the Web-based CF Administrator interface by way of the http://<servername>/CFIDE/Administrator/index.cfm. You'll be prompted for a password that's set at the installation of CF (or that can be changed within the Administrator, as we'll discuss in Part 2). Figure 1 shows the Administrator interface as of ColdFusion 5.

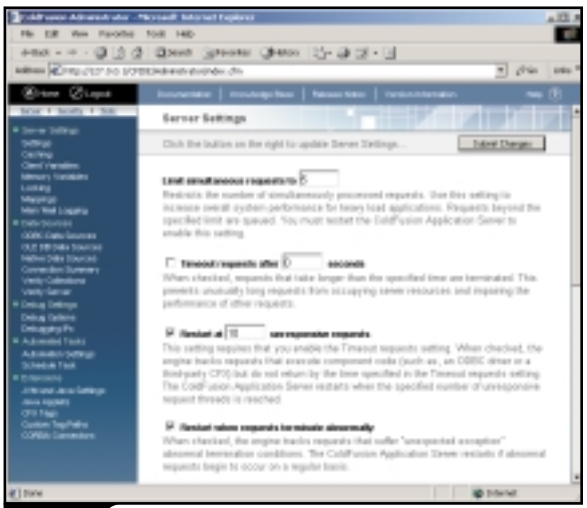


FIGURE 1: CF Administrator interface

For security reasons, you may want to restrict access to the Administrator beyond the simple built-in password protection with either Web server or operating system security. (The Knowledge Base article on this topic, ID 10954, and any other KB article discussed here can be found at www.allaire.com/support/knowledgebase/searchform.cfm [the form for searching by KB ID is at the bottom of the page]).

A Variety of Administration Settings and Tasks

We can classify administration settings and tasks in a number of ways. The Administrator interface breaks tasks down into many categories, but for the sake of this article – and in the hope of helping to make things a little clearer for the newcomer or itinerant administrator – I've broken them into the following groupings:

- Developer-oriented
- Performance-related
- Security-related
- Miscellaneous

We don't have room in this article to delve into all these topics, even at an overview level, so Part 2 will cover the security and miscellaneous settings and tasks.

I hope anxious readers interested in performance and security won't skip the next section on developer-oriented tasks. While such features may seem less important to you, they're very important to your developers and may not be that obvious.

Developer-Oriented Settings and Tasks

Most developers will be familiar with at least one task that a CF administrator must perform, and many administrators may (dangerously) limit their activities to only this: managing data sources. It's a vital task for CF developers since it's hard – though not impossible – to do much query-based application development without a data source.

Notice that I say "not impossible," because a new feature of CF5, the DBTYPE=" dynamic" and ConnectString attributes of CFQUERY, allow access to a database without a defined data source. That's something that will surprise many developers (and administrators).

Indeed, one of the main points I'd like to raise in this article is that there may be aspects of managing a CF server that would surprise both developers and administrators. CF is constantly changing, and many administrators limit their activity to only the bare necessities – or just those things that the developers on the server ask about. It's good for developers to know what they can ask for, and for administrators to know of things that developers can do (like that dynamic data source capability) if the administrator is not paying careful attention.

Because of the potential for abuse, it's also one of the many things that can be restricted by an administrator, along with nearly a dozen tags; in Part 2 we'll discuss what CF calls "basic" security.

Getting back to the subject of data sources in general and connection strings in particular, another new facet of CF5 is that connection strings can also now be entered in the data source definition. This can help solve the problems of connecting with some database servers as well as provide connection attributes not otherwise supported in the Administrator interface.

Creating data sources is covered in much more detail in the new CF5 Administrator manuals, including such subjects as creating optimal definitions for particular databases and drivers, using OLE DB as an alternative to ODBC where appropriate, using the new Merant Wire Protocol drivers in CF5, and using native drivers.

Accessing Remote Data Sources from Your LocalHost Server

Before leaving the subject, I'd like to point out a couple of topics that may interest developers who have their own local copies of CF Server, but are working in an environment with remote or LAN-based database servers. You probably write your code to leverage data sources defined on the central CF Server where you deploy your code, but you may be able to define a data source on your local CF Server.

This can be very helpful not only for testing your code locally, but also for accessing CF Studio's ability to view a database, its structure and data, and use Studio's query builder against it. You may not have RDS access to that remote CF Server, but it's generally pretty easy to set up RDS access to your own local server. With data source definitions on your local server pointing to remote databases, you can get to that database and perhaps avoid bothering with a remote RDS connection.

Also, note that if you want to connect to databases using native drivers (part of the Enterprise version of ColdFusion) you need to install the appropriate client drivers for the intended DBMS. This is covered in detail in CF 5's *Advanced CF Administration* book.

BEA Eworld

www.bea.com/events/eworld/2002

Various Developer-Related Settings

Several other administration matters relate to how developers can perform their work on the server. They include managing:

- Mappings (for making CFINCLUDE and CFMODULE)
- C++ and Java custom tags (CFXs)
- Verity collections
- Site-wide error handling and missing template handlers
- Java settings to support CFOBJECT Java calls and CFX Java custom tags
- Server-side debugging settings (be aware of some security aspects of turning on debugging, discussed in Part 2's security section)

If you're not familiar with each of these capabilities, your developers may not be getting the most from the programming capabilities of ColdFusion so you should learn more about these settings. There have been several ***CFDJ*** articles on each of these subjects (including Java configuration settings and CFXs, setting up debugging and error handling, and managing Verity collections).

A few other aspects of Administrator settings relate to enforcing software quality checks on developers. These include a strict attribute validation feature (which is related to some changes introduced in Release 4) and the persistent memory variable locking support that can either check that all needed locks are provided in code, automatically enable read-only locks where appropriate, or turn on single-threaded sessions.

The last one eliminates any risk of concurrent updates to a given user's session scope by multiple browser requests from a single user. These can occur as a result of a framed page, a user having multiple browser windows open or refreshing a page before it's completed, or from two users sharing the same session (whether through coding mistakes or subversion).

There are also a handful of other settings in the remaining categories that can influence developers and their programming. Let's move on to the performance-oriented category.

Performance-Oriented Settings and Tasks

While CF is easy to install and administer, there are many default settings in the Administrator that may not best serve your environment. You should thoroughly understand and investigate the following settings. Many of them could have a significant impact on existing applications, so use care when changing them. Some of the settings are self-explanatory, but each is explained in the "Installing and Configuring CF Server" manual (or related manual in previous releases). In some cases, I offer some information or recommendations that may not be self-evident. The settings include:

- Limiting the number of simultaneous requests running on the CF Server.
- The number of seconds after which a long-running request should be timed out.
- The number of unresponsive requests to be allowed before restarting the server.
- Restart when requests terminate abnormally.
- The template cache size (how much space to allocate to holding the compiled/interpreted version of all executed templates); determining this size can seem a black art, but the "performance monitor" or "server reports" feature described later indicates that it needs to be increased if the number of "cache pops" is nonzero.
- The "trusted cache" setting can improve performance by not looking for changed code templates once interpreted, if the code is not expected to change during the server run.
- Limiting the number of minutes a cached database connection can remain inactive before it's released.
- Limiting the maximum number of cached queries allowed to be stored on the server (how many unique instances of queries using CACHEDWITHIN or CACHEDAFTER may be cached).
- Where client variables are stored by default (whether in the registry, a given database, or client cookies, which can be overridden by a given application on the CFAPPLICATION tag).
- Specifying the number of days before purging unused data in client storage, as well as whether to disable performing updates to certain predefined client variables for every visitor ("hitcount" and "lastvisit") for client variables stored in a database.
- The default and maximum timeouts allowed for application and session variables; programmers can override the default in CFAPPLICATION but if they set a value higher than the maximum, it's ignored. The maximum time for session timeouts on installation of CF is only 20, which may be too low for many applications. Consider increasing it to 60.

Other performance features of the administrator include:

- Whether to log slow pages to server.log for analysis and performance tuning.
- The available logs that track application and server errors and other monitoring data (found in the cfusion/logs directory by default).
- How to view, search, archive, and purge those server logs as well as how to download them from remote servers (features all enabled in the new "logs" area of the CF5 admin).
- Whether to use operating system logging facilities for performance monitor counters (if you're not watching the statistics, don't bother collecting the data).
- How to view those performance statistic samplings made available in the "monitor" area of the CF5 Administrator, or within the NT performance monitor or cfstat command line utility (for other platforms).
- How to set up and use the new probe and alert features of CF5 to monitor server health and cause automatic repair, notification, or other actions.

> **"It's really amazing how many people either program with or administer CF without ever reading the manuals"**

A couple of less obvious settings may also influence performance. Be careful setting the probe intervals (how often a probe fires, if probes are enabled) as well as the scheduler refresh interval (how often CF looks to see if scheduled tasks are ready to run). If either is set lower than you really need, you're asking CF and related services to perform more work than necessary. Another less obvious matter is that of taking care to repair, optimize, and perhaps purge Verity collections, as appropriate.

Besides these global settings, there are a couple of data source–specific settings that can also have an effect on performance, including "limit connections" and "maintain database connections," which are both set with the "CF Settings" button during data source definition. A performance-related choice is whether to use either OLEDB, native, or the new Merant drivers where appropriate.

Non-Administrator Performance Tasks

I had said that the article wouldn't be a review of the pages in the Administrator, but that was certainly a review of many settings controlled by the Administrator. There are several other aspects of managing a CF Server that can influence performance. Some require active involvement, while others are built-in enhancements in CF. Among the latter are CF5's improved internal performance, reduced memory footprint, and better memory management. Similarly, CF5 adds SNMP (or MIB) support that facilitates viewing of CF performance information within centralized IT management products, as well as enhanced hardware load-balancing integration.

Indeed, if you have a very high traffic site, you should consider either hardware- or software-based load balancing. Since Release 4, the Enterprise version of ColdFusion has had software-clustering capability available with its integrated Bright Tiger support. Both types of load-balancing (and failover) features are discussed at length in the advanced ColdFusion Admin manual.

Another (perhaps less scalable) way to support larger traffic loads is CF's available "distributed mode." Discussed thoroughly in the Release 4 CF Administration manual or available in a PDF for download via KB 21966, this feature enables a CF Server to run on a server that's separate from the Web server, or indeed for several Web servers to share a single CF Server.

Whether you use clustering or any other means of distributing CF Server requests, another widely regarded truism is that you should not install multiple servers (such as CF Server, database servers, or a mail server) on the same physical box. The operating system and each service will perform much better when fewer services are contending for limited resources (CPU, memory, and disk processing).

Many other facets of performance are more specific to your particular server operating system. Again, the Knowledge Base comes to the rescue in article ID 1172, "Platform-Specific Performance Settings." It's from 1999 but still timely in many ways.

Before leaving the subject of performance-related settings and tasks, there are some additional logs that you should monitor to ensure both the performance and stability of the server environment. You may not notice them because they're not CF logs, per se. They're installed in support of the new CF5 graphing feature and – in the Enterprise version of CF – the aforementioned archive, reporting, probe, and monitoring features. Each of these is enabled by way of an embedded instance of Macromedia's JRun Java Server engine. While there's not much you can (or should) do with respect to these embedded JRun Servers, be aware that they each create their own logs in the CFUSION\jrun\logs directory, which you should monitor.

There's also a known problem with the management service that can cause the server on which it (and the corresponding CF Server) is installed to run to 100% CPU utilization, thus locking the server. There's a fix in the form of a patch available from Macromedia, and that leads nicely to our last topic.

Learning More, Staying Updated

Even the most well-informed and savvy CF administrators can't rest on their laurels. There are always new Knowledge Base articles, security bulletins, and patches (or hot fixes) released to help keep your server running in top form. It's incumbent upon you as a CF administrator to keep abreast of these changes and notices. In some instances (the case of security bulletins), there's a notification service you can join to be informed as soon as they're released.

Regarding patches (or hot fixes), they're a relatively new phenomenon. Previously, most changes to CF were introduced in point releases. With the availability of patches (software updates related to specific problems), we can get targeted resolutions to problems more quickly. The only challenge is finding if they're available. They're discussed in KB 20371. It's up to you to keep an eye out for them. (You may hear about them more quickly if you join a mailing list of other CF enthusiasts, discussed in a moment.)

In the case of Knowledge Base articles, the location for finding them was mentioned before. One trick you may want to use to stay up to date on the latest KB articles is to visit the site, choose a product of interest (CF, Studio, JRun, etc.) and perform a search with no keywords. That will provide a list of the latest articles in order, with the most recent first. Since new ones aren't released that often, this is a simple way to keep up on the latest news for each product.

As for security bulletins, Macromedia does an admirable job of keeping us informed of issues (even ones that aren't CF problems specifically but may affect users of related software such as IIS). The bulletins are offered at www.allaire.com/security/, and again you can sign up on the "notification service" to be informed of any bulletins.

RTFM – Read the "Fine" Manuals

I've mentioned the CF documentation several times. It's really amazing how many people either program with or administer CF without ever reading the manuals. It's a testament, perhaps, to how easy both programming and managing a CF Server can seem, but in both cases failing to read the docs can lead to unexpected problems as well as missed opportunities.

For instance, did you know that the Verity K2 Server brings more than just improved performance and scalability for indexing and searching data? It also adds features for indexing and searching XML documents based on stylesheets, as well as features for spidering/crawling through documents to find referred documents, etc. These are covered in Chapters 7 and 8 of the "Advanced CF Admin" manual. But if you never bothered to look at the manuals, you probably wouldn't know about them.

Another surprise may come from learning that both Windows NT and Windows 2000 Professional have built-in limitations that may make them unsuitable for supporting even medium-traffic sites. They both support only 10 concurrent TCP/IP connections. (It's discussed on page 5 of "Installing and Configuring CF Server.")

The good news for administrators is that a copy of the paper documentation is provided with the purchase of the server. For developers, this means that often the docs sit on the shelf of an administrator who may not care to read the manuals. The good news for developers (or anyone interested in obtaining the manuals) is that Macromedia sells the complete set (both administration and programming manuals) for only $50.

Dead trees aren't the only medium for reading the manuals. They're available online within the CF Server at http://<yourserver>/CFDOCS/dochome.htm (unless they weren't installed, as recommended in KB 16258 for production servers). This HTML format is also fully searchable.

If you don't have them installed locally, you can now view them online (and even read/add comments to them) at http://livedocs.allaire.com.

You can also download PDF versions from www.allaire.com/developer/documentation/coldfusion.cfm.

Remember, too, that in CF5, beyond the two administration manuals, there's also a great deal of context-sensitive help within the CF Administrator. Indeed, there's a lot to be found in that Administrator Help that's not documented in either of the administrator manuals (which will surprise anyone who relies solely on those books for information). If you'd like to see those help files outside the Administrator interface, they're at http://<yourserver>/CFIDE/Administrator/help/cfadminhelp.htm.

> **"Windows NT and Windows 2000 have built-in limitations, supporting only 10 concurrent TCP/IP connections"**

There's still more documentation that you may want to know about as a CF administrator. In addition to the "Advanced CF Admin" manual's coverage of Verity K2 Server and Merant database drivers, there are separate manuals for each installed on the CF Server. The former is discussed in KB 21648 and the latter is at cfusion/bin/odbcref.pdf and odbcsupp.pdf.

Other Information Sharing Resources

Finally, you should do more than rely on your own experience and the manuals to get you through the job of administering a CF Server. As has been said, things are frequently changing and there's also lots more to the task than could be covered in this article. As was also alluded to earlier, there are mechanisms for staying in touch with others who share your pain.

