


U.S. $8.99 (Canada $9.99)

# ColdFusion Developer's Journal

ColdFusionJournal.com
October 2003 Volume: 5 Issue: 10







SYS-CON MEDIA

## ColdFusion and SQL Server Permission Integration

*A guide to using DTS packages* PART 2

*James Blaha*
page 26

This is for you.

The hardcore, the dedicated, the enthusiast. The one with a passion for figuring things out. The one who builds things . . . and takes them apart again.

You give us new ways to see things.You show us things we never saw in the first place. You make things better. You make us better.

MX 2004 is here. Run with it.


Now available.
# Macromedia MX 2004.

www.mx2004.com


# It's everybody's PDF™

Finally, a software company that offers affordabe yet flexible PDF solutions to meet every customer's needs. Using activePDF™ to automate the PDF creation process eliminates the need for end-user intervention so your employees can concentrate on what they do best.

Licensed per server, activePDF solutions include a COM interface for easy integration with ColdFusion. Dynamically populate PDF forms with information from a database, convert ColdFusion web pages to PDF on the fly, dynamically print reports to PDF using CF and much more. Users can also merge, stitch, stamp, secure and linearize PDF, all at a fraction of the cost of comparable solutions. Download your free trial version today!

activePDF
Leading the iPaper Revolution

www.activePDF.com

# A Magazine Is Born. . .

By Robert Diamond

One of the most exciting times for a publishing company occurs when it's launching a new publication. Here at SYS-CON Media, we're going through that time right now as we introduce our newest publication, ***MX Developer's Journal***. *MXDJ* will be covering the full range of Macromedia's new MX technologies including Flash, Flash Professional, Flash Communication Server, Dreamweaver, Fireworks, FreeHand, Director, and of course my personal favorite – ColdFusion.


The ColdFusion content in *MXDJ* will be separate from that here in ***CFDJ*** (though you might recognize a few familiar names), thus increasing the amount of great CF coverage available every month.

We've always had requests to *CFDJ* for more info on other Macromedia projects, but never had the room. *MXDJ* now provides that space! Take advantage of our special limited-time charter subscription offer and don't miss a single issue. Please go to www.sys-con.com/mxcharter.cfm.

There's a preview issue of *MXDJ* in PDF format online now at sys-con.com/mx , and the first print issue will be available at MAX 2003. MAX 2003, for those of you who don't know, is Macromedia's official developer conference, and it takes place this year November 18–21, 2003 at the Salt Palace Convention Center in Salt Lake City, Utah. Those of you who have been to previous Macromedia and Allaire conferences know how valuable they are in terms of classes, keynotes, and community.

I've always been a big supporter of conferences for all of their educational values and, in years past, Macromedia's have been the cream of the crop. Some of the highlights of this year's upcoming classes include: Advanced Techniques for Using ColdFusion with SQL Server Track (Dave Gallerizo); Rapidly Build Web Services Applications with ColdFusion and Studio Track (Simon Horwith); Architecting ColdFusion Applications (Simon Horwith); ColdFusion Performance Analysis and Tuning (Jim Schley); MVC (Model View Controller) and ColdFusion (Leo Schuman); Object-Oriented ColdFusion (Simon Slooten); The Ideal ColdFusion Application: A ColdFusion/Java Hybrid (Drew Falkman); and ColdFusion Components from the Ground Up (Kai Koenig). Those are just a few highlights from the slew of 25+ CF sessions (www.macromedia.com/macromedia/conference/).

If you're going to MAX 2003, one other thing that you can't miss is CF Underground V, an all-day presentation by TeraTech on November 17, before MAX begins. CF Underground is a pre-conference event that features speakers, games, and other special events. If you're going to the conference, it's a no-brainer to get things off on the right foot with some additional learning.

One other topic of interest this month – one that I've touched on here before – is that of Macromedia's blogs, which are very useful in helping you to keep an eye on all the various areas of MX development. There's a particularly interesting discussion currently going on in Christian Cantrell's (Macromedia Server Community Manager) blog in which he asks, "What's Your Dream ColdFusion Feature?" It's well worth a look (and weighing in) at www.markme.com/cantrell/archives/003351.cfm. As he promises, "The person who comes up with the coolest, most interesting, and innovative new feature request will win tons of respect from everyone who reads this weblog, guaranteed."

## About the Author

*Robert Diamond is vice president of information systems for SYS-CON Media, and editor-in-chief of* **ColdFusion Developer's Journal.** *Named one of the "Top thirty magazine industry executives under the age of 30" in* Folio *magazine's November 2000 issue, Robert holds a BS degree in information management and technology from the School of Information Studies at Syracuse University. Visit his blog at* www.robertdiamond.com.