There are many mailing lists including CF-Talk and others run from www.houseoffusion.com, our own ***CFDJ*** list run from www.sys-con.com/coldfusion/, CF lists at http://p2p.wrox.com (under Macromedia), as well as lists run by various CF user groups and those available at various CF-oriented e-zines like defusion.com.

Besides mailing lists, many of these sites also run Web-based discussion forums. The granddaddy of them all (in the CF world) is the ColdFusion forums, available at http://forums.allaire.com/coldfusion/. While not to be relied upon as a tech support resource for urgent problems, it's monitored by both Macromedia staff and enthusiastic CF supporters who help each other solve both common and unusual problems.

Conclusion

It's easy to feel alone and perhaps lost when facing the task of managing a CF Server. I hope these numerous resources for continuous learning and notification, as well as the hints in the article, will help you feel more confident about your role. I hope they'll also motivate you to explore the possibilities for maximizing the administration of your CF Server to improve things both for you and the community of developers on your server.

In Part 2, we'll conclude with a review of the remaining two categories of administrative settings and tasks. The main focus will be security: both ColdFusion Basic and Advanced and what they can allow or restrict, as well as other security facets of administering a CF Server. I'm confident that many will be surprised by some of the possibilities.

Till then, take care and good luck reviewing your own administration role. And again, please share your feedback either online or via my e-mail address, carehart@systemanage.com.

About the Author

Charles 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

A Cure for Arachnophobia

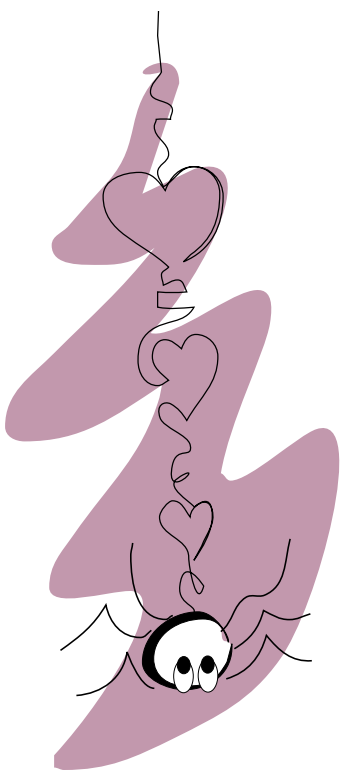
BY **BEN FORTA**

A workable solution

Barely a week goes by without someone asking me about ColdFusion and search engine–friendly URLs. This is one of those topics that ColdFusion developers have been discussing for a long time – I first started a thread on this subject on the Allaire Developer's Forum close to five years ago. As this topic keeps coming up (and because three of you e-mailed me to ask about it this morning), I decided to scrap the column I was writing in favor of an explanation of all this once and for all.

Search Engines and ColdFusion

Here's the problem. Many search engines index sites by spidering them – starting at a known point, indexing that retrieved page, parsing it for embedded URLs, and then retrieving and indexing them too. They keep doing this until they have retrieved and indexed all linked pages in a site.



At least that's how it's supposed to work. Where things get complicated is when dynamic pages are indexed, or rather, when they're not. Some spiders won't index URLs that are dynamic, the theory being that since the content changes all the time there's no point in indexing it – after all, it may not even be the same content on subsequent accesses. So this link would be followed and indexed:

```
<A HREF="catalog.cfm">Catalog</A>
```

but this one may not:

```
<A HREF="catalog.cfm?dept=10">
Catalog</A>
```

You can see the problem: ColdFusion developers (as well as ASP, PHP, Perl, Python, and JSP developers, among others) build dynamic pages. That's why we use ColdFusion – if all content was static we'd use plain old HTML without any server-side processing at all. We use ColdFusion because we want and need dynamic content – content that spiders may choose to ignore. If search engine indexing is a requirement for you, that can pose a real problem.

The Basic Solution

The solution to this problem is actually quite simple. What is it that tips off the spider, telling it that the page is dynamic? It's the query string portion of the URL, starting with the question mark. That question mark separates the URL (the file to be retrieved or the script to be executed) from any passed parameters (usually in name=value pairs).

The solution is to simply not have a query string – no question mark, no values passed after it, and no name=value pairs. Simple, eh? Yep, until you actually need passed parameters. Then what?

Well, look at this URL:

```
http://host/path/catalog.cfm/10
```

What happens when this is processed? catalog.cfm is the file to be executed and anything after the file name is ignored altogether. Even though the Web server and ColdFusion ignore it, that doesn't mean you have to. Using CGI variables (like PATH_INFO) you can access the complete URL – even parts of it that may be ignored by other processes or applications. The following snippet will display the complete path (minus any host and protocol information, although that's available in another CGI variable if needed).

```
<CFOUTPUT>#CGI.PATH_INFO#</CFOUT
PUT>
```

Using the above URL this will display:

```
/path/catalog.cfm/10
```

We now know how to get the passed information. Next you need to remove the path and script name. You could do this by searching for the CFM file, but there's an easier way using yet another CGI variable, this time SCRIPT_NAME, which contains the name of the script being executed (here /path/catalog.cfm):

```
<CFSET
query_string_length=Len(CGI.PATH
_INFO)-Len(CGI.SCRIPT_NAME)>
<CFSET
query_string=Right(CGI.PATH_INFO
, query_string_length)>
```

The first <CFSET> gets the length of PATH_INFO and subtracts

the length of SCRIPT_NAME from it, giving us the length of the section at the end (the data we want). The second <CFSET> uses Right() to extract the desired data, saving it in a variable named query_string. Displaying query_string would result in this output:

```
/10
```

That's how you get the data off the end of the URL.

> *Note:* Depending on the Web server and OS being used you may find different CGI variables or different values in them. You can use <CFDUMP VAR="#CGI#"> to see all the CGI variables available to you (and you'll be able to adapt the code here accordingly).

Making It All Work

Now that the basic concept is clear, let's take this one step further. How can you pass multiple parameters? Well, you could pass three parameters, each separated by a /:

```
http://host/path/catalog.cfm/10
/0/B12
```

But you would have no way of knowing which was which, what their names should be, and they'd always have to be in order (something you typically don't worry about when working with URL parameters).

While the technique is sound, the implementation can be improved by simulating name=value pairs. Look at this example:

```
http://host/path/catalog.cfm/
dept/10/user/0/item/B12
```

There are now six values passed to the URL; each set of two is a name and value pair. Here it's dept=10, user=0, and item=B12. The advantage of this enhancement is that the variable name is known, the order of pairs is irrelevant, and the URL is easy to construct.

All that is required now is a simple way to extract these values and turn them into real URL values (after all, your application code shouldn't have to worry about manipulating this data; as far as it's concerned these are URL parameters plain and simple). This is where ColdFusion's list functions come in handy:

```
<!--- Extract "query_string"
from full path --->
<CFSET
query_string_length=Len(CGI.PAT
H_INFO)-Len(CGI.SCRIPT_NAME)>
<CFSET
query_string=Right(CGI.PATH_
INFO, query_string_length)>

<!--- How many items in
"query_string? --->
<CFSET
items=ListLen(query_string,
"/")>

<!--- Loop through list, pair
of items at a time --->
<CFLOOP FROM="1" TO="#items#"
STEP="2" INDEX="i">
 <!--- Save this URL parameter
--->
 <CFPARAM
NAME="URL.#ListGetAt(query_
string, i, "/")#"

DEFAULT="#ListGetAt(query_
string, i+1, "/")#">

</CFLOOP>
```

The first two <CFSET> statements simply extract the virtual query_string (as we saw earlier). Next we need to know how many items query_string contains. All ColdFusion list functions take an optional delimiter as a parameter, and so ListLen(query_string, "/") returns the number of items delimited by / (in this case, six).

Next, a <CFLOOP> is used to loop through the items, from one to however many there are, stepping two at a time (because every two is a set). Within the loop <CFPARAM> is used to create the variables. The value passed to NAME is the first of the pair (i) and the value passed to DEFAULT is the second (i+1). For the first set of

values (/dept/10) the <CFPARAM> would be:

```
<CFPARAM NAME="URL.dept"
DEFAULT="10">
```

There you have it. By the time the </CFLOOP> is reached, a set of URL parameters would have been created. This code could be placed once at the top of a page, and all other code could refer to URL parameters without knowing they were actually faked. In addition, as <CFPARAM> was used (instead of <CFSET>), you wouldn't overwrite variables if URL parameters actually did exist.

Clean, safe, and spider friendly, too.

Custom Tags to the Rescue

Of course, you wouldn't want all that code in every page – this type of processing is ideally suited for custom tags – thus my <CF_FakeURL> tag. Listing 1 shows the complete code.

Let's take a quick look at the tag. It starts with comments and a description, as all code should. Then a <CFPARAM> is used to define the default delimiter (a slash). Next the virtual query string is extracted using two <CFSET> tags (as we did earlier).

For this code to work there must always be an even number of elements (there are two for every parameter, one for name and one for value). The next <CFIF> statement uses the MOD operator to make sure the number of items in the list is a multiple of two – if this is not the case the list is not processed.

Within the <CFLOOP> the two values (current and next) are extracted, and then <CFPARAM> creates the URL variable.

Now all you need to do is call <CF_FakeURL> in your page and any faked URL parameters will be available for you to use. When you create your URLs just change this:

```
file.cfm?LName=Ben&FName=Forta
```

to this:

```
file.cfm/LName/Ben/FName/Forta
```

Simple as that.

Summary

Arachnophobia is the irrational fear of spiders – a fear that many ColdFusion developers seem to be suffering from. As you can see, with a little ingenuity and some good old CFML, there's a very workable solution to help you overcome this fear. Enjoy!

@ BEN@FORTA.COM

ABOUT THE AUTHOR

Ben Forta is Macromedia's senior product evangelist and the author of numerous books including the recently published ColdFusion 5 Web Application Construction Kit *and its sequel,* Advanced ColdFusion 5 Development. *For more information on Ben's books visit www.forta.com.*

Listing 1

```
<!---
<CF_FakeURL>

 Creates URL parameters from values passed as part of the
path (as opposed to in the query_string). This technique
helps ensure that search engines and spiders index pages
that might have been ignored (as some ignore any dynamic
URLs).
 Using this tag, a URL like this:

foo.cfm/FName/Ben/LName/Forta

will be processed and two URL variables will be created:
URL.FName=Ben and URL.LName=Forta.

 To use, simply place <CF_FakeURL> anywhere in your page
(before the first URL variable is needed). By default / is
used as the delimiter; an alternate delimiter may be speci-
fied in the DELIMITER attribute.

Ben Forta - ben@forta.com
12/1/2001
--->

<!--- Delimiter, defaults to / --->
<CFPARAM NAME="ATTRIBUTES.delimiter" DEFAULT="/">

<!--- Extract "query_string" from full path --->
<CFSET query_string_length=Len(CGI.PATH_INFO)-
Len(CGI.SCRIPT_NAME)>
<CFSET query_string=Right(CGI.PATH_INFO,
query_string_length)>

<!--- How many items in "query_string? --->
<CFSET items=ListLen(query_string, ATTRIBUTES.delimiter)>

<!--- Must be an even number, if odd, bail --->
<CFIF items MOD 2 IS 0>

 <!--- Loop through list, pair of items at a time --->
 <CFLOOP FROM="1" TO="#items#" STEP="2" INDEX="i">

  <!--- Get each pair, first is name, second is value --->
  <CFSET i1=ListGetAt(query_string, i, ATTRIBUTES.delim-
iter)>
  <CFSET i2=ListGetAt(query_string, i+1, ATTRIBUTES.delim-
iter)>

  <!--- Save this URL parameter --->
  <CFPARAM NAME="URL.#i1#" DEFAULT="#i2#">

 </CFLOOP>

</CFIF>
```

CODE LISTING

The code listing for this article is also located at

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

Far too often we listen to the naysayers who tell us that something can't be done and give poorly founded reasons as to why our troubles persist. The ColdFusion Application Server is no exception to their folly. If you ask people for the drawbacks of ColdFusion, most will reply "speed" or "stability." Let me be the first to tell you that it does not have to be that way.

ColdFusion Feature

Managing the Cold

By Tim Nettleton

The system described in this article was built to change the way we think about our code and applications. COSMOS was designed to change our perceptions of the ColdFusion Application Server and to enhance the ColdFusion experience.

If you have ever looked in the /cfusion/log/ directory you've probably seen one or more of the many ColdFusion-generated error/information logs. These text files can easily grow to hundreds of MB and contain the best indicators of "what happened." As with any other service or application, a regular review of system logs should be a part of normal administration. Unfortunately, because of their large size and the fact that the data is segmented into so many logs, it's difficult to get a complete picture of performance, problems, and failure.

Developers who work on a dedicated server can use the ColdFusion Administrator to view these logs. This can be accomplished by clicking on "Log Files" and then downloading the entire log via a browser. Unfortunately, this is usually not possible given the size of most logs and the remote connection speed.

For shared developers, the critical information is unavailable due to the nature of the shared environment and security. In most cases, developers know only what a site user tells them or what they trap using CFTRY/CFCATCH and CFERROR. Even with these mechanisms in place, the larger picture is unavailable and the majority of performance issues go unnoticed and unattended.

The above issues hinder administrators and developers alike. The result is:

- No true time or site correlation for ColdFusion Application Server events.
- Time is wasted attempting to data mine text.
- Site administrators, developers, and business owners don't know there is a problem.
- A negative stigma is created based on a lack of timely and organized information.

To be successful, a solution must have several characteristics:

- Run autonomously, centrally, and constantly
- Contain error lookup with "clean code" examples
- Return all logs and bounced e-mails
- Bring symmetry to the generated data through trending, aggregation, and normalization
- Be fast without affecting performance on the managed server

The solution is COSMOS. Written mainly with ColdFusion, it's an integration of ASP, DOS, Perl, ADSI, and CallXML. It's a remote management platform that leverages the file system, registry, metabase, service controls, and performance counters. Currently, COSMOS contains over 16-million server events aggregated into an MS-SQL database. Captured within a maximum of 40 seconds, these events include all of the following:

- Application errors
- CF Application Server stop/starts
- Hung threads
- Long-running templates
- Missing templates
- Scheduled task results
- Undeliverable e-mails
- Mail sent

How does this affect you? By returning timely, accurate, and relevant information, a developer can see immediately where improvement is needed. At Hostcentric, it's now possible to access several views, graphs, and aggregations that provide a new perspective on the ColdFusion experience.

There are over 20 COSMOS reports available to a dedicated client, most of which are also available for shared customers. The following is a list of some reports with a brief description of how they impact the development and maintenance cycle.

Information Listings

There are several listings available, each with similar characteristics. A listing allows the user to select the maximum number of records to view per screen and how far back to examine data. It also allows the user to progress backward from that point to review previous messages. The majority of listings allow filtering to a single IIS root. They also provide direct access to the complete original error and a corrected code context lookup.

General Application Error Listing

Application errors are the best view into the progress and developmental completeness of a site (see Figure 1). A well-coded site generates no application errors. This listing provides a top-down view of the most recent application errors for all IIS roots. By clicking on the error message on the right, a popup window displays the error message as displayed to a site visitor.