*robert@sys-con.com*

SYS-CON MEDIA

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

**vp, business development**
Grisha Davida grisha@sys-con.com
**group publisher**
Jeremy Geelan jeremy@sys-con.com
**technical director**
Alan Williamson alan@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
**director of sales & marketing**
Megan Mussa megan@sys-con.com
**advertising sales manager**
Alisa Catalano alisa@sys-con.com
**associate sales managers**
Carrie Gebert carrieg@sys-con.com
Kristin Kuhnle kristin@sys-con.com

**sys-con events**
**president, events**
Grisha Davida grisha@sys-con.com
**conference manager**
Michael Lynch mike@sys-con.com
**national sales manager**
Sean Raman raman@sys-con.com

**customer relations**
**circulation service coordinators**
Niki Panagopoulos niki@sys-con.com
Shelia Dickerson shelia@sys-con.com
Edna Earle Russell edna@sys-con.com
**manager, jdj store**
Rachel McGouran rachel@sys-con.com

**sys-con.com**
**vp, information systems**
Robert Diamond robert@sys-con.com

**web designers**
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com
**online editor**
Lin Goetz lin@sys-con.com

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

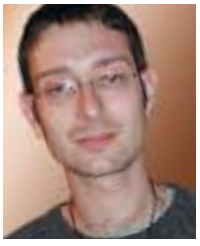
**subscriptions**
**Subscribe@sys-con.com**
1 888 303-5282
For subscriptions and requests for bulk orders, please send your letters to Subscription Department
Cover Price: $8.99/issue
Domestic: $89.99/yr (12 issues)
Canada/Mexico: $99.99/yr
all other countries $129.99/yr
(U.S. Banks or Money Orders)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



# Tales from the List

## Why run ColdFusion MX 6.1?

At the beginning of the third week of August, Mario Cilotta posted a thread to the *CFDJ* List asking for arguments in favor of upgrading from ColdFusion MX to ColdFusion MX 6.1. There are several reasons that I've chosen to focus this month's article on this thread.



By Simon Horwith

One reason is that it's a nice change of pace from the "problem-specific" threads usually featured in this column; it's not that often that business decision questions are posted to discussion lists. Another reason is that ColdFusion MX 6.1 is new, and a popular topic. My main reason for focusing on this thread however, is because it is not uncommon for developers to argue the case of using or upgrading ColdFusion to the higher-ups who make business decisions but aren't technically savvy.

ColdFusion MX 6.1 is a crucial upgrade for anyone doing ColdFusion development, but how you convince people of this fact is not always as easy as you'd think. It's also a very different "sell" than prior versions of ColdFusion (version 5 and prior).

Mario's post specified that he has two servers, one running ColdFusion MX for J2EE and the other running the stand-alone ColdFusion MX server. He explained that his request to upgrade the servers was denied due to "budgetary issues" and that "The Powers That Be" had asked him for compelling reasons to perform the upgrade. He stated that he was planning to really push for J2EE on all of the servers because of the ability to run multiple instances, and that he also planned to use the increase in <CFMAIL> performance as another reason to upgrade. Mario wanted more fuel before talking with the person resisting the upgrade. Obviously, many list members responded immediately to tell Mario that the upgrade is free, so budget shouldn't be an issue.

I also responded by reminding Mario that I'd recently posted the performance matrix for MX 6.1, which shows drastic performance benefits; and that the new server not only has better mail support, but better (and more varied) database support, better Web service support, a more robust and stable component framework, etc. I also reminded him that when running multiple instances of CF, the garbage collection doesn't impact server performance anywhere near as much as it will on a single instance. I told him that in fact I couldn't think of a single reason not to upgrade from MX to MX 6.1. Stephen Moretti seconded the argument of performance being much better and added to my list of compelling reasons that the installer is much better.

Mario sent a response to clarify that what he really needed was something that would make his boss(es) see that ColdFusion can coexist with J2EE applications (J2EE is his boss's development platform of choice). He was looking for something in plain English that describes how

*—continued on page 50*

### About the Author

*Simon Horwith is chief technology officer of eTRILOGY Ltd., a software development company based in London, England. Simon has been using ColdFusion since version 1.5 and is a member of Team Macromedia. He is a Macromedia Certified Advanced ColdFusion and Flash developer and is a Macromedia Certified instructor. In addition to administering the CFDJ-List mail-list and presenting at CFUGs and conferences around the world, he has also been a contributing author of several books and technical papers.*
*simon@horwith.com*

# Encapsulating Session State Management

## Use these techniques to simplify and improve your code

ColdFusion developers rely on session state management and the SESSION scope extensively. But as applications grow in complexity, so do the number of SESSION variables, and the risk of overwriting or misusing them. It need not be that way; with a little work (and ColdFusion Components), SESSION use can be clean, simple, and highly organized.

By Ben Forta

### When Session Data Proliferates

First, an introduction. The Web is stateless. Or put in terms that actually mean something, every request on the Web stands on its own two feet. The data received by a form submission, for example, is only available in the receiving page and not to subsequent requests. Or, user credentials specified at login are not kept until logout. And similarly, items put in a shopping cart won't still be in the cart at checkout time.

But wait a minute, those statements can't be true, can they? After all, we've all put items in shopping carts and then checked out, and we've all logged in to sites that remembered us until logout, haven't we? If the Web is stateless, how is that data maintained? That's the job of session state management, a mechanism that creates the illusion of state in a stateless world.

Data to be maintained between requests is stored on the server, along with an id that designates the client that the data belongs to. That id is sent back and forth with each and every request, so that the server can associate the maintained data and give the illusion of persistent data. In ColdFusion all this is accomplished using SESSION variables. Developers simply refer to variables like #SESSION.Firstname# and ColdFusion takes care of all the details (setting and receiving session identifiers, maintaining the SESSION data, and ensuring that the correct data is used when referring to variables with the SESSION scope).

Okay, end of introduction.

So you need to track session information, great. The first thing you do is set SESSIONMANAGEMNT="yes" in your <CFAPPLICATION> tag (which makes a lot of sense; after all, you can't use the SESSION scope without having first instructed ColdFusion to enable that functionality. Once enabled you are free to save any data within SESSION. For example, a simple variable:

```
<CFSET SESSION.FirstName="Ben">
```

or a more complex data type:

```
<CFSET SESSION.cart=ArrayNew(1)>
```

or even queries:

```
<CFQUERY DATASOURCE="dsn"
         NAME="SESSION.profile">
SELECT *
FROM Users
WHERE user_id = #FORM.user_id#
</CFQUERY>
```

There is no limit to what can be stored in the SESSION scope. That's a good thing, but it's also a liability. Why? Consider the following:

- SESSION data is available all over your application. What's to stop you from accidentally setting the same SESSION variable defined elsewhere (in code that you may not have looked at recently)?
- Unlike simple variables that are created and used in a single page, SESSION variables by their very nature are created and used all over your application. Which means that any time a change is made to how SESSION variables are used, you run the risk of breaking code that uses those variables.
- The more you use the SESSION scope, the more memory ColdFusion needs to store that data, and as the number of connected users increases, so does that resource usage. Without paying attention to all the data being stored in SES-

SION, can you be sure that you are not wasting resources?

This is just the tip of the iceberg. The key here is that because SESSION variables are so easy to create and use, their use can quickly get out of hand unless some semblance of structure is imposed.

## The Basics of Encapsulation

The word encapsulation is one of those terms that means lots of different things and usually ends up being misused most of the time. But at its simplest, encapsulation is a technique by which applications are separated into parts so that code need not know the inner workings of things it doesn't need to know.

For example, stored procedures are one of the best known forms of encapsulation. A stored procedure contains one or more database instructions in the form of SQL statements, but all that is hidden from the stored procedure user who simply makes a call to obtain data or to perform some other operation. What happens within the stored procedure is not important, what is important is that it does what it's supposed to do, it just works.

ColdFusion Custom Tags (or rather, well-written ColdFusion Custom Tags) offer another form of encapsulation. A tag is invoked to perform an operation, the details of which are concealed within the tag.

Encapsulation thus does several things:

- ***Encapsulation simplifies the creation of client code:*** Be it a stored procedure, a Custom Tag, or a Java object, client code need not worry about the inner workings of what is being invoked, and can concentrate on the task at hand.
- ***Encapsulation makes changes safe:*** Encapsulated objects, like stored procedures or Custom Tags, have defined input and output, and that is all the client code ever interacts with directly. This means that code within the encapsulated object is free to change, so long as the input and output stays the same. Database schema changing, for example? That can be buried within the stored procedure.
- ***Encapsulation helps resource management:*** When all access to an entity, for example, a database, occurs via a single entry point, it becomes possible to very effectively manage reuse, caching, and more.

I'm using the term encapsulation a little more loosely than most object-oriented developers would like, but having said that, this is indeed what encapsulation is all about. (*Note:* This idea was explained in detail in ***CFDJ,*** Volume 4, issue 10.)

## ColdFusion Components and Encapsulation

ColdFusion Components (CFCs for short), first introduced in ColdFusion MX, are a way to create reusable objects in ColdFusion. Although not objects in the purest sense, they do provide basic object functionality wrapped within the simplicity that is uniquely CFML. (*Note:* ColdFusion Components were introduced in detail in a two-part column that appeared in ***CFDJ,*** Volume 4, issues 6 & 7.)

Two of the most important aspects of CFCs is that they can store data internally, and they can persist. Let me explain. Within every CFC is a special scope named THIS. THIS contains some default data, but it can also be used to store data of your own. For example, the following method accepts two arguments (first and last name) and then saves them into the THIS scope:

```
<CFFUNCTION NAME="SetName"
            OUTPUT="no">
   <CFARGUMENT NAME="NameFirst"
               TYPE="string"
               REQUIRED="yes">
   <CFARGUMENT NAME="NameLast"
               TYPE="string"
               REQUIRED="yes">
   <CFSET THIS.NameFirst=ARGUMENTS.NameFirst>
   <CFSET THIS.NameLast=ARGUMENTS.NameLast>
</CFFUNCTION>
```

To invoke this method you could use the following code (assuming the previous method was saved in a file named user.cfc):

```
<CFINVOKE COMPONENT="user"
          METHOD="SetName"
          NAMEFIRST="Ben"
          NAMELAST="Forta">
```

This next method returns a string made up of the saved first and last name:

```
<CFFUNCTION NAME="GetName"
            RETURNTYPE="string"
            OUTPUT="no">
   <CFRETURN THIS.NameFirst & " " &
THIS.NameLast>
</CFFUNCTION>
```

So, SetName saves the name and GetName retrieves it, so the following code should save my name and return it as a string:

```
<CFINVOKE COMPONENT="user"
          METHOD="SetName"
          NAMEFIRST="Ben"
          NAMELAST="Forta">
<CFINVOKE COMPONENT="user"
          METHOD="GetName"
          RETURNVARIABLE="FullName">
```

If you were to execute this code, however, you would throw an error. The SetName call will work, but GetName will complain that THIS.NameFirst and THIS.NameLast do not exist. Why? After all, they were just set in SetName?