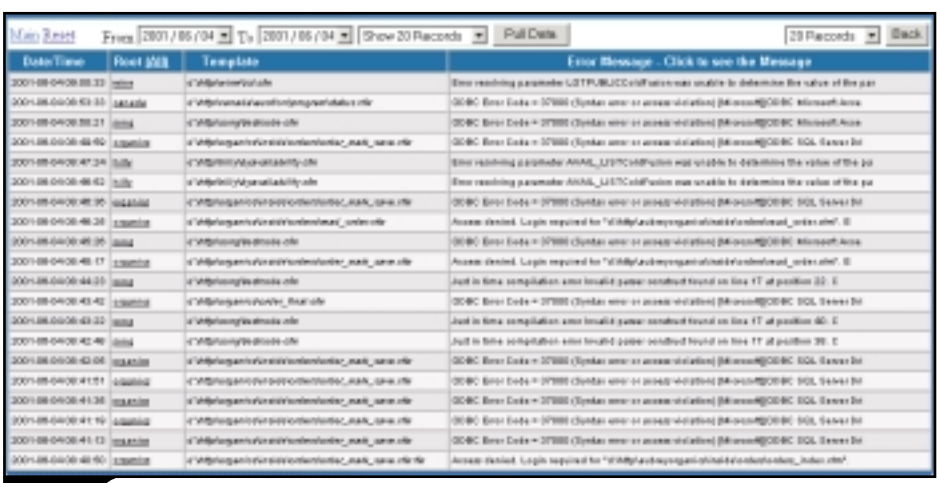


FIGURE 1: General application error listing

> **"How does this affect you? By returning timely, accurate, and relevant information, a developer can see immediately where improvement is needed"**

General Missing Template

This applies to all .cfm templates requested by the Web server, but not found. In most cases, the developer doesn't even know that people are getting "404 File Not Found" messages. If a search engine indexes your site or a user bookmarks a page, a change in the site causes missed business. The solution is to use the default missing template handler in ColdFusion Administrator or to add a CFERROR TYPE="REQUEST" in your site's Application.cfm.

Long-Running Template Listing

This applies to the processing time for pages that take longer than expected. The determination of how long is too long is configured in the logging/settings section of ColdFusion Administrator. A typical setting is 45 seconds, though anything taking that long would most likely be canceled or ignored by the calling client. In addition, a script running for 45 seconds could help identify a performance bottleneck for the application server. By default, CF Administrator doesn't enable this counter. Over time, a development team should ratchet this value as low as possible to get the best diagnostics.

Undeliverable CFMAIL Listing

When ColdFusion is unable to deliver a message to the server specified in a CFMAIL script, the original template is renamed and filed in the /cfusion/mail/undeliver/ directory. An error message is also written to the Mail.log or Error.log describing the problem that prevents proper delivery. This listing binds those two pieces of information together.

The following popup allows an administrator to correct and resend the message from the original server. This function is indispensable for any business that relies on CFMAIL to reliably carry e-mail, and can't accept undelivered messages.

Hung Thread Listing

This is probably the greatest indicator of a performance and stability problem. Hung threads are ColdFusion's method of alerting us that it was unable to completely process the requested template. This is usually the result of code or database issues. CF4.x and above has an option in the Administrator to have CF "restart at *x* unresponsive requests."

When the hung thread count matches the defined threshold, ColdFusion reaches a critical point and will stop/restart itself to avoid excessive downtime. Constant examination of hung threads is necessary to avoid application server failure. At the end of this article I've included three links that help to define more fully the causes of hung threads.

> "As always, COSMOS can determine if your task started, succeeded, or failed based on the logs"

Scheduled Task Listing

Most scheduled tasks run completely unnoticed until someone realizes that a critical function has not processed in days. This listing is not much to look at but, under the hood, a huge modification and improvement has been created for the executive service.

As always, COSMOS can determine if your task started, succeeded, or failed based on the logs. Furthermore, COSMOS will allow you to define a target string in the page HTML and record the generated content from the target URL to the database. If a scheduled task does not return the defined string, an e-mail containing the content and diagnostics can be generated at the time of failure. In addition, the actual HTTP response (CFHTTP.FILECONTENT) is zipped and written to the database.

Aggregation and Stratification

More commonly called a GROUPING, the next series of graphs were created to help identify the greatest problems quickly. By examining the data based on time, date, and IIS root, we can gather a greater understanding of where faults exist.

Application Log Stratification by IIS Root

Over a selectable time span, this graph allows you to see which sites are having the greatest incidence of errors (see Figure 2). By clicking on the blue horizontal bar on the right, you're driven back to the general application error listing but with an additional sort parameter that isolates errors created by the target root.

Time/Error Graph

Especially useful in determining if your day is getting better or worse, this graph breaks down the server errors by 10 minute increments over a selectable date span. This is often used to diagnose a recurring failure point over a multiple day or week period.

Application Errors Stratified by Date

Similar to the previous idea, this graph groups the number of errors by the date that they occurred (see Figure 3). This helps to identify programming trends and can easily indicate a "bad day" for an application. By clicking on the blue bar, your browser is taken to the application log stratification by IIS root. Clicking on the "Time Graph" button brings you to the next graph.

Long-Running Template Aggregation by IIS Root

Similar to the previous root aggregations, this has several prominent exceptions. Because a long-running page has a value associated with the processing time, I've included a column for the sum and average values. Using this display, it's possible to extract the templates most often run beyond acceptable limits, thus demanding the greatest processing time. This affects performance, though not necessarily a failure, and is a fantastic indicator of templates that need to be addressed before they become a stability issue.

Hung Thread Aggregation by IIS Root

This graph will often tell which application is responsible for killing the server. Over a selectable data span you can easily see which sites are causing CF to lose processing threads and tie up resources. The blue horizontal bar links back to the hung thread listing for a given root.

The Big and the Bad

With the thousands of errors, tasks, and events returned each hour, it's easy to become overwhelmed. In addition, not all errors have the same weight on the application server or urgency to a business owner. To resolve this problem a system of alerts and probes runs in the background. Operating at several intervals, the most relevant problems are quickly pulled out of the pool and matched with a type and severity. Once a probe identifies an event or error candidate, the alert has an option of paging, e-mailing, or calling (using CallXML) the administrator.

A Final Look

When did your application server last crash and why?

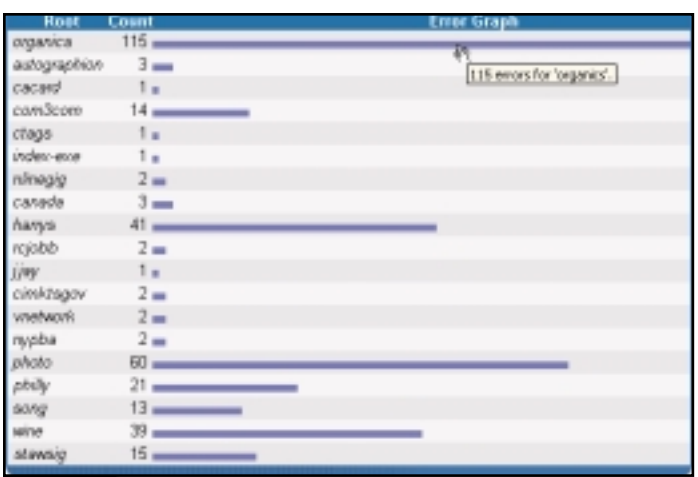


FIGURE 2: Application log stratification by IIS root

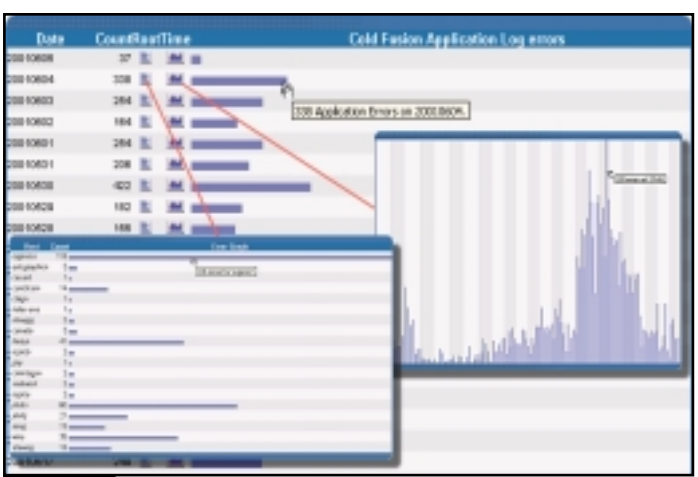


FIGURE 3: Application errors stratified by date

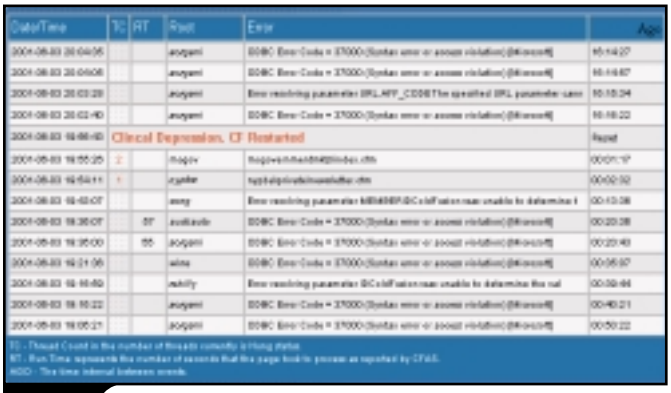


FIGURE 4: Event chronology

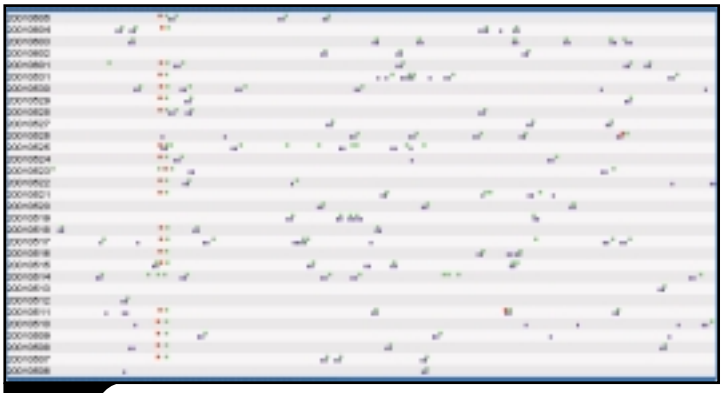


FIGURE 5: Spectral analysis

- ***Event Chronology:*** As the first view that brings together data from multiple sources (see Figure 4), it provides a chronological view of all application errors, hung threads, long-running templates, and application server failures. The graph threads events are based on time in order to provide a trace leading up to a failure.
- ***Spectral Analysis:*** This graph is unique because it rapidly identifies problems that would otherwise slip under the wire (see Figure 5). The three colors representing CF stops (red), starts (green), and hung threads (purple) are graphed relative to a 24-hour time line. By viewing all hung threads that led to server failures, a complete understanding of the root performance issues is garnered.

Summary

Tonight at 1 a.m. your database is going to run out of space and begin throwing application errors. Maybe your mail server stops relaying your order confirmations. There are a thousand permutations to a preventable and containable failure. Will your customers be the first to let you know?

In the end, owners and developers, shared and dedicated, all have the same concerns: stability and performance. Using this system makes that realization no more than a few seconds away.

Related Articles

- http://allaire.com/Handlers/index.cfm?ID=8627&Method=Full
- http://allaire.com/Handlers/index.cfm?ID=2497&Method=Full
- http://allaire.com/Handlers/index.cfm?ID=1505&Method=Full
- http://allaire.com/Handlers/index.cfm?ID=1540&Method=Full
- http://support.microsoft.com/support/kb/articles/q174/4/96.asp

About the Author

Tim Nettleton is a senior engineer at Hostcentric's Orlando division. He has worked with ColdFusion for four years.

@ TIMOTHY.NETTLETON@HOSTCENTRIC.COM

A Fusebox Tutorial

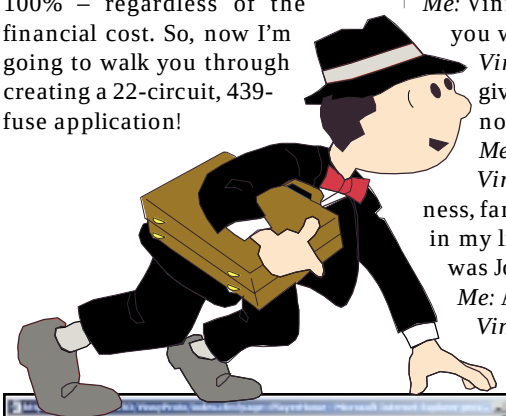
Meet my client Vinny

BY HAL HELMS

For some time, Macromedia has outlined its vision of ColdFusion interoperating seamlessly with Java.

In the November issue of ***CFDJ*** (Vol. 3, issue 11) John Quarto-vonTivadar and I gave an overview of Fusebox 3. Now, over the next two months I want to walk through building a Fusebox application.

The size of a sample application is problematic – too large and you run out of magazine space and user patience; too small and the example may seem trivial and the need for any methodology unclear. However, we writers get paid enormous sums to make tough decisions like this, and when we do our editors back us 100% – regardless of the financial cost. So, now I'm going to walk you through creating a 22-circuit, 439-fuse application!



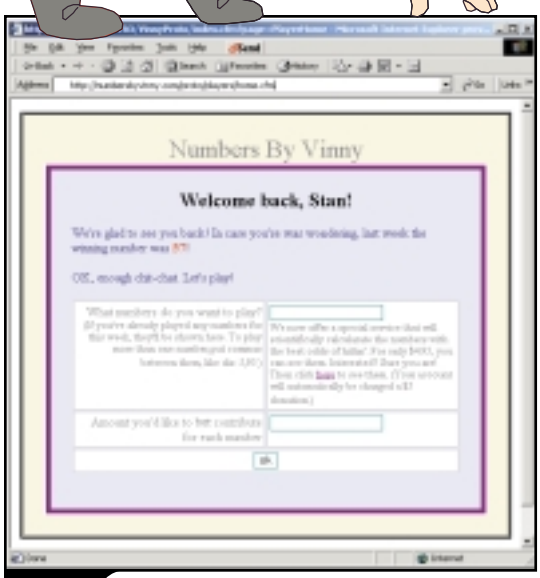


FIGURE 1: Player's home page

[*Editor's note:* Oh, no you're not. I'll give you two circuits, five fuseactions, and a few fuses – that's it.]

After further consideration, I've decided that the readers would be better served by a somewhat smaller application, something like…two circuits and a half-dozen fuseactions or so.

The application we'll build is one I've borrowed from my "FastTrack to Fusebox 3.0" class. First, let me introduce you to Vinny, the client for this application.

Vinny: Yo.

Me: Vinny, you have an application you want built?

Vinny: Sheesh, every class you give I gotta show up for and now I'm in a magazine?

Me: You're getting famous, Mr. V!

Vinny: Yeah, well in my business, fame ain't so good. The last guy in my line of work who got famous was John Gotti and he's in jail.

Me: Ah, yes, well…

Vinny: Well, nothing. Okay, here's what I need. I got this social club with members who like to play a little game in which they pick numbers, and then every Sunday night we have this little drawing to see if anybody picked the same number I picked.

Me: Sounds like a lottery.

Vinny: No. A lottery is illegal and involves gambling. What we're doing is just playing for the fun of it.

Me: But Vinny, you told me before you have muscle…

Vinny: Our customer service representatives, you mean. Yes, we're a full-service social club and it's true that our fun game does have some costs associated with it, and so we request that if people wish to play our game, they consider making a contribution for each number chosen.

Me: Isn't that functionally the same thing as a bet?

Vinny: I dunno about functional, but it definitely ain't the same thing as far as legal.

Me: Okay, whatever. You want to have players pick numbers and then make "donations," is that about it?