The problem with the above invocation is that the user component is being invoked twice, two separate invocations that have nothing to do with each other. Each <CFINVOKE> loads the component, invokes the appropriate method, and then unloads the components. So when GetName is executed there is no NameFirst and NameLast in THIS; those were in the previously invoked instance.

The solution? Persistence. Aside from being a type of file, a CFC is a ColdFusion data type, an object. <CFINVOKE>, as used above, loads the object, uses it, and then unloads it. But those steps can be separated. Look at this example:

```
<CFOBJECT COMPONENT="user"
          NAME="userObj">
<CFINVOKE COMPONENT="#userObj#"
          METHOD="SetName"
          NAMEFIRST="Ben"
          NAMELAST="Forta">
```

Here the component is being loaded as an object (which it actually is). The <CFOBJECT> instantiates (creates an instance of) the user object, but does not invoke any method. Rather, it stores the object in a named variable. <CFINVOKE> then invokes the previously loaded object; notice that the value passed to COMPONENT is the object (as opposed to the name of the CFC). Once an object is loaded it can be used multiple times, and as it is the same object being used over and over, all invocations share the same

object and thus the same internal THIS scope.

Here's a corrected version of the code to set and get the user name:

```
<CFOBJECT COMPONENT="user"
          NAME="userObj">
<CFINVOKE COMPONENT="#userObj#"
          METHOD="SetName"
          NAMEFIRST="Ben"
          NAMELAST="Forta">
<CFINVOKE COMPONENT="#userObj#"
          METHOD="GetName"
          RETURNVARIABLE="FullName">
```

<CFOBJECT> instantiates the object, SetName stores the values into THIS, and GetName returns it (possibly to be displayed).

This can also be accomplished using object style invocation. For example, the object instantiation could be performed using:

```
<CFSET userObj=CreateObject("component",
"user")>
```

and the GetName could be executed as:

```
<CFSET FullName=userObj.GetName()>
```

or used directly for display as:

```
#userObj.GetName()#
```

## Session Encapsulation

So, components are objects and can persist. The examples thus far loaded the object as local variables (type VARIABLES, the default variable type). But other scopes may be used too. Components may be loaded into REQUEST, for example:

```
<CFOBJECT COMPONENT="user"
          NAME="REQUEST.userObj">
```

and components may even be loaded into persistent scopes like SESSION:

```
<CFSET SESSION.user=CreateObject("component",
"user")>
```

Which brings us back to session state management. Instead of defining and accessing all sorts of SESSION variables throughout your application, you could define just one, an object (an instantiated ColdFusion Component). If your application uses user data you may want to create a user.cfc, which would contain all user information (including obtaining information from underlying databases). To check if a user has logged in you'd use code like this:

```
<CFIF NOT IsDefined("SESSION.user")>
... redirect to login page ...
</CFIF>
```

The login page would authenticate the user (if needed) and then create an instance of the user object in the user's SESSION scope:

```
<CFOBJECT COMPONENT="user"
          NAME="SESSION.user">
```

You may then want to initialize the object so as to populate the internal THIS with any needed information (user name, color preferences, language choices, and so on). Perhaps you'd simply pass the user id to an initialization method immediately after object creation:

```
<CFSET SESSION.user.Init(user_id)>
```

Init() should probably return a true or false flag (indicating whether or not the initialization was successful), and within the CFC you'll probably want each method to ensure that Init() was called before begin executed, but that is all internal to the CFC.

What about user logout? Simple; when a user logs out you'd kill SESSION.user like this:

```
<CFSET StructDelete(SESSION, "user")>
```

so that on a subsequent request the login and initialization process would restart.

With user.cfc you can do anything, so perhaps you'd have methods like these:

- ChangePassword
- GetLanguage
- GetFirstName
- GetLastName
- GetFullName
- GetDisplayName
- IsMember
- IsAdmin

As you need new methods, you'll simply add them to the component, and the client code (your ColdFusion code that uses the component) need know nothing of the internal object workings.

And this goes beyond user processing. Consider a shopping cart example. Shopping carts are stored in SESSION variables, but instead of storing arrays or structures or arrays of structures of whatever in SESSION and accessing them directly, you'd create a cart component. To start shopping you'd create an instance of the cart:

```
<CFOBJECT COMPONENT="cart"
          NAME="SESSION.cart">
```

When an item is to be added you'd call the appropriate method:

```
<CFINVOKE COMPONENT="#SESSION.cart#"
          METHOD="AddItem"
          ITEMID="#itemid#"
          QUANTITY="#FORM.qty#">
```

Other methods would update or remove items, and perhaps a list method would return a query (a ColdFusion query created within the CFC using the QueryNew() function) for displaying or processing. There is no limit to what you can do within a CFC, and your CFC code can evolve and adapt independent of any calling code.

## Conclusion

ColdFusion Components are objects. CFCs facilitate the encapsulation of data and logic, and they can be made to persist if needed. The combination of these two features makes CFCs perfect for managing session state. With minimal work the techniques described here can be used in any application, and doing so will both simplify and improve your code.