Vinny: Yeah, that's right. And have it e-mail me when a donation is made.

Me: Got it. Thanks, Vinny.

> **"I like short, well-defined fuses, as this better accommodates the maintainability and reusability of the code"**

In the class we work through a wireframe and prototype for the application, but in this article I'll start from the point of having an approved (or "frozen") prototype. To keep the size manageable, we'll just deal with the page that lets the user make a bet…

Vinny: A donation
Me: A donation…and then thanks the user for that donation.
Vinny: Hey, what about the full application with CSRs and Admin and nested layouts? I thought you said you was gonna do the whole thing.
Me: Well, I decided it was in the best interests of my readers…
Vinny: Ha. A little pressure and you fold like a cheap suit.

Figure 1 shows the first prototype page, the player's home page. The first step in creating a Fusebox 3 application is to identify the *exit points* of a fuse and any dynamic content on the page. Exit points are ways in which the user (or the system) can exit the current fuse and return to the fusebox. On this page there are two exit points. One is the link in the sentence "Then click here to see them." The other is the "OK" button that submits the form.

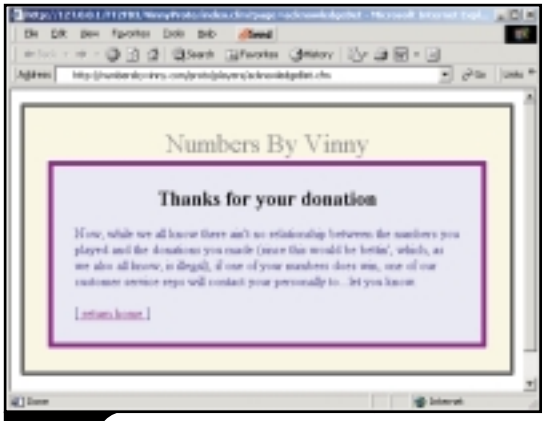


FIGURE 2: Fuseaction "acknowledge bet"

I'll start by giving the exit points names, names that will become fuseactions. I'll call the link "buyProbableWinningNumbers"; the "OK" button I'll name "placeBet". These aren't the only fuseactions I'll need; I'll need a fuseaction that will display the player's home page. I'll call this "home".

Next, I'll decide the circuits that each of these fuseactions fits into. In making this decision, I'm trying to create a coherent model that will allow for code reuse and, just as important, code maintenance. The fuseaction "buyProbableWinningNumbers" I'll place in the circuit "Game". When the player clicks the "OK" button, the fuseaction "placeBet" will be called. This fuseaction belongs to "Game" as well. In addition to these two fuseactions, this circuit will have a fuseaction, "chooseWinningNumber", that will automatically choose a winning number and record this number in a database. I'll schedule this fuseaction to run every Sunday night with ColdFusion's scheduler.

The fuseaction, "home", will be part of a circuit called "Player". This circuit will handle the interface that the player will see. I also need to have a fuseaction called "acknowledge Bet" that will display the page in Figure 2.

There's a link on this page – an exit point – but it leads right back to the Player's fuseaction, "home", which we've already accounted for.

In a full-sized application there would be a good deal more, of course. We haven't accounted for a great many points of functionality, but let's proceed with what we have. Fuseactions are the requests made of an application, but the actual work is done with fuses that are called by fuseactions.

By convention, fuses are given a prefix to indicate the kind of work they do. Fuses that present information on screen are prefixed with "dsp_". Fuses that do work without any display involved have an "act_" prefix. When fuses interact with databases, they have a "qry_" prefix. Fuses that exclusively return to the fusebox with another fuseaction have a "url_" prefix.

Once circuits and fuseactions are identified, fuses must be assigned to fuseactions. I'll begin creating a schematic using a visual outlining tool called Visual Mind available at www.visual-mind.com.

We've made excellent progress. We've converted the prototype and user requirements into a Fusebox schematic (see Figure 3). Now it's time to write Fusedocs for the fuses. Fusedocs allow us to document what each fuse's responsibilities are.

SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645
Telephone: 201 802-3000 **Fax:** 201 782-9600

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

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
online editor
Lin Goetz lin@sys-con.com

production
vice president, 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

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 Pesnick michael@sys-con.com
show assistant
Niki Panagopoulos niki@sys-con.com
registration assistant
Jaclyn Redmond jaclyn@sys-con.com

customer relations/JDJ store
manager, customer relations /JDJ store
Anthony D. Spitzer tony@sys-con.com
customer service liason
Patti Del Vecchio patti@sys-con.com

web services
web designers
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@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

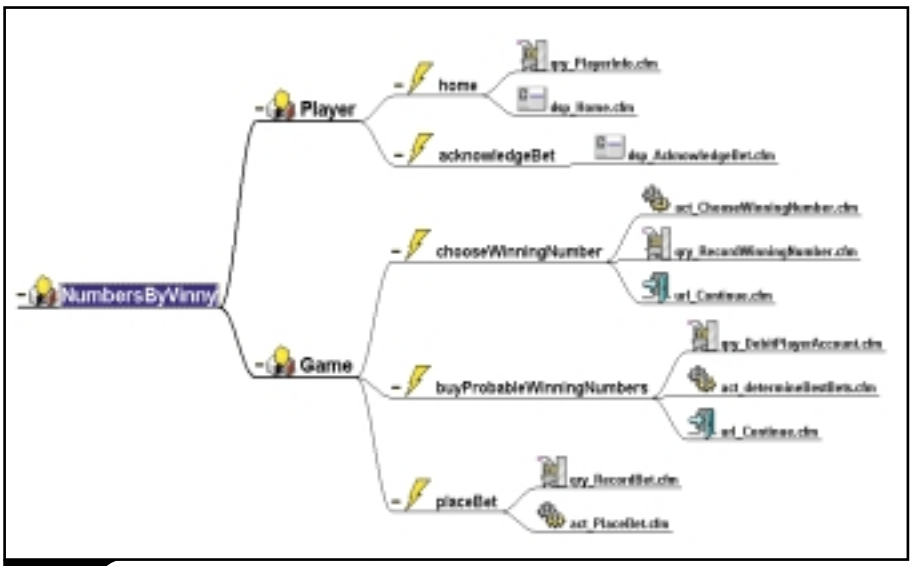


FIGURE 3: Fusebox schematic

When added to the prototype pages (where applicable), these "fusestubs" make the actual coding of an application simple, straightforward, and easy. Let's look at the Fusedoc for one fuse, "dsp_Home.cfm" (see Listing 1).

I've written about Fusedocs before in previous issues of ***CFDJ***. They're XML-based and there's a DTD available for them at www.halhelms.com. The main idea of Fusedocs is that they should provide the information a coder needs to write a fuse without the coder becoming immersed in either the application or any database that might be involved.

Once I have all the Fusedocs written, I'll write the code for these fuses. I like short, well-defined fuses, as this better accommodates the maintainability and reusability of the code. If a fuseaction involves a number of separate actions, I'll call multiple fuses to perform them.

For each fuse I write a test harness. While these can be quite involved, at their simplest they define and set the variables needed by each individual fuse and then call the fuse(s) needed to fulfill the fuseaction. For example, a test harness for the fuse "dsp_Home.cfm" might look like Listing 2.

I'm using a custom tag I wrote about in a previous issue of ***CFDJ*** (Vol. 3, issue 7), QuerySim.cfm, to create a record set without actually hooking into a database. With that taken care of, I can include the fuse itself and see if it works. While this unit testing won't tell me if the entire application works – that's something beyond the scope of an individual fuse – it will give me confidence that all simple, irritating bugs have been exterminated.

When combined with more robust bounds checking, test harnesses can test to see what happens when something unusual occurs. For example, in shopping cart what if a user types "two" instead of "2"? What if they decide they want to order 1.5 books? Or scarier, what if they order -2 books: Do we end up owing them money? Test harnesses can resolve those issues so that when integration begins, we're dealing only with integration issues.

But that will have to wait till next month when we set up the Fusebox 3.0 environment and integrate individual fuses.

Vinny: You'se had all better be back here next month. I want my app done.

Me: Gee, Vinny, haven't you ever heard the expression, "You can catch more flies with honey than vinegar"?

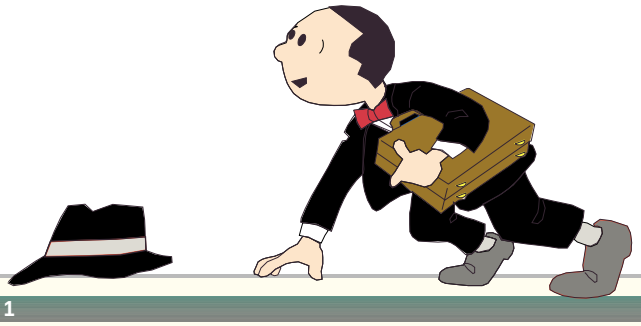
Vinny: Okay. Let me rephrase that: I would be most grateful if your readers would endeavor to participate in next month's exercise so that my application might be finished in an expeditious manner.

Me: Wow, Vinny, I'm impressed. I'm sure everyone will be back next month to help.

Vinny: Yeah. I hope so. Otherwise, it would be a shame if they was to have an accident…

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.

@ HALHELMS@TEAMALLAIRE.COM



Listing 1

```
<!---
<fusedoc fuse="dsp_Home.cfm" language="ColdFusion" version="2.0">
<responsibilities>
 I am the player's home page. In order to run, I must receive the
recordset, PlayerInfo.
</responsibilities>
<properties>
 <history author="Hal Helms" email="hal.helms@teamallaire.com" />
</properties>
<io>
 <in>
  <recordset name="PlayerInfo">
   <string name="playerID" />
   <string name="firstName" />
   <string name="lastName" />
   <list name="numbersInPlay" />
  </recordset>
  <recordset name="WinningNumber">
   <number name="number" precision="integer" />
  </recordset>
 </in>
 <out>
  <list name="chosenNumbers" scope="form" />
  <number name="amountBet" scope="form" />
  <string name="playerID" scope="form" />
 </out>
</io>
--->
```

Listing 2

```
<cf_QuerySim>
PlayerInfo
playerID,firstName,lastName,numbersInPlay
100|Hal|Helms|14,23,91
</cf_QuerySim>
<cf_QuerySim>
WinningNumber
number
57
</cf_QuerySim>
<cfinclude template="dsp_Home.cfm">
```

CODE LISTING

The code listing for this article is also located at

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

ColdFusion Feature

By Anton Prakash

Recently, I was heading a project that involved migrating our company Web site (www.bridgerfunding.com) from NT to Solaris and Linux. This article is based on the experience gathered from the project. If you're running a ColdFusion-based Web site or application on the NT platform and would like to move to UNIX or Linux, reading this article may help make your project a success.

Why Migrate In the First Place?

The decision to migrate to the UNIX platform is a hard one. Before making the decision to migrate our site, I did a great deal of research using the CF Forum to see what people feel about this. I found that quite a few people wanted to migrate from NT but weren't sure of the outcome, so they were still researching the issue. I decided therefore to attempt the migration in a testing environment and then make a decision based on my own findings.

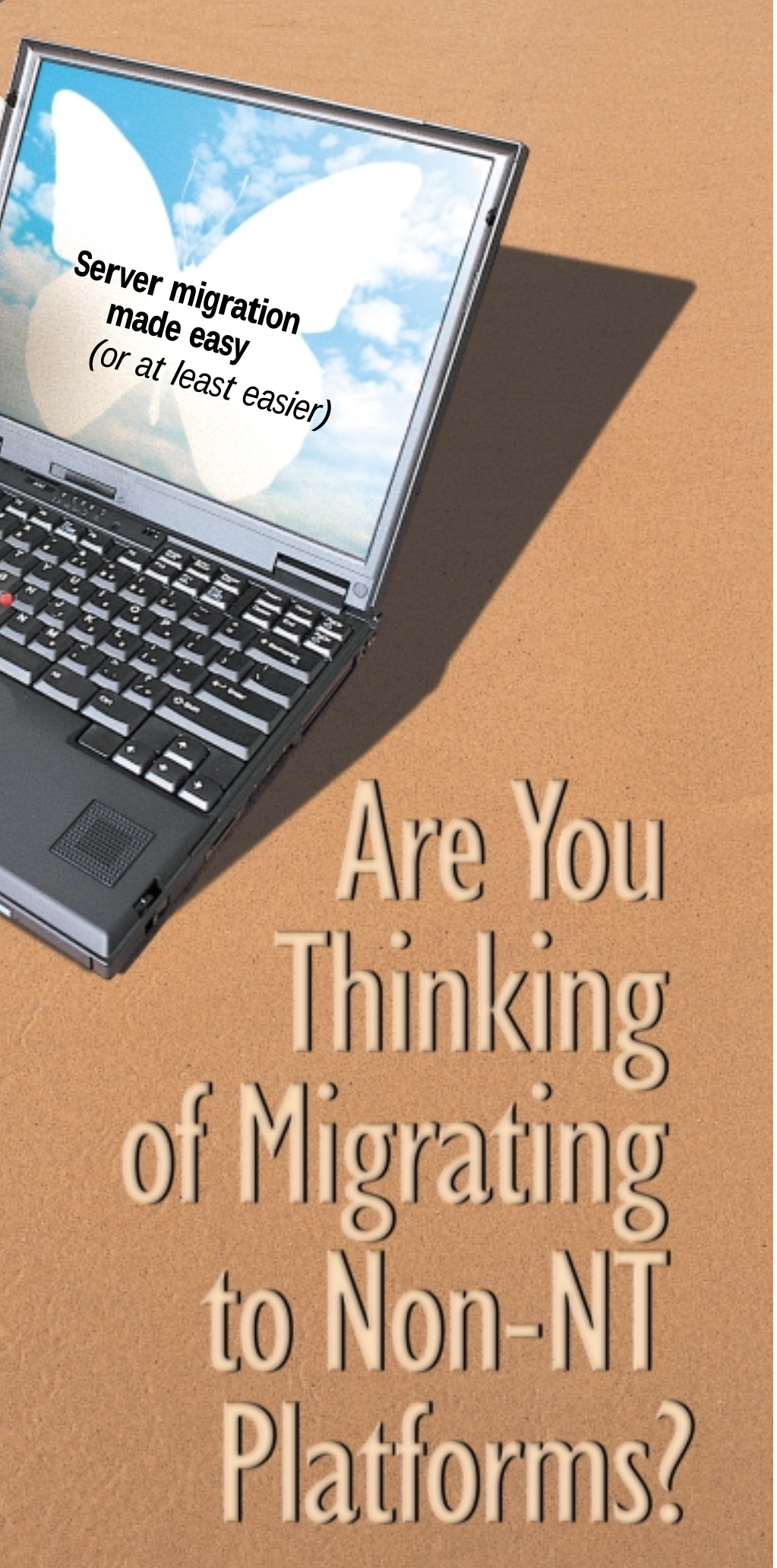
The position at Bridger Commercial Funding was that we were running our site on an NT server with ColdFusion Professional 4.5 as the application server and a SPARC Solaris machine as the database server. While the Solaris server was almost administration-free, the NT server constantly required maintenance. There were occasional crashes too.

Your own reason for migrating could be similar to ours, or totally different. But you must be seriously considering migrating your site to non-NT platforms...or you wouldn't have read this far!

Solaris and Linux?

We have an ULTRA SPARC 5 machine running Solaris 7 as the development server. Our production server is a Sun E220R machine with the same OS. I wanted to have a staging/testing environment where deployed applications would be well tested before being moved to production. Since another SPARC/Solaris server would have been expensive, and because in any case I wanted to give open source technologies a try, I decided to go for a Linux machine.