### About the Author

*Ben Forta is Macromedia's senior product evangelist and the author of numerous books, including* ColdFusion MX Web Application Construction Kit *and its sequel,* Advanced ColdFusion MX Application Development, *and is the series editor for the new "Reality ColdFusion" series. For more information visit www.forta.com.*

*ben@forta.com*

# BlueDragon *is* CFML

BlueDragon

**BlueDragon runs CFML natively within**

**A Standalone Server**
**Any J2EE Application Server**
**The .NET Application Framework**

New Atlanta COMMUNICATIONS

With over 10,000 corporate and individual customers in over 65 countries around the world, New Atlanta has provided reliable server-side products since 1997.

# Extending ColdFusion

## A simple way to track sessions

Welcome to another edition of **Extending ColdFusion**, where we discuss the multitude of ways you can extend your ColdFusion programming with user-defined functions (UDFs), custom tags, CFCs, and other methods. In this edition, we are going to delve into a "gray" area of CF development – use of hidden/internal methods. These are methods that exist within ColdFusion but are not documented. Be warned – while all of the following code works just fine in MX 6.1, there is no guarantee that it will continue to work in future versions. That being said, there is some cool functionality if you are willing to take some minor risks!

One of the most requested features for ColdFusion is a simple way to track sessions. As you know, ColdFusion has very simple and easy-to-use session management. What it does not have is a way to grab a list of sessions for an application. This can be useful for many reasons, and as I said, has long been requested. The UDF we will look at will use one of the unsupported methods of the ColdFusion server to get this information:

```
<cfscript>
function getSessions(appName) {
   var tracker =
createObject("java","coldfusion.runtime.SessionTracker");
   return tracker.getSessionCollection(appName);
}
</cfscript>
```

By Raymond Camden

This UDF, written by Samuel Neff (sam@rewindlife.com), is extremely simple. One line grabs an instance of the Java object, coldfusion.runtime.SessionTracker. The second line calls a method on this object, getSessionCollection(), that returns a structure of sessions based on the application name you pass in. The structure is keyed by a unique session identifier, but if you really want to identify a particular session you'll have to look at the data. Figure 1 shows an example of running the following call: <cfdump var="#getSessions(application.applicationName)#">


**Figure 1:** An example of running <cfdump var="#getSessions(application.applicationName)#">

### What is application.applicationName?

This is one of the default values that will always exist in the application scope. As you can probably guess, it returns the name of the current application.

So what are some uses of this code? On CFLib.org, I have a scheduled event that runs once an hour. It records the total number of sessions to the database. In my administrator, I can then display a report of how many sessions were active on an hourly basis. CFLib.org also supports an optional login system. Since this information is stored in the session scope, I cannot only record how many active sessions there are, but how many sessions are from

# Liberate your ColdFusion Talents

## NQcontent v2
WEB CONTENT MANAGEMENT

### The only CMS with the developer in mind

NQcontent is the ColdFusion based Content Management platform that eliminates repetitive development work and extends your talents to give you room to deliver robust, dynamic, websites that work all the time!


- Best Content Management Tool 2002 **WINNER**
- Most Innovative Application 2002 **WINNER**
- Best e-Business Software 2002 **WINNER**
- Best Web Application 2002 **WINNER**

Visit **www.nqcontent.com** for more information

people who are active members of the site. The following code shows the script called by the scheduled task:

```
<cfscript>
function getSessions(appName) {
   var tracker =
createObject("java","coldfusion.runtime.SessionTracker");
   return tracker.getSessionCollection(appName);
}
</cfscript>

<cfset nSessions = 0>
<cfset nLoggedIn = 0>

<cfset sessions = getSessions(application.applicationName)>
<cfset nSessions = structCount(sessions)>

<cfloop item="s" collection="#sessions#">
   <cfif structKeyExists(sessions[s],"username")>
         <cfset nLoggedIn = nLoggedIn + 1>
   </cfif>
</cfloop>

<cfquery name="updStats" datasource="#application.dsn#">
   insert into tblStats(numberSessions,numberLoggedIn)
   values(#nSessions#,#nLoggedIn#)
</cfquery>
```

The first few lines are simply the UDF repeated again. Then we create two variables – one for the total number of sessions and one for the number of logged-in users. To get the total number of sessions, all we need is the number of keys of the struct, which is returned by structCount(). To get the number of logged-in sessions, we loop over the structure and check for the "username" key. If it exits, it's a session with a logged-in user. Lastly, we insert both values into the database. (The table contains a column with a timestamp that defaults to the current time.) Figure 2 shows a graph report of the last 12 hours of sessions on the site.

As you can probably tell, there are numerous possibilities

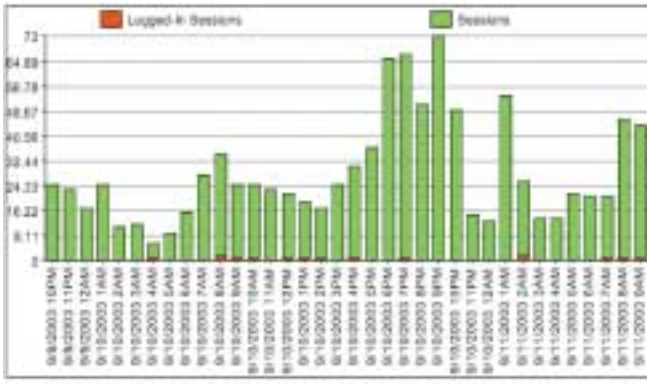


**Figure 2:** A graph report of the last 12 hours of sessions

available with this UDF, but please remember that this is an unsupported function. As I said in the beginning, this feature has been requested for many years. If you would like to see this "hack" added as a real feature of ColdFusion, I strongly urge you to voice your opinion at www.macromedia.com/go/wish. To see more hidden features of ColdFusion sessions and applications, see the blog entry at rewindlife.com: www.rewindlife.com/archives/000069.cfm