We now develop in the Solaris, stage it in the Linux before moving to the Solaris machine, and everything hums together pretty nicely.



Are You Thinking of Migrating to Non-NT Platforms?

Having said this, the best approach is to use a Linux server as the migration server and test the migrated site or application in this server, then just copy the migrated site from Linux to Solaris.

Considerations

The following concepts are worth considering before migrating your site/application:

- ***Build the Box Right***

Regardless of the OS you choose, make sure the box is built with all the required components. For Solaris, you need to apply all the required patches given in the release notes. The Solaris command *showrev –p* will give you the list of all patches installed in the system. Give this command:

```
showrev -p | grep <patchname>
```

to see if the required patch is available in the system. For Linux, you need to have the required lib files set up. Also, consider building the *mod_coldfusion.so* for your machine instead of using the downloadable one.

- ***Case Sensitivity***

The non-NT platforms are case sensitive, so when you migrate your code from NT to Solaris/Linux, one of the first major time-consuming modifications that you need to carry out is making the code conform to the case-sensitive requirements.

Unlike Java, ColdFusion code isn't case-sensitive. Listing 1, for example, shows a perfectly valid CF segment. However, any filename referred to inside of the code *is* case sensitive, as shown in Listing 2.

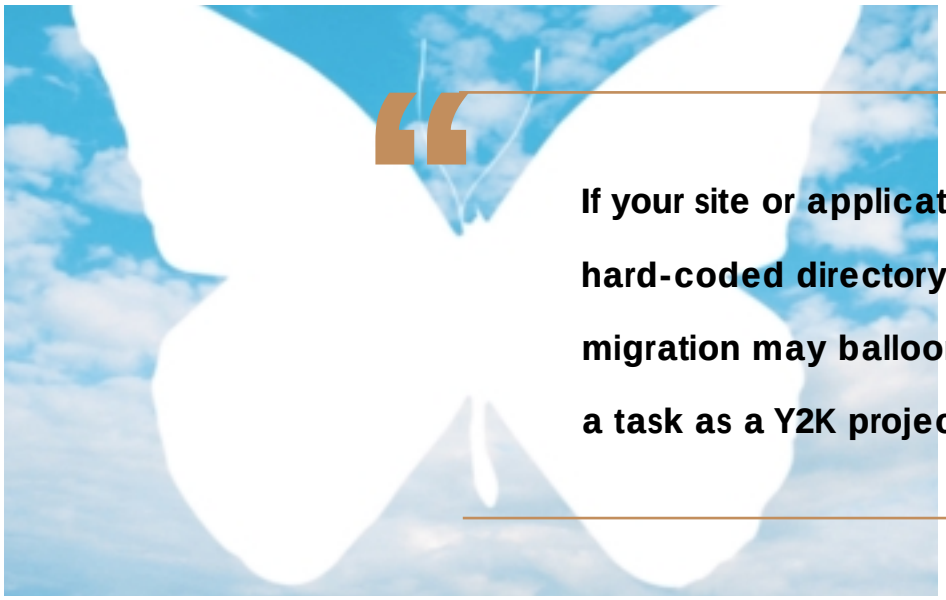
The last thing to mention about case sensitive filenames concerns Application.cfm. In a ColdFusion environment under NT, you can name this application.cfm or AppLiCATion.cfm or any combination of upper and lower case letters – it doesn't matter. But in Solaris/Linux it needs to be Application.cfm with an uppercase A.

- ***Directory Naming Conventions***

If your site or application code contains hard-coded directory names such as

```
<cfset documentDirectory =
"F:\hr\policy\docs\"
```

your migration may balloon to be as huge a task as a Y2K project. You'll need to assemble the troops in your Web department to browse through the code for hard-coded directory references and eliminate them all.



> **If your site or application code contains hard-coded directory names, your migration may balloon to be as huge a task as a Y2K project"**

The best programming practice is to set up environment variables – and in ColdFusion the powerful Application.cfm comes in handy. In UNIX/Solaris, the directory structure is formed with a / (forward slash).

- ***ODBC Connectivity***

If you're running the site with an Access database (hopefully not), then before you attempt to migrate the application, first migrate the database to some enterprise-level database server. If you're not interested in layering out the application and database in different machines, then migrating to MS SQL Server is not an option. For those who don't want to pay an expensive license fee for something like Oracle, my-SQL is a good option.

If you're migrating from ColdFusion Professional and are considering using ColdFusion Enterprise for Solaris/Linux, read this next part carefully. If your code contains ODBC-specific ColdFusion functions such as *CreateODBCDateTime()*, you need to modify this in order to use the Native Connectivity feature; this is one of the biggest differences between Professional and Enterprise. However, in order to make use of the native connectivity, you must use database-specific replacements such as *sysdate* in Oracle (instead of *#CreateODBCDate(Now())#*). You'll run into issues if you need to migrate the database to something else.

- ***Web Server***

We were initially running the site with iPlanet's Web server. While iPlanet provides a great means of administration, we've experienced problems accessing the site from clients in Network Address Translation (NAT) based access to the Internet. (In proxy-based access, there were no performance issues.) Because of this, we had to find an alternative. An obvious choice for us was Apache, a Web server supported in both Solaris and Linux platforms.

- ***Code Placement***

Whatever Web server you use, consider keeping the code in a directory, not directly under the *DOCUMENTROOT* directory of the Web server. This way, you'll be able to maintain a cleaner code directory, while not working on the Web server directory for security and disk space considerations. There are two ways to do this: you can either create mappings as appropriate, or you can use soft links (which is my personal favorite).

- ***Administration***

For those who are more used to a GUI environment, it can be a little daunting in the beginning to realize that you have to actually enter some character-based commands when administering ColdFusion Server. Don't worry: the administration is quite simple. (Refer to Listing 3 for a sample administrator session.)

For those of you unfamiliar with UNIX/Linux platforms, I suggest learning the basics of UNIX commands such as setting environment variables and checking the processes using *ps*. You should also try to get some expertise on using the Vi text editor if you don't have a GUI editor such as Emacs. One good starting point could be the tutorial found at the following URL: http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Editors/ViIntro.html.

If you use CF Studio for development, be sure to change the setting to UNIX, or else every time you edit the code, Studio will insert blank lines between every line. (Options -> Settings -> File Settings ->Format when saving.)

Recommended Eight-Point Action Plan

1. Document the current ColdFusion settings such as Mail Server address and Verity collections. The NT registry can't be migrated to the Solaris/Linux and needs to be done manually.
2. Prepare a Linux/Solaris machine as the migration server. Once the project is over, you can use this as your new development/staging/testing server.
3. Re-create all the CF administration settings in the migration server.
4. Create a zipped archive of the code in the NT server and FTP the file to the Solaris/Linux server to a directory not directly under the DocumentRoot (htdocs). I prefer to create a directory such as /appl/<website> and have soft links under htdocs.
5. Untar the zipped archive in the directory.
6. Change the Application.cfms to be named with an uppercase "A" in all instances.
7. Test the site or application.
8. Fix the code wherever required to adhere to the Linux/Solaris CF code requirements. Most changes will be of a case-sensitivity or ODBC nature.

About the Author

Anton Prakash is the manager of systems development at Bridger Commercial Funding, a commercial real estate financial services company in Mill Valley, CA. He's been working with a variety of UNIX flavors for the past nine years and developing systems in Perl, 4GL, ColdFusion, and Java.

@ ANTON@BRIDGERFUNDING.COM

Listing 1

```
<cfset theSource = "hrDSN">
<cfoutput>#THESOURCE#</cfoutput>
```

Even though the variable here is spelled two different ways, it's OK.

Listing 2

Let's say you have a script called hrEntry.cfm that has a form like this:

```
<form name="hrEntry" method="post" action="HRPROCESS.CFM">
…..
…..
…..
</form>
```

If the filename of the file referred to in the action parameter is hrProcess.cfm, then Object Not Found is going to be returned on submission of this form.

Listing 3

Once logged in to the server, go to the ColdFusion bin :

```
$cd /opt/coldfusion/bin
```

If you want to stop the server,

```
$./stop
```

If you want to start up the server,

```
$./start
```

If you want to check the status of the ColdFusion processes,

```
$./cfstat
```

The results of the above command will give you valuable statistics about the ColdFusion server performace such as Number of Requests Queued and Average Request Time.

```
$ps –aef | grep cold
```

To see if ColdFusion processes are running. If you get any UNIX errors when you try to access the site, issue the following command :

```
$ls –ltr /tmp | grep cf
```

This should output the existence of cfide and cfserver files which are of "socket" type (starting with "s" in the file permission string). If they don't exist, you need to restart ColdFusion.

CODE LISTING

The code listing for this article is also located at

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

Adding New Help Topics to Studio

Increase your productivity

BY CHARLES AREHART

Ever wished you had easy access to Help on non-CF topics such as JavaScript, SQL, and DHTML? Did you know you could easily add such new Help topics to Studio?

In this month's **Journeyman ColdFusion**, I'll show you how easy it is. I'll also point out some existing references you can use to easily extend Studio this way. In just a few minutes, you'll be able to open the Studio Help tab and see JavaScript Help as shown in Figure 1.

Figure 1 shows ColdFusion Studio 4.5, but this discussion also applies to both older and newer versions of CF Studio, as well as Allaire's HomeSite and JRun Studio tools.

The Studio Help Tab

While many may never notice it, CF does have built-in Help that includes several manuals. To get to the Help tab, use Studio's menu commands to select Help>Open Help References Window, or click the "book and question mark" icon at the bottom of the resource window (the left pane where you normally see directories and files). In Figure 1 it's the sixth available tab and the one selected in the screenshot.

By selecting that tab, Studio will show you several available manuals, including all the Macromedia ColdFusion manuals that come with ColdFusion Server (including the installation and administration manuals and the CFML reference), as well as several that come only in Studio and aren't available in print (several Studio customization books).

If you don't have access to the paper versions of these manuals, it's a real lifesaver. If you never got the print version of *Using ColdFusion Studio*, you'll definitely want to read that book here. There's a lot to Studio (including this Help feature) that many never learn about. Things aren't always obvious.

The Help tab even includes HTML and HDML/WML (wireless) references, an added bonus that many never realize. However, your excitement may dwindle when you lament that there's no other Help for other important things such as JavaScript, DHTML, and SQL.

The good news is that you can easily add support for those kinds of Help documents, if you can find them in HTML format. I'll discuss how to extend Studio to support those additional Help files in a moment. First, let's walk through reading a Help document.

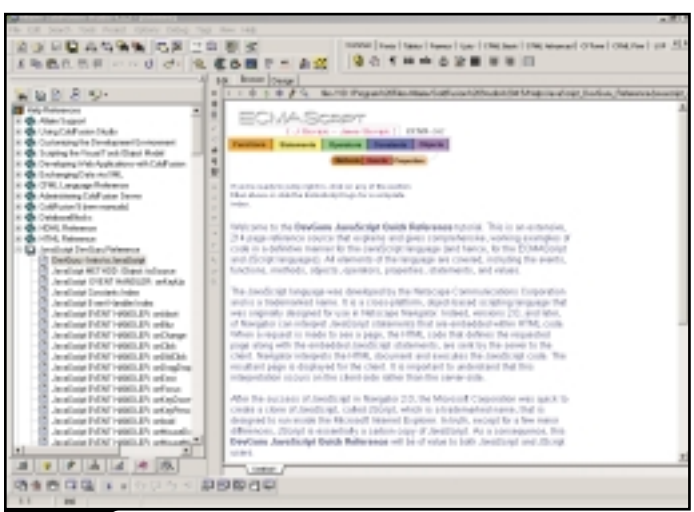


FIGURE 1: Studio display of devGuru JavaScript Help

Reading a Help Topic

When you open the Help References window, Studio shows you a list of all the manuals. When you double-click on one or click the plus sign to the left of it, the manual opens to show all its chapters. Like a normal Windows Explorer-type interface, you continue that process of expanding sections.

When you can finally select (double-click) a specific Help topic, Studio will open that "page" of the document by browsing it inside Studio's editing pane in browse mode. Since the Help documents are just HTML files, they open like any other file you might browse inside of Studio. (Fortunately, if you have trouble browsing CFML documents, there's no similar problem with Help documents.)

Since the documents are HTML and Studio is using a real browser to view them, you can follow any hyperlinks offered in the docs. Indeed, most of the ColdFusion Help documents also offer a set of up, left, and right arrow icons at the top right of each page to help navigate through the document. Use the up arrow to go to the "parent" level of whatever page you're using (goes to a section, chapter, or document list, as appropriate). You can also select another document directly in the left resource pane.

After reading a document you can return to the page you were editing (if any) by pressing the "edit" tab above the edit/browse window, or you can press F12 or select the menu command View>Toggle Edit/Browse to achieve the same effect. (One nice new feature in Studio 5 is that there's now a separate pane to hold the Help document, so the browse pane remains dedicated to its true purpose.)

One last trick you should know about: you don't have to go through this elaborate process of pointing and clicking to open the document

Expanding/Collapsing Trees in Windows by Keystrokes

Here's a trick worth sharing for this interface and indeed for nearly any Windows Explorer interface, such as File Open dialogs. You don't need to use mouse-clicks to expand and collapse levels. If you can place the cursor on the item to be expanded or collapsed, you can use keyboard shortcuts instead. If the cursor is on a level with a plus sign, use the keyboard right-arrow keystroke to expand it. If it's already expanded, use the left-arrow to collapse it. If it's already collapsed, a left arrow will jump to the parent, where another left arrow will now collapse the parent.

list, select the appropriate one, find the section of interest, and open the help topic.

If you're editing a CF page (or HTML, HDML, or WML, to name a few), you can simply press F1 while the cursor is on a given tag (of any of those sorts) and Studio will take you directly to the help topic for that tag in whatever reference it determines is appropriate (CF tags open the CFML reference, HTML tags open the HTML reference). This is very cool, and a lot of people don't realize it's available. As of CF Studio 4.01, it also works for CF functions.

The Help feature is very useful, and once you learn a couple of things it will almost certainly become something you use all the time. (And again, this all applies to CF Studio as well as HomeSite and JRun Studio.)

However, the premise of the article was how to add new Help items. Like so much else in Studio, it's easy – once you know the secret.

Adding New Help References to Studio

I mentioned that the Help documents in Studio are all HTML files. In fact, the concept of a book in the Studio Help interface is simply a directory of files related to one reference. And the set of available "documents" are merely specially named directories all stored under a Help directory wherever CF Studio is installed.

The exact directory name for that will vary by user and by any choices you made on installation, but it might be in something like C:\Program Files\Allaire\ColdFusion Studio\Help (or perhaps it will say ColdFusion Studio 4.5 or ColdFusion Studio 5. Just be aware of which to use to effect changes to the version of Studio you're interested in modifying).

In any case, you'll notice that the Help directory has a series of directory names that look familiar but a little odd (see Figure 2).

(By the way, you may notice that my Windows Explorer shows the complete path to the file I'm exploring, which can be very helpful when you need to cut and paste it. In Windows 2000, at least, you can turn that on from within Windows Explorer with Tools> Folder Options>View>Display the Full Path in the Address Bar and choose "Reset All Folders" to apply it to all folders.)

Notice that the directory names are composed of the same name as that used in Studio, except that when shown in Studio, the underscore between words is automatically converted to a space (CFML_Language_Reference becomes CFML Language Reference).

Inside each directory is a set of files, mostly HTML files. The names of these files may not appear to be meaningful, but when Studio opens the files, it will display them within their given reference manual. (Studio uses the title declared within each file's HTML <title> tag to determine the words to show as the topic description.) We'll discuss this more later.

One last point: any help files (even a single one) must be placed within a subdirectory under the Help directory. Files in the Help directory itself won't appear within Studio's Help tab.

With this knowledge, we can now add new Help files to Studio. If you can locate any set of HTML files organized within a given directory,

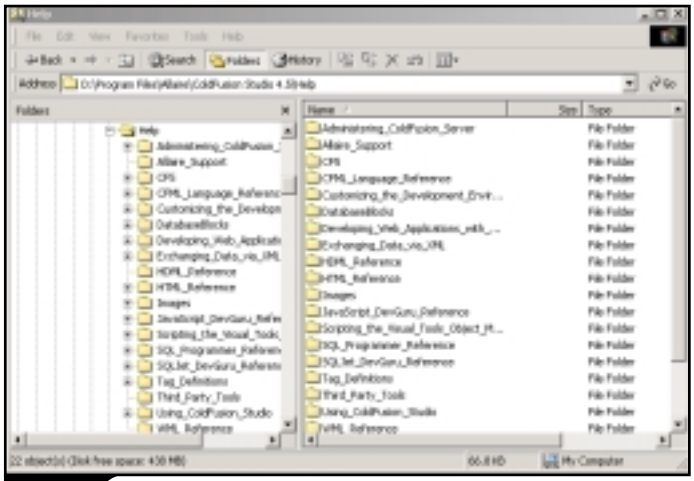


FIGURE 2: The directories within Studio's Help directory

simply place that directory within Studio's Help directory. Rename the folder to use underscores to represent spaces. When you open Studio's Help References window, the new "manual" will appear. (You can right-click on any of the Help topics in the Studio references pane and click "refresh" to refresh the listing without restarting Studio.)

How to Find Existing HTML Help on Various Topics

Where can you locate HTML-based Help files? And on what topics? Well, there may be any number of them in existence. You may have noticed in Figure 2 that I've added JavaScript and SQL (Jet) references (Jet is the MS Access database engine, but much of the SQL is equivalent in other databases).

I found those at www.devguru.com. There they offer these and other HTML references online. They're freely viewable on this Web site, but the cool thing about installing them inside Studio is that they offer a downloadable version for a nominal fee of $5 per "manual." They also offer references in HTML and CSS2; XHTML, XML DOM, and XSLT; WML and WMLScript; ASP, VBSCRIPT and ADO; and WHS (Windows Scripting Host).

I've also added Help whenever I've found it elsewhere, if I thought I might need it inside Studio. For instance, the CD of the book *SQL Programmer's Reference* offered another (and less Access-centric) SQL reference (though it was truncated due to production problems in making the CD and the publisher is not planning to fix it).

Again, with any of these, just copy the directory of Help files to the Studio/Help directory, replace any spaces with underscores in the directory name, and refresh the Help references window.

Some CF-aware software might also install Help automatically, as was the case with Database Blocks, a very inexpensive ColdFusion code-generating tool from CommerceBlocks.com. Macromedia offers a way to add the CF5 Help documents to your Studio 4.5.2 Help reference (as well as tag editors) via a "tag update kit" at www.macromedia.com/software/coldfusion/productinfo/upgrade/.

If you have any suggestions about where to find more, please share them via the feedback capability when viewing this article online at www.sys-con/coldfusion/, or send them to me at carehart@systemanage.com and I'll organize a list of them on my site.

Help File Organization

While it's easy to copy a set of files into the Help directory and rename the directory, a couple of other minor challenges can make the new Help file not as useful as it could be. While I mentioned that the title in any such HTML Help page will become the Help topic name in Studio, not all documents have a title. If there's no title, the name of the file will be shown as the Help topic, and, if the document for some reason has an empty title, Studio will show the topic with no description at all.

Also, if the Help file includes subdirectories, the Studio Help engine will automatically render them as "chapters" within the document, each expandable to show their respective documents.

Another challenge is the order in which each "reference" will appear within the list of all reference manuals. You'll notice that the preinstalled Help references, such as the CFML Language Reference, all appear first in the list. Any that you add will by default appear at the bottom of the list.

> **"The concept of a book in the Studio Help interface is simply a directory of files related to one reference"**

You can override this (and other presentation aspects) by modifying a file in the Help directory called xmltree.xml. See the last section in the first chapter of the Macromedia manual "Using ColdFusion Studio" for more information on this file (remember, the manual is accessible from within Studio). The file already has several settings in it for the existing Help documents, which may help you extend it.

Remember: You don't need to edit this when adding Help files. Added references will appear in alphabetical order at the bottom of the list with other preinstalled manuals not defined in the booktree.xml file. (If you're going to edit the book tree.xml file, it would be wise to save a copy of the file under a new name in case you want to revert back to the original.)

I hope this review of how to add new Help topics in Studio is helpful. It's easy and extensible, and references are available that can increase your productivity as a day-to-day programmer – the goal of the **Journeyman** column. Till next time.

ABOUT THE AUTHOR

Charles 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. Charles is also a columnist for Java Developer's Journal

@ CAREHART@SYSTEMANAGE.COM

Letters to the Editor. . .

Quality Code

I just finished reading Jackson Moore's article "Untrusted Data Sources" (Vol. 3, issue 10) in ***CFDJ***. It was an excellent article and I wished more programmers believed in ensuring quality code and clean data.

By the way, how did you get published in ***CFDJ***? I wouldn't mind contributing sometime.

Steve parks
sparks@houston.rr.com

Editor's Note: All readers are welcome to submit article proposals. For information please read the author guidelines at www.sys-con.com/coldfusion/ and click on "authors."

When to Use ColdFusion

The various articles in ***CFDJ*** that review ColdFusion are extremely confusing to me and almost seem misleading. In an *n*-tier Web environment (i.e., 3+ tiers), *where* does ColdFusion reside and what is its purpose? The articles I've been reading call it "Web application" software; others call it an "application server." In a three-tier environment the following tiers exist:
Web tier: Web server – IIS, Apache, etc.
Middleware tier: application server – WebLogic, etc.
Back-end tier: databases – Oracle, Sybase, legacy, etc.

Where does Cold Fusion reside in this architecture and what purpose does or can it serve if a system already has a WebLogic app server running? ColdFusion sounds like it's *not* an application server in the sense that WebLogic is, and offers a completely different set of Web-based development tools.

It sounds as if Cold Fusion is simply *not* a technology that should be used as the platform for hosting enterprise-based Web applications; instead, ColdFusion components can be integrated into an enterprise-based Web application that's hosted by a true application server like WebLogic. Is this a fair statement?

I am in the initial phases of a large project that will be based on a three-tier architecture. We're definitely looking at using J2EE, EJB, and an application server like WebLogic. However, one of the additions to this system will involve the integration of another ColdFusion-based application (v4.5). This is going to "force" my department into purchasing ColdFusion and has raised several questions regarding whether we need to purchase an application server. Staff personnel are assuming that ColdFusion will provide application server-like features (like WebLogic) and thus negate the need for a WebLogic-like application server. This sounds like a big mistake and a false assumption as well.

If ColdFusion doesn't come with a true J2EE-based application server and offers an alternative way of developing Web-based applications, that's fine. It seems that the alternative solutions provided by ColdFusion may not be the best way to host enterprise Web applications but could help enhance them and coexist with app servers like WebLogic.

Any comments to help clear this up would be greatly appreciated.

Dave Kees
davekees@yahoo.com

I'd ignore the term application server. *As you've noted, it means different things to different people. ColdFusion actually predates that term.*

In your flowchart ColdFusion is the middle tier, or part of it. It's also often considered the Web tier (although it's not a Web server). It's also sometimes the back-end tier, doing all the underlying processing too (generally not a good practice, but sometimes of value). It's not a server like WebLogic or WebSphere in that it doesn't provide some of the infrastructure that those products provide. In fact, in Neo it will optionally run on top of those products to extend them.

ColdFusion can be the middle layer, but it can be other layers too – it depends on what you want to do. Because the product is used in different ways, what users perceive ColdFusion as differs greatly. This is confusing to folks who

see the world as a more structured, layered place; while ColdFusion can play in that space, it was not designed only to play there. Most users use CF any old way – with only a Web server, with COM or EJB bits, and more.

Which is why it is so hard to answer that question.

In the Neo era this will change and become clearer for those who want it to. For those who don't, it'll continue to do it all, as it does now.

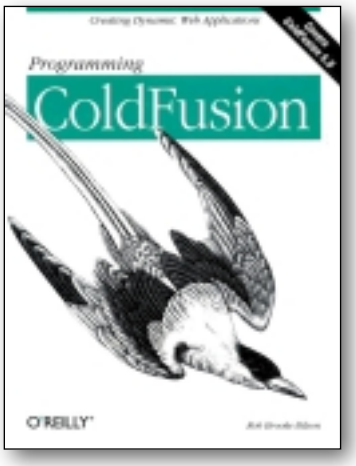
Ben
ben@forta.com

Letters may be edited for grammar and clarity as well as length.

Please e-mail any comments to Robert Diamond robert@sys-con.com

Programming ColdFusion

By Rob Brooks-Bilson
O'Reilly and Associates
953 pages



REVIEWED BY
MARK CYZYK

As the author points out in the introduction: "[Y]ou'll find this book loaded with strategies, hints, tips, and tricks that you can apply to your own projects. I've tried to include all the useful ColdFusion tidbits that I've discovered over the years so that you can benefit from my experience." And Brooks-Bilson, one of the ColdFusion community's most senior developers and a frequent contributor to the Allaire Forums, comes through.

For instance, on page 135 we find a short example of how to include a small chunk of JavaScript to force confirmation before form data is posted – an incredibly simple and useful technique that most CF developers will use over and over again. Then, a few pages further, in an illuminating discussion of variable scopes and thread locking, the author serves up a tasty tidbit pointing out that merely copying a structure to another variable scope won't really do the trick because the structure name is actually only a pointer to the variable. Yet, using the Duplicate() function will force the actual structure, not merely a pointer, to be copied to the new scope. (This is something I immediately stuck in my mental "I-did-not-know-that" file.)

Such snippets and tips are scattered throughout the book and make it a truly useful and enlightening programming resource.

The book, however, is not merely a collection of these tips and tricks. Rather, the book is – as are most entries in the O'Reilly catalog – a well-organized, comprehensive, readable work. It progresses naturally from introductory material to discussions of such advanced topics as using the Verity search engine, regular expressions in CF, creating custom tags, calling external objects, and graphing.

Of the chapters in the book, the two that I found most interesting were the chapters on the Verity engine and on using regular expressions, both advanced topics to be sure.

In the Verity chapter, Brooks-Bilson provides a great example of how to use the Verity engine to create a "Top Ten List" for your Web site. Essentially, he advises reaping search terms and saving them to a back-end table. Then, once this list is at a critical mass, use a simple SQL COUNT function to count the number of times each search string occurs. The result set can then be sorted and looped over, and URLs to the top 10 items can be dynamically generated. In this chapter he also provides a nice example of how to use Verity to index XML files and to map an XML tagset to the built-in CF_TITLE variable.

The regular expressions chapter discusses what they are and the ColdFusion functions that employ them, and provides 10 examples of useful search-and-replace regular expressions. It then provides example code used to build a Web-based regular expression wizard – for a simple forms-based system that will allow you to input a string, choose from a dropdown list of CF functions you want to apply to the string, type in the regular expression you think will do what you want, and test the whole thing with a click of a button. I think this example, as with most of Brooks-Bilson's fine examples, will come in handy in the future.

> **"A truly useful and enlightening programming resource"**

Having reviewed O'Reilly books in the past, I feel compelled to comment on the book qua book, i.e., on the book as a physical, tangible object. Simply stated, O'Reilly has the best typography, page design, and graphical conventions in the industry. This is truly a case where style has a direct impact on substance, or at least on our comprehension and interpretation of the substance of the book. These conventions, as well as the readability of its prose, make the book a pleasure to browse and to use as a reference tool. My only specific complaint in this respect is with the organization of the list of CF functions in the Appendices: the functions here are organized alphabetically, not by functional area. So, for example, a string function such as LTrim() is sandwiched between LSTimeFormat() and Max() instead of being grouped in a section with all other string functions. Organizing this portion of the book by functional area instead of alphabetically would have made a better reference work.

This small criticism aside, however, Rob Brooks-Bilson has written a fine book on ColdFusion programming that will serve the needs of CF programmers, from beginners to advanced.

ABOUT THE AUTHOR
Mark Cyzyk is the Web architect at Johns Hopkins University.

@ MCYZYK@JHU.EDU

wireless EDGE conference&expo

INTERNATIONAL WIRELESS BUSINESS & TECHNOLOGY CONFERENCE & EXPO

CONFERENCE & EXPO

CONFERENCE & EXPO

Plan to Exhibit

Provide the Resources To Implement Wireless Strategy

The conference will motivate and educate. The expo is where attendees will want to turn ideas into reality. Be ready to offer solutions.

Shaping Wireless Strategy for the Enterprise

Santa Clara, CA

May 7-9, 2002

WirelessEdge will provide the depth and breadth of education and product resources to allow companies to shape and implement their wireless strategy. Developers, i-technology professionals and IT/IS management will eagerly attend.

Plan to Attend the 3-DAY Conference

WHO SHOULD ATTEND

Mobile & Wireless Application Professionals who are driving their enterprise's wireless initiatives:

- Program Developers
- Development Managers
- Project Managers
- Project Leaders
- Network Managers
- Senior IT and Business Executives

Conference Tracks

Track One: Development
- WAP
- i-Mode
- Bluetooth / 802.11
- Short Messaging
- Interactive Gaming
- GPS / Location-Based
- Wireless Java
- XML & Wireless Technologies

Track Two: Connectivity
- Smart Cards
- Wireless LANs incl. Bluetooth
- UMTS/3G Networks
- Satellite Broadband

Track Three: Wireless Apps
- Education
- Health Care
- Entertainment
- Transport
- Financial Services
- Supply Chain Management

Track Four: Hardware
- Cell Phones/ WorldPhones
- PDAs
- Headphones/ Keyboards / Peripherals
- Transmitters/ Base Stations
- Tablets

Track Five: Business Futures
- Wireless in Vertical Industries
- The WWWW
- Unwired Management
- From 3W to 4W: Issues and Trends
- "Always-On" Management
- Exploiting the Bandwidth Edge
- Unplugged Valueware
- Wireless Sales & Marketing

FOR INFORMATION CALL

201 802-3069

SPEAKER PROPOSALS INVITED

WWW.SYS-CON.COM

SHAPE YOUR WIRELESS STRATEGY SAVE THE DATES!

EXCLUSIVE SPONSORSHIPS AVAILABLE

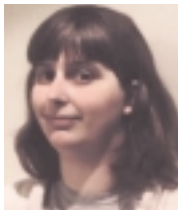
Rise above the noise. Establish your company as a market leader. Deliver your message with the marketing support of

SYS-CON MEDIA

SYS-CON EVENTS

THE LARGEST WEST COAST WIRELESS BUSINESS & TECHNOLOGY CONFERENCE OF THE YEAR!