Now, what do you do if you don't want to use hidden functions? It is possible to track sessions, but it's just a little bit more difficult. One simple way is to include the following code in your Application.cfm file:

```
<cfparam name="application.sessions" default="#structNew()#">
<cfset application.sessions[session.urlToken] = now()>
```

All this code does is define an application structure called sessions and then sets a key equal to session.urlToken, a value that is unique per session, with the value of the current time. Why do we set the time? ColdFusion automatically "clears up" sessions that have expired. For our solution, we have to do it ourselves. So, if you wanted to count the number of sessions online, you could simply loop over each value, check if it's more than your session timeout, and if so, remove the key. If you wanted to store additional information, you could simply use a more advanced structure:

```
<cfparam name="application.sessions" default="#structNew()#">
<cfset application.sessions[session.urlToken] = structNew()>
<cfset application.sessions[session.urlToken].lastHit = now()>
<cfset application.sessions[session.urlToken].whatever = whatever>
```

In this example, I create a substructure for each session and save the time of the last hit, along with a variable called whatever.

In conclusion, I hope you find the methods described above useful for your projects. If you would like to suggest a particular UDF, custom tag, CFC, or some other extension for this series, please e-mail me at jedimaster@mindseye.com.


## About the Author

*Raymond Camden is co-technical editor of* **ColdFusion Developer's Journal** *and a senior software engineer for Mindseye, Inc. A longtime ColdFusion user, Raymond is a co-author of the "Mastering ColdFusion" series published by Sybex Inc, as well as the lead author for the* ColdFusion MX Developer's Handbook. *He also presents at numerous conferences and contributes to online webzines. He and Rob Brooks-Bilson created and run the Common Function Library Project (www.cflib.org), an open source repository of ColdFusion UDFs. Raymond has helped form three ColdFusion User Groups and is the manager of the Acadiana MMUG.*

*jedimaster@mindseye.com*


# For answers call us at 1-866-EDGEWEB

3 3 4 3 9 3 2

When calling your web host for support you want answers, not an annoying song stuck in your head from spending all day on hold. At EdgeWebHosting.net, we'll answer your call in two rings or less. There's no annoying on-hold music, no recorded messages or confusing menu merry-go-rounds. And when you call, one of our qualified experts will have the answers you're looking for. Not that you'll need to call us often since our self-healing servers virtually eliminate the potential for problems and automatically resolve most CF, IIS and ASP problems in 60 seconds or less with no human interaction. Plus, our multi-user support system allows you to track support requests for each of your engineers individually, lookup server availability, receive a copy of all errors on your site in real time, and even monitor intrusion attempts on your site in real time. **For a new kind of easy listening, talk to EdgeWebHosting.net**

**By the Numbers:**

- 2 Rings or less, live support
- 100% Guarantee
- 99.998% Uptime
- 150 MBPS Fiber Connectivity
- 24 x 7 Emergency support
- 24 Hour free backup

EDGE WEB HOSTING

*What are you WAITING for?*

**www.edgewebhosting.net**

**Win a year of free hosting***

*On Shared Hosting or the equivalent value
See http://edgewebhosting.net/cfdj for details

**Shared Hosting ¥ Managed Dedicated Servers ¥ Semi-Private Servers**
**ColdFusion ¥ SQL Server ¥ .NET ¥ Self-Healing Servers ¥ Value Priced**

# Design Patterns in ColdFusion: Creational Patterns

## Constructing and using CFC instances PART 1

With the recent introduction of ColdFusion MX 6.1, developers continue to explore the new features and improvements ColdFusion has to offer. What you may discover, as I have, is that we have hardly scratched the surface of what is possible with ColdFusion Components (CFCs). Introduced last year with the release of ColdFusion MX, CFCs were a radical leap forward for the language. Getting used to this new paradigm, and getting the most out of it, will take a while even for the most seasoned CF developer.

By Brendan O'Hara

ColdFusion MX is built on top of the Java/J2EE platform; in fact MX 6.1 now includes a full version of JRun, Macromedia's Java application server. ColdFusion has the ability to directly connect to and utilize Java natively. This gives developers the tremendous power of Java in addition to the Rapid Application Development (RAD) capabilities of ColdFusion and CFML.

However, most of us aren't Java experts, and J2EE projects, although highly scalable and stable, aren't easy or fast to develop. This is especially true in comparison to ColdFusion. In addition to this new (in MX) and expanded Java interactivity, we also have a CFML extension to the language. ColdFusion Components, or CFCs, can be used in many of the same ways as Java objects. They are not pure OOP but they are pure CF. That is, after a small learning curve on creating and using CFCs in your code, the actual writing of CFCs is done completely in ColdFusion so as to make it open to the entire range of CF developers.