Core ColdFusion 5



REVIEWED BY
SUSAN
MATTESON

Core ColdFusion 5 (CD-ROM)
By Eben Hewitt
Prentice Hall
969 pages

The other day I overheard someone say that ColdFusion is "hot" right now. ColdFusion *hot*? For years we've all listened as fellow developers hyped each new acronym in Web programming, never knowing the power of ColdFusion. Perhaps now ColdFusion is finally receiving the recognition it deserves as one of the easiest and fastest ways to develop Web applications. One look at a bookstore will tell you that ColdFusion really is hot right now. Since the official release of ColdFusion 5.0, many new books have already come out on CF 5.0, including *Core ColdFusion 5* by Eben Hewitt. Our secret club is no longer secret and everyone wants to join.

Core ColdFusion 5 is a part of Prentice Hall's very successful Core series. Adding ColdFusion to this series shows that it's being considered on a new level, and Eben Hewitt is an excellent author to help bring ColdFusion into the spotlight. The hefty volume, at 969 pages, is an excellent reference and guide for all levels of developers.

In my mind what makes this book a standout is its focus on practical code and useful tips. Hewitt is not just a lecturer or instructor, he's a developer like us who has to deal with the same kinds of problems we do. Useful techniques and solid best practices can be derived from the example code listings that are all through the book.

> **"Hewitt is not just a lecturer or instructor, he's a developer like us who has to deal with the same kinds of problems we do"**

Many ColdFusion books make good reference books. With *Core ColdFusion 5*, the text as well as the code is a good reference source. I often find myself thumbing through the book's code listings to see the elegant way Hewitt illustrated a certain concept in the example application. A major achievement of this book is its refined blend of theory and practice. No sooner do you read about a theoretical concept than you encounter an example that puts it into practice.

Its focus on practicality is best illustrated in the book's extensive series of notes. Many technical books often have "notes" that merely refer the reader to a related area in an appendix. The notes in this book are notes for real developers.

Many of the notes and tips in each chapter point out common mistakes made by both beginners and more experienced developers, as well as mistakes Hewitt admits making in his own projects.

The first four chapters introduce ColdFusion, Internet and server concepts, hardware, and installation. More experienced developers will just want to skim through these sections. I would have preferred it if the book didn't devote so much space to these beginning concepts. I always dislike seeing a ColdFusion book, or any advanced programming book, include chapters on basic Web concepts, but they almost always do. However, if you feel you need a refresher on basics such as HTTP, the conversational style employed here makes them easier to read through. Among the interesting features in the introductory chapters are sections on Linux installation and the merger between Allaire and Macromedia.

The next few chapters are an introduction to the CFML tag base and cover other foundation concepts such as passing data between templates, conditional logic, flow control, functions, data types, and databases. While these topics are found in most ColdFusion books, the casual writing style, many reference tables, and numerous code examples keep these sections fresh and useful for any developer level. Chapter 9, "Designing a Relational Database," also includes a curious but extremely interesting history of the relational model that illustrates, as does the whole book, Eben Hewitt's encyclopedic knowledge.

> **"*Core ColdFusion 5* should be the first stop for new ColdFusion developers and the always on-hand reference for experienced developers"**

The more complex topics covered in the later chapters are the real meat of the book. The flexibility and power of ColdFusion are well demonstrated in the chapters about loops, CF script, scope, e-mail, file management, and custom tags with numerous examples and illustrations of concepts. The chapters covering user-defined functions and custom tags are particularly well done; since Hewitt knows the real-world deadlines we all face as developers, he reminds us to check the Developer's Exchange for already written custom tags.

Advanced developers will enjoy the chapter devoted to the new graphing capabilities of ColdFusion 5.0 as well as the thorough sections on XML, WDDX, and CFHTTP. Nine extensive appendices include a tag reference, CF job resources, common errors, best practices, certification requirements, and additional advanced subjects such as using CF with WML development.

The CD-ROM is one of the most useful I have seen included with a ColdFusion book. It includes 30-day evaluations of ColdFusion 5.0 Enterprise Application Server for Windows, Linux, Solaris, ColdFusion Studio 4.5, HomeSite 4.5, Macromedia Spectra 1.5 Application Framework, and JRun Java Server 3.0.1 for Linux and Windows.

Core ColdFusion 5 should be the first stop for new ColdFusion developers and the always on-hand reference for experienced developers. The final chapter even includes a complete e-commerce application and banner ad server, again illustrating the book's focus on practical code examples, useful tips, and a generous and knowledgeable author. This book will always be open on your desk.

ABOUT THE AUTHOR

Susan Matteson is a self-employed Web developer in Portland, Oregon. She has been involved with Web development for four years as a ColdFusion programmer, Web developer, and graphic designer.

@ SUSAN@GEEKGATEWAY.COM

Protecting Your Web Site Images with ColdFusion

How to apply an image security mechanism

BY STEVEN LEWIS

In a world where information is power, images need protection as much as text documents.

This article describes a way to apply an image security mechanism to protect files in an application that follows a ColdFusion security model. The examples that I give will work in a Windows environment and I'm sure they can be adapted to work in other environments with minimal changes.

The described method isn't to prevent people from downloading images from your site, for example, by right-clicking and hitting *save as* or *file save as*. The method I'm proposing will prevent people from "hotlinking" to your site and using your images without your permission.

Flawed Security Model

Most sites that use ColdFusion to authenticate users are using some type of hidden variable not readily accessible to the user, such as client or session variables that annotate if a user is logged in or not. This is perfect if all your pages are CF, since client and session variables are only visible to your CFM files. However, this brings up the issue of images.

Any time the <IMG> tag is called, the browser requests a file that isn't CF, thereby opening a hole in your security model. Images don't contain the CFM extension, and thus become readily available to anybody who knows the URL. My research indicates that there's no need to develop a complex way to protect files, <CFIMPERSONATE> and <CFCONTENT> will work just fine.

NTFS and the <IMG> Tag

At the moment, the only way for CF developers to display images on their Web pages is to use the <IMG> tag. This is fine if the images you have belong to the public domain. On a Windows-based system the de facto method of protecting files that are Web accessible is to apply NT File Security (NTFS) to the directory that stores the images. So if you were to apply NTFS to your image directory (assuming you're using your database for authentication and authorization), users would have to enter a username and password to view images the first time they make a request for one. If users fail to enter the password, or enter an incorrect one, they'd receive either a broken image link or maybe even a nasty HTTP 401 error saying "Unauthorized: Logon failed due to server configuration."

The only time you should consider using the <IMG> tag without NTFS enabled is when the images are not considered to be sensitive, or when they're available in the public domain – for example, minor graphics or images that aren't proprietary to your company.

Images with NTFS and <CFIMPERSONATE>

On the other hand, you wouldn't want to refer to an image on your company's Web site that gave a detailed synopsis of the corporate network without protecting it. If the directory does contain images that you want to display that are also sensitive , some type of mechanism to get the images is needed.

With the <CFIMPERSONATE> tag we have a valid mechanism to access images that are contained within an NTFS-protected directory. The purpose of this tag is to allow a developer to write a script that will impersonate a user located either within the Advanced Security Model or on the operating system (OS). To access the images that are located within the NTFS-protected directory, we'll need to use an account that's located on the OS.

To use the <CFIMPERSONATE> tag to access a directory that's been protected with NTFS, we need to do the following:

1. Declare a domain in which the user account is located
2. Provide a username and password
3. Enter a type of OS

Entering a type of OS will tell the ColdFusion Server to attempt to access the image as the declared user located in the OS. Once this is done, all commands located between the start and end <CFIMPERSONATE> tag will be performed as the stated user (see Listing 1).

Images and <CFCONTENT>

Another method of securing images is to use the <CFCONTENT> tag, a very versatile tag with many uses beyond those I'm explaining in this article. Some people say that they've encountered problems using <CFCONTENT> with Windows NT Service Pack 6a, but the examples given here will all work fine with Service Pack 6a.

The burning question is: How can <CFCONTENT> be used to display images inside a Web page and still enable you to keep your pages formatted properly? Whenever you

JavaOne
www.java.sun.com/javaone

use <CFCONTENT> you'll have to provide the exact path where the image file is located. Since the exact location of the file is required, this enables you to store all of the images outside of any Web-accessible directories. This in turn provides added security, because your images are now inaccessible to people who are either not physically on the machine or outside of your site. For <CFCONTENT> to be accessed, it has to be inside a CFM page, and inside your CFM files. This way you're applying your security model, thereby protecting your images.

Using <CFCONTENT> to Display the Images

To keep a Web page properly formatted, <CFCONTENT> will need to be called from a separate file (see Listing 2). The trick behind keeping your page aligned properly is to call the CFM page from within the <IMG> tag. See Listing 3 for more details.

```
<IMG
SRC="/someDir/displayImage.cfm?i
mageId=23" width="200"
height="200">
```

The reason for using the <IMG> tag is that the browser will provide the appropriate image formatting based upon the attributes provided within the <IMG> tag. The <IMG> tag is in effect acting like a separate ColdFusion request for a file – and, for all practical purposes, that's just what it is. Whenever an <IMG> tag is called, CF is making a separate request for the image information and placing it inside the <IMG> tag. Since a ColdFusion page is doing the request within the <IMG> tag, your security model is being applied against the user requesting the image file.

Final Recommendations

What I've described here is an alternate method of protecting images that you wish to make accessible via the Web in an <IMG> tag. Although <CFCONTENT> and <CFIMPERSONATE> are both effective means of displaying protected images on the Web, they place an extra burden upon the ColdFusion server. Every time an image is requested with the <CFCONTENT > or <CFIMPERSONATE> tags, it requires the server to do additional processing. If you have a site with a large volume of daily hits, this type of processing can add up. I'd recommend that you use this method only for images that really need to be protected.

ABOUT THE AUTHOR

Steven Lewis is an Allaire certified ColdFusion developer employed by Booz Allen & Hamilton, where he's the technical lead for a collaboration tool development project. He has a BS in information decision systems from San Diego State University.

Acknowledgments

I'd like to thank Angelo Alverez for recommending that I submit this article to ***ColdFusion Developer's Journal***. But most of all I'd like to thank Rowan Kelly for helping me figure out this method of protecting images.

@ LEWIS_STEVEN@BAH.COM

Listing 1: Accessing a directory protected with NTFS

```
<CFIMPERSONATE
 SECURITYCONTEXT="domain"
 USERNAME="Name"
 PASSWORD="Password"
 TYPE= "OS">
 <img src="/NTFSdir/someImage.gif">
</CFIMPERSONATE>
```

Listing 2: Page calling image information from DB

```
<CFQUERY QUERY="getImage" DATASOURCE="DS">
SELECT file, mimeType
FROM images
WHERE id = #url.ID#
</CFQUERY>


<CFSET fileLocatoin = "C:/images">
<CFCONTENT FILE="#fileLocation#/getImage.file"
MIME="getImage.mimeType">
```

Listing 3: Page that displays image

```
<html>
<head>
<title>Show me some images</title>
</head>
<body>
  <table>
    <tr>
      <td>
         <img src="/someDir/showImage.cfm?id=5">
      </td>
      <td>
         Hello World!
      </td>
      <td>
         <img src="/someDir/showImage.cfm?id=10">
      </td>
    </tr>
 </table>
</body>
</html>
```

CODE LISTING

The code listing for this article is also located at

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

CF USER GROUPS

Birmingham, AL
Jeff Bryant
jeff.bryant@mindspring.com

Huntsville, AL
Brian Yagar
brian.yager@redstone.army.mil

Phoenix, AZ
Sean Tierney
sean@pubcrawl.net

Tucson, AZ
Darcey Spears
dmspears@uswest.net

Bay Area, CA
Nathan Dintenfass
nathan@changemedia.com

Southern California
Leon Chalnick
lchalnick@advantasolutions.com

Southern California - Los Angeles
Sid Sheres
sheres@gte.net

Orange County, CA
Peter Amiri
peter@amiri.net

Sacramento, CA
Geoff Lilley
geofflilley@hotmail.com

San Diego, CA
Robert Munn
rob@illuminez.com

Orleans, Ontario
Clovis Paquette
clovispa@comnet.ca

Edmonton, Alberta
Jeff Acker
jeff.acker@topham.ab.ca

Lethbridge, Alberta
Chris Sorensen
chris@csystems.ab.ca

Montreal, Canada
John Kopanas
kopanas@videotron.ca

Ottawa, Canada
James Milks
jamesm@videotron.ca

Toronto, Canada
Kevin Towes
KTOWES@PangaeaNewMedia.ca

Vancouver, Canada
Cameron Siguenza
cameron@evolutionb.com

Colorado Springs, CO
Anthony Starkey
astarkey@mentiscorp.com

Denver, CO
Robi Sen
robi@granularity.com

Rocky Mountain
Mike Miller
mmiller@ewtechnology.com

Hartford, CT
Chris Graves
graves@rapidcf.com

New Haven, CT
Alfred DiMarzio
cfug@hbgraphics.com

Stamford/Westchester, CT/NY
Barry Hyman
bah@abodofoto.com

Delaware
John McKown
john@delaware.net

Gainesville, FL
Taner Aktas
aktas@ufl.edu

Jacksonville, FL
Russ Johnson
russ.johnson@cyberfusionsolutions.com

Orlando, FL
Randy Drisgill
rdrisgill@vshift.com

South Florida
Kevin Langevin
kevin@cfug-sfl.org

Tallahassee, FL
Benjamin Bloodworth
ben@electronet.net

Tampa Bay, FL
Kenneth Beard
kbeard2323@hotmail.com

Atlanta, GA
Adam Churvis
president@prodenhance.com

Atlanta, GA
Seth Land
gacfug@figleaf.com

Columbus, GA
Rob Seebach
rseebach@sfcts.com

Boise, ID
Paul Jenkins
pjenkins@doi.state.id.us

Chicago, IL
Robert Burns
bob-burns@mediaone.net

East Central Illinois
William Steiner
WilliamS@hkusa.com

Quad Cities
Jamie DeVolder
JamieD@4CS.COM

Springfield, IL
Gary Ashbaugh
ashbagd@mail.ioc.state.il.us

Central Indiana
Joshua Kay
josh@dataquix.com

Northern Indiana
Graham Pearson
gpearson@nicfug.org

Cedar Valley CFUG
David Chandler
David@ChandlerResearch.com

Des Moines, Iowa
Kevin Schmidt
schmidt@hungrycow.com

Louisville, KY
Tim Newton
TNewton@MercuryFusion.com

New Orleans, LA
Kathy Hester
kathyhes@yahoo.com

Annapolis, MD
Matt Schuster
mattschuster@erols.com

Baltimore, MD
Bret Peters
bmorecfug@figleaf.com

Broadneck High School CFUG
Rick Blaha
IRiCkYl@aol.com

Maryland
Michael Smith
Michael@teratech.com

Southern Maryland
Russ Mahan
rmahan@erols.com

New England Develpor's Network
Steve Casco
steve@ateaze.com

Detroit, MI
Matt McDonald
matt@stoneage.com

Mid-Michigan Group
Rick Mason
Rick@SeedChoices.com

Twin Cities, MN
Dan Chick
dan@danchick.com

Kansas City, MO
Bryan Laplante
directors@kcfusion.org

SoMoCFUG
Jeremiah Andrick
jandrick@theinteriorlife.com

St. Louis Metro Area
Dan Anderson
dan@psiwebstudio.com

Helena, MT
Tom Marino
tmarino@falcon.com

Omaha, NE
Mark Kruger
Mkruger@CFWebtools.com

Las Vegas, NV
Dave Byers
pegarm@lvcm.com

Concord, NH
Neil Giarratana
NGiarratana@ininet.com

Central New Jersey
Hien Nguyen
HNguyen7@prius.jnj.com

Northern New Jersey
Cindy Pomarlen
cjp@neotech-associates.com

South Jersey
Mark Stewart
m-stewart@home.com

Albuquerque, NM
Christopher Jungmann
jungmann@swcp.com

Albany, NY
Thomas McKeon
tmckeon@newkirk.com

Long Island, NY
Christopher Collins
chrisc@sitespan.com

New York, NY
Michael Dinowitz
mdinowit@houseoffusion.com

Rochester, NY
Dave Horan
dhoran@fusionproductions.com

Syracuse, NY
Laurie Ferger
lferger@twcny.rr.com

Charlotte, NC
Dave Crawford
dcrawford@charlotte-cfug.org

Greenville, NC
Steven Forehand
forehands@mail.ecu.edu

Raleigh, NC
Cheryl Elia
ccfug.mgr@listserv.unc.edu

CF USER GROUPS

Mid-Ohio Valley
Dave Hannum
hannum@ohio.edu

Cleveland, OH
Matt Newman
mnewman@newchanneltech.com

Columbus, OH
Angelo McComis
Angelo@CompuServe.com

Oklahoma City, OK
Glen Collymore
Glen-Collymore@ouhsc.edu

Tulsa, OK
Casey Bourbonnais
casey@cbfenterprises.com

Portland, OR
Jeff Cram
jcram@isitedesign.com

Roseburg, OR
Sharon Bland
sbland@douglasesd.k12.or.us

Harrisburg, PA
Neil Ross
info@codesweeper.com

Philadelphia, PA
Chris Swansen
cswansen@etechsolutions.com

Pittsburgh, PA
Kyle Jenkins
kyle@vaporbiz.com

State College, PA
Brian Panulla
brian@elmwoodmedia.com

Greenville, SC
David Goldfield
Dave@goldfield.com

Knoxville, TN
Doug McCaughan
doug@colknox.com

Memphis, TN
Michael Kubicki
mkubicki@univ-solutions.com

Nashville, TN
Chris Mosier
chris.mosier@e-maginative.com

Austin, TX
Mike Dickinson
cranch@texas.net

Dallas, TX
Patrick Steil
pmsteil@imailbox.com

Houston, TX
User Groups
usergroups@allaire.com

San Antonio, TX
Douglas White
doug@dwhite.ws

Utah State College
Jon Nelson
jnelson@nstep.net

Salt Lake City, UT
Mike George
ueberhund@hotmail.com

Provo, UT
Josh Trefethen
josh@exciteworks.com

Vermont
Sarah Moulton
info@mtbytes.com

Hampton Roads, VA
Raymond Camden
morpheus@deathclock.com

Northern Virginia
Bret Peters
novacfug@figleaf.com

Richmond, VA
Robert Baird
bbaird@btgi.net

Figleaf CFUG
Bret Peters
cfug@figleaf.com

Seattle, WA
Shannon Smith
president@cfseattle.org

Walla Walla, WA
Sarah Lewis
sarahlewis@alhena-design.com

Milwaukee, WI
Ron Kurtus
ron@ronkurtus.com

Wyoming
Matt Rodosky
mrodosky@wyoming.com

Nepal, Asia
Sudhir Raj Joshi
sjoshi@iapex.com

Australia
Marc Dimmick
marc@ibaseglobal.com

Canberra, Australia
Kunal Bhatia
kbhatia3@csc.com.au

New South Wales, Australia
Kym Kovan
kymk@microset.com.au

Northern Territory, Australia
Fred Lecul
flecul@paspaley.com.au

Queensland, Australia
Barry Moore
barry_moore@yahoo.com

Southern Australia
Brett Hales
brett.hales@baesystems.com

Tasmania, Australia
Mark Heseltine
HeseltineM@logica.com

Victoria, Australia
Jordan Thomas
jordan@vector7.com.au

Belgium
Denis Wauthy
d.wauthy@switchon.com

Rio de Janeiro, Brazil
Marcello Frutig
frutig@astrolabio.com.br

Sao Paulo, Brazil
Marcantonio Silva
marco@channelmedia.com.br

Central Europe
Sven Slazenger
slazenger@interlake.net

China
Feng Jian Feng
webmaster@cfwindow.com

Finland
Tero Pikala
tero.pikala@kolumbus.fi

France
Fabien Graille
FG500001@exchange.FRANCE.NCR.com

Ramstein Air Base, Germany
Tom Nunamaker
tom@nunamaker.com

Hong Kong
Shashank Tripathi
shashank.dp.tripathi@hk.pwcglobal.com

Hungary
Zolton Sheshler
ozoltan@yahoo.com

India
Zahid Java
zahidjava@yahoo.com

Indonesia
Agus Basuni
agus@digitaldevelopment.com

Ireland
Daryl Fullerton
daryl@thebiznet.net

Dublin, Ireland
Justin MacCarthy
maccarthy@iol.ie

Italy
Andrea Veggiani
andrea@dinamica.it

Okinawa CFUG
Michael Bradford
michael@devapps.com

Korea
HyeHyun Seo
hhseo@otozone.com

Malaysia CFUG
Halmi Yasin
affroman@affroman.com

Netherlands
Ron Pasch
ronnieo@dds.nl

Auckland, New Zealand
Amanda Irvine
amanda.irvine@delta.co.nz

Romania
Andrei Oprea
andrei@oprea.org

Johannesburg, South Africa
Steven Ringo
steven@tutuka.com

Spain
Jorge Pueyo
jpueyo@lander.es

Sweden
Stefan Gudmundsson
stefan@cfug-se.org

Zurich, Switzerland
Martin Buerlimann
bue@sercon.ch

Taiwan
Jack Jair
jack@leetide.net

Bangkok, Thailand
Paul Hastings
paul@tei.or.th

Turkey
Oguz Demirkapi
oguz@cftr.net

Dubai
Maryam Ebrahimi
CFUG@digiba.com

London
Niklas Richardson
niklas@ukcfug.org

Northern England
Stephen Moretti
stephen@cfug.org.uk

Southern UK
Stefan Elliott
cfug@four-runner.com

Subscribe *Now*

for Instant Access to CF Advisor

CFAdvisor.com!

12 months $89
24 months $169

Now on CD!

THE MOST COMPLETE LIBRARY OF EXCLUSIVE *CF ADVISOR* ARTICLES!

3 YEARS
40 ISSUES
200 ARTICLES
ONE CD

ONLY $71.99

Since May 1998, CF Advisor™ has provided the CF community with in-depth information that quickly pays dividends on the small subscription investment. For just $89.00, you get 12 full months of uninterrupted access to CF Advisor™ – including:

- usable code
- skill-building articles
- tips
- news updates
- conference reports
- interviews with industry movers-and-shakers.

You get full access to our sizable archives covering such subjects as:

- custom tags
- functions
- variables
- framesets
- style sheets
- structures
- javascript usage
- debugging techniques
- programming practices

and much, much more...

Subscribing now guarantees that you'll immediately receive access to everything you need for keeping up-to-date with ColdFusion, no matter where you are. All the content is available exclusively over the Net and is regularly updated throughout the month.

THE MOST COMPLETE LIBRARY OF EXCLUSIVE *CF ADVISOR* ARTICLES ON ONE CD!

"The Complete Works"

CD is edited by *ColdFusion Developer's Journal* Editor-in-Chief Robert Diamond and organized into 21 chapters containing more than 200 exclusive *CF Advisor* articles.

Easy-to-navigate HTML format!

E-Commerce	CFBasics	Object-Oriented CF
Interviews	Reviews	WDDX
Custom Tags	Scalability	Upgrading CF
Fusebox	Enterprise CF	Programming Tips
Editorials	Wireless	CF Tips & Techniques
Databases	Verity	ProgrammingTechniques
News	Source Code CF	Forms
CF & Java	Applications	

Order Online and Save 10% or More!

www.JDJSTORE.com

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

SYS-CON MEDIA

135 CHESTNUT RIDGE ROAD
MONTVALE, NJ 07645

WWW.SYS-CON.COM

THE INSIDER INTELLIGENCE YOU NEED...

TO KEEP AHEAD OF THE CURVE

FREE E-Newsletters

SIGN UP TODAY!

Go to www.SYS-CON.com

The most innovative products, new releases, interviews, industry developments, and plenty of solid *i*-technology news can be found in SYS-CON Media's Industry Newsletters. Targeted to meet your professional needs, each e-mail is informative, insightful, and to the point. They're free, and your subscription is just a mouse-click away at www.sys-con.com.

SELECT THE INDUSTRY NEWSLETTERS THAT MATCH YOUR NEEDS!
CHOOSE ONE – OR TRY THEM ALL!

- Java Developer's Journal Industry >Newsletter
- WebLogic Industry >Newsletter
- WebServices Industry >Newsletter
- WebSphere Industry >Newsletter
- XML Journal Industry >Newsletter
- ColdFusion Industry >Newsletter
- wireless Industry >Newsletter
- Linux Newswire Industry >Newsletter

Don't Delay!
Subscribe for FREE!
at www.sys-con.com

Exclusively from the World's Leading *i*-Technology Publisher SYS-CON MEDIA

CFDJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
ABLECOMMERCE	WWW.ABLECOMMERCE.COM	888.801.1333	2
ACTIVE PDF	WWW.ACTIVEPDF.COM	888.389.1653	4
ASSETNOW	WWW.ASSETNOW.COM		31
CFADVISOR.COM	WWW.CFADVISOR.COM		51
CFDYNAMICS	WWW.CFDYNAMICS.COM	800.422.7957	19
CFXHOSTING	WWW.CFXHOSTING.COM	866.CFX-HOST	11
COLD FUSION DEVELOPER'S JOURNAL	WWW.SYS-CON.COM	800.513.7111	41
EMPIRIX	WWW.EMPIRIX.COM/DOUBLE/CFM		9
EVOLUTIONB	WWW.EVOLUTIONB.COM/PARTNERS		55
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM	877.215.HOST	25
INTERMEDIA.NET	WWW.INTERMEDIA.NET	800.379.7729	56
JAVA DEVELOPER'S JOURNAL	WWW.SYS-CON.COM	800.513.7111	23
JDJ STORE	WWW.JDJSTORE.COM	888.303.JAVA	49
MACROMEDIA	WWW.MACROMEDIA.COM/GO/MASTERING	877.460.8679	15
MACROMEDIA	WWW.VUE.COM/MACROMEDIA	877.460.8679	29
MACROMEDIA	WWW.MACROMEDIA.COM/GO/USERGROUPS	877.460.8679	13
MINERVA	WWW.MINERVA.NET	888.667.7231	27
PACIFIC ONLINE	WWW.PACONLINE.NET	877.503.9870	33
PAPERTHIN	WWW.PAPERTHIN.COM	800.940.3087	37
RACKSPACE	WWW.RACKSPACE.COM	888.850.1470	3
SYS-CON MEDIA, INC.	WWW.SYS-CON.COM	800.513.7111	45
WEB SERVICES JOURNAL	WWW.WSJ2.COM	800.513.7111	39, 43
WEBLOGIC DEVELOPERS JOURNAL	WWW.SYS-CON.COM/WEBLOGIC	800.513.7111	44
WIRELESS EDGE CONFERENCE & EXPO	WWW.SYS-CON.COM	201.802.3069	47
XML JOURNAL	WWW.SYS-CON.COM	800.513.7111	17

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.

CFDJnews

Certification Available for Dreamweaver Developers

(San Francisco) – Macromedia, Inc., has announced the availability of Dreamweaver Developer Certification, which validates a user's understanding of Dreamweaver, Web page design, Web authoring, and Web technologies.

Dreamweaver certification is part of the larger Macromedia Certified Professional Program, which demonstrates a proficiency in designing and developing Web sites to employers and clients. In addition to Dreamweaver, exams are available for Macromedia ColdFusion and Macromedia Flash developers.
www.macromedia.com/go/certification/

PageScreamer Product Suite Expanded

(Arlington, VA) – Crunchy Technologies, a provider of secure, accessible e-business and e-government software products and custom applications, has introduced PageScreamer plug-ins for Macromedia's ColdFusion, HomeSite, and JRun Studio to further help organizations develop long-term practices that enhance accessibility for users with disabilities.

These new products join PageScreamer, a tool developed to help the federal government and commercial entities comply with Section 508 of the Rehabilitation Act Amendments of 1998.
www.crunchy.com

GeoNorth Ships QueryMill 2.0

(Anchorage, AK) – GeoNorth, LLC, is now shipping version 2.0 of its QueryMill product. QueryMill is a thin-client, ColdFusion/HTML database-querying solution that allows customers or distributed users to perform ad hoc querying or run predefined queries. Implemented as a custom tag, QueryMill works with any ODBC or OLEDB data source.
www.geonorth.com

Ericom's PowerTerm Now Supports ColdFusion

(Hackensack, NJ) – Ericom, Software, Inc., announced that its PowerTerm Host Publisher software now supports the Macromedia ColdFusion Web application server.

PowerTerm Host Publisher provides a toolkit to integrate legacy applications to the Web to drive CRM, sales automation, real-time transaction processing, and other e-business initiatives. Adding support for the ColdFusion environment enables organizations to leverage corporate data and business logic while developing Internet applications for e-business.
www.ericom.com

PaperThin Inks Distribution Agreements

(Boston) – PaperThin, Inc., announced that it has signed distribution agreements with five new strategic partners.

These Web consulting and hosting partners will work closely with PaperThin to resell, implement, and/or host CommonSpot Content Server, PaperThin's flagship content management solution, in the U.S. and Europe.

Joining PaperThin's partner network are CFDynamics (www.cfdynamics.com), Imagicians Interactive (www.imagicians.com), Kudos Information Limited (www.kudos-idd.com), Ringger (www.ringger.com), and Woodbourne Solutions (www.woodbournesolutions.com).
www.paperthin.com

Macromedia Releases Dreamweaver 4 Jump Start Kit

(San Francisco) – Macromedia, Inc., introduces the free Macromedia Dreamweaver 4 Jump Start Kit, which offers comprehensive resources to help emerging Web developers master Dreamweaver 4 and Macromedia Fireworks 4. The kit includes video tutorials, book excerpts, templates, and product extensions, and is provided to new customers as part of the registration process as well as customers upgrading to Macromedia Flash 5, Dreamweaver 4, Dreamweaver UltraDev 4, or the eLearning Studio.
www.macromedia.com

Ektron Releases eMPower Version 3.5

(Amherst, NH) – Ektron, Inc., announced eMPower 3.5, the newest version of its flagship Web content management application designed for small- to midsize organizations, corporate and departmental intranets/extranets, hosts, and portal environments.

New features include eWebEditPro 2.1, JavaScript syndication, Macromedia Dreamweaver and Ultradev extensions, display tag, and turnkey hosting.
www.ektron.com

CFDJ Sister Magazine "Most Trusted"

(Montvale, NJ) – Evans Data Corporation has ranked ***Java Developer's Journal*** as "the most trusted developer publication" among developers who use Java, according to **SYS-CON Media**, the world's leading *i*-technology publisher. The research results were published in the Evans Data Developer Marketing Patterns 2001 Annual Report, an independent market research report prepared by the leading market research firm serving software and developer markets.

"We are very pleased to see, in our sixth year of publication, that ***Java Developer's Journal*** continues to serve the fast-growing Java developer community as the hands-down leader of quality Java information in the world. ***JDJ***'s unmatched leadership is not based solely on its circulation – which is larger than all other Java publications put together – but also on the high quality of its editorial content," said Alan Williamson, editor-in-chief of the magazine. "The Evans Data report confirms what we hear from our readers and ***JDJ***'s sponsors and advertising partners every day."
www.sys-con.com

Macromedia

www.macromedia.com/downloads