So what are CFCs good for? Are they simply static function libraries? Are they object-light components used for Flash Remoting or Web services? Are they a replacement for custom tags? Are they objects? The answer to all of these questions is "Yes." They can be all of those things and so much more. If you have read a number of my previous articles you know I love CFCs and use them for just about everything.

### Static Methods and Function Libraries

Since we are at release 6.1, I'll assume everyone has had some experience with user-defined functions, or UDFs. The script-based variety was introduced in ColdFusion 5.0. UDFs are extremely useful ways to expand the functionality of ColdFusion. UDFs are custom functions we create to eliminate repetitive coding of simple processing routines or to ensure that those routines aren't forced to live inside a custom tag as they had to before ColdFusion 5.0. In ColdFusion MX, a tag-based UDF syntax was introduced, using the <CFFUNCTION> tag, that further expanded UDFs and also makes up the basis for the writing methods inside a CFC. To use a CFC method as a static (or nonpersistent) function, use this syntax:

```
<cfinvoke  component="tax"
    method="getRateByState"
                returnvariable="taxrate"
                state ="PA" />
```

This method returns the sales tax rate for a passed-in state code. This is one approach. Another is to use the <CFOBJECT> tag to create a variable that is a reference to an entire collection of functions or a function library. To create a reference to your function library, you call <CFOBJECT>, the same as creating any other CFC reference:

```
<cfobject component="myTaxFunctionLibrary"
  name="tax">
```

Since this CFC instance "persists" you can access any function from your library using <CFINVOKE> as shown:

```
<cfinvoke component="#tax#"
   method="getRateByState"
   returnvariable="taxrate"
                state ="PA" />
```

You can also call the function inline; simply reference with dot (.) syntax as in #CFC.Method(args)#. This is available because the variable "tax" references the CFC instance, which is our function library.

```
#Tax.getRateByState("PA")#
```

While this is the way static function libraries would be done in Java, it isn't in my opinion the best way to do it in ColdFusion. I'll explain why. In general, objects have Class data and Instance data as well as Class methods and Instance methods. Class data is the same for every instance of a CFC. Instance data is different for most or all instances of the CFC. Instance data is usually set in a constructor or initialization method often named init(). If a CFC method

doesn't use any "instance data" and doesn't make any <CFQuery> database calls, it probably shouldn't be an instance method. Instance methods access instance-specific data for dynamic runtime processing. Class methods act exactly like UDF functions. They process dynamically, but they do so based only on passed-in arguments, since like a UDF, they have no instance data to use.

In Java you are forced to create static functions within a Class because everything is an object in Java. ColdFusion, while more structural in nature, does not have that "limitation". Perhaps the best solution in ColdFusion is to treat static functions similarly to a JavaScript function library. You are able to create a .CFM file containing all of your UDFs. You can then <cfinclude> this at the top of your page (or perhaps in Application.cfm) and you will now have these UDF functions available within your pages. The syntax for this is below:

```
<cfinclude template="/myUDFLibrary.cfm">

PA Tax Rate: <cfoutput>#GetRateByState("PA")#</cfoutput>
```

In that case you do not need a prefix to identify the UDF, since the function doesn't belong to a CFC instance, but you do need to be careful you do not have two UDFs with the same name. In fact, if you have a CFC with 50 methods and a .CFM file with 50 UDFs, although the functions are identical, they do not take the same amount of time to process. The CFC takes significantly more time to process although you could cache the CFC instance in the Session, Application, or even Server scope. In that case it will take a little longer the first time through, but after that, no time at all.

So what rule do I go by when making this determination? I ask myself the following questions:

1. Do I have a CFC that the function is directly related to?
2. Does the function access a database or another file?
3. Do I need to call that function directly from Flash?
4. Do I want/need to call that function as a Web service?

If any of these are true, I put it in a CFC. For 95% of standalone functions, a .CFM file as a UDF library is the best bet. By far the most comprehensive collection of UDFs on the Web is the repository at the Common Function Library Project (www.cflib.org). The site's creators, Raymond Camden and Rob Brooks-Bilson, as well as other notables such as Ben Forta, have crafted literally thousands of useful functions to extend ColdFusion 5.0, MX, and MX 6.1.

## Components As Components?

Whether utilizing Flash Remoting or Web services, there are numerous ways to get information out of CFCs. In Flash Remoting, unless you store the CFC instance in the Session scope, all access is through static or nonpersistent methods. Web services, like a remote procedure call, do not actually create a session when they connect. They cannot store a CFC instance in Session scope and then retrieve it so they are also completely static. When CFCs are used with a number of related methods but do not use any instance data and/or they need to be accessed remotely, then they are less "objects" than "components." They do not persist data, are less dependent on instantiation, and are more likely to be accessed via Web services to perform some specific task on passed-in data. This is more akin to how COM objects are routinely used. Many in the software development industry have been extolling the virtues of a new type of programming referred to as "component-based development." Think of it as an extension or perhaps an alternative to object-oriented design. While ColdFusion is not entirely object-oriented or component-based, it has the ability and facility to work in both paradigms.

For developers interested in utilizing design patterns with Flash Remoting, I must highly recommend Sean Corfield's article on the Macromedia Web site that talks extensively about using the Facade design patten with CFCs for Flash Remoting. The article can be found at www.macromedia.com/devnet/mx/flashremoting/articles/facades.html.

## Components As Objects!

Now we arrive at the heart of the matter as it relates to design patterns and object creation. One thing we should be aware of is that objects need to be instantiated. In ColdFusion our native "objects" are CFCs so they need to be instantiated to persist across method calls and be treated like real objects in the OOP sense. Creational patterns tend to encapsulate and abstract the instantiation process. That is, we may make creating a CFC through <CFOBJECT> or createObject() the responsibility of another CFC or perhaps even a custom tag. It can be useful for simplifying the code needed to initiate certain processes and to manage the construction and destruction of a large number of CFC object instances. Although it draws closer with every

release, ColdFusion MX 6.1 is not purely object oriented. As CF developers we are likely to have little experience with the benefits and drawbacks of any type of "object creation." What do we need to know or what can we relate this to? We need to know how to reference a CFC object, how to create a specific type of CFC object, and using design patterns, how to vary the actual type of CFC being created. Below is the normal instantiation process as it works in Java and ColdFusion. First you reference the object, then you call a constructor method to create the instance:

```
Java:
ClassName myInstanceName;

ColdFusion:
<CFObject component="ClassName"
                    name="myInstanceName">
```

You would then call a constructor like this:

```
Java:
myInstanceName = new ClassName(args);

ColdFusion:
<CFInvoke component="#myInstanceName#"
                    method="init"
                    argumentcollection="#argsStruct#">
```

This ColdFusion "constructor" isn't an official constructor as we see in Java. In Java the constructor is always the class name following the new keyword. In CF you can call constructors any name you want or eliminate them completely. In addition, any code in a CFC outside of a <CFFUNCTION> is processed once at the time the CFC is initialized. This generally happens on a call to <CFOBJECT> or createObject() or before the first method is invoked. Without the dynamic initialization provided by a constructor method or initialization outside of <CFFunction>, your CFC is more or less static in nature. What that means is that although you may have data that persists across method calls, if the data isn't set into the "this" or "variables" scopes through some form of dynamic initialization, then you aren't taking advantage of the fact that the instances are the same. You are therefore treating it more or less like a static UDF.

You can also simplify the call by combining the two statements to call the constructor and return a new CFC reference at the same time.

```
Java:
ClassName myInstanceName = new ClassName(args);

ColdFusion:
<CFInvoke component="ClassName"
                    method="init"
                    argumentcollection="#argsStruct#"
                    returnvariable="myInstanceName">
```

It should be noted that this combined statement example requires that the Init() method return "this", which is the internal CFC reference to itself.

## Creational Patterns

As we have already said, creational patterns encapsulate and abstract the instantiation process. This actually means a number of things. When dealing with classes or types of CFCs, creational patterns vary the class being instantiated. That is, a method that returns a CFC instance may return different types of CFC instances depending on passed-in parameters. Additionally, creational patterns encapsulate knowledge of which CFC we are actually using. This allows us to completely hide how instances are created and assembled. In fact creational patterns almost always delegate instantiation to another object.

As we do every month when talking about object-oriented design patterns we like to refer to the original "intent" of the pattern as laid out in the book (*Design Patterns: Elements of Reusable Object-Oriented Software)*. As the preeminent work on the subject of design patterns, its authors, the so-called "Gang of Four," established the general "intent" of the following creational patterns:

- ***Factory pattern:*** "Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a client defer instantiation to subclasses."
- ***Abstract factory:*** "Provide an interface for creating families of related or dependent objects without specifying their concrete classes."
- ***Builder pattern:*** "The Builder pattern allows a client object to construct a complex object by specifying only its type and content. The client is shielded from the details of the object's construction."
- ***Prototype pattern:*** "Create a prototype instance, then create copies. Create objects by asking, 'I don't know what type I want, but give me one the same as this.'"

Overall, creational patterns gain most of their advantages in their inherent flexibility. These techniques allow the developer full control over what gets created, who creates it, and how it gets created even though each object may be chosen to be created and instantiated dynamically. You write a CFC so that it can instantiate other CFCs without being dependent on any of the CFCs it instantiates. Take the following example:

```
<cfcomponent displayname="Person">
   <cfset Variables.Name = "">
   <cffunction name="Init" access="public" returntype="struct">
     <cfargument name="Name" required="Yes" default="">
     <cfset Variables.Name = Arguments.Name>
          <cfreturn this>
   </cffunction>
   <cffunction name="getName" access="public" returntype="string">
          <cfreturn Variables.Name>
   </cffunction>
</cfcomponent>
```

This CFC uses an Init() method as a constructor, which returns "this", a reference to the CFC itself, to the calling page. It takes one

**Introducing MAX.**

People like to say there's nothing new under the sun. We disagree. When it comes to digital experiences, we believe there are vast spaces of uncharted territory. And now we've created a conference dedicated to the idea.

**MAX 2003 is the ultimate Macromedia conference. It all happens November 18-21 in Salt Lake City.**

At MAX, there's a lot going on that will help Macromedia® ColdFusion® MX developers build. Master Macromedia software through hands-on sessions. Hone your skills in seminars delivered by industry experts. Share ideas with 2,000 like-minded developers and designers. Keep your competitive edge with inside information on our newest products. And more. It all adds up to a conference dedicated to getting the most out of your ideas.

We've scheduled 29 sessions dedicated solely to ColdFusion MX developers, covering topics like: CFCs, web services, Java™ integration, .NET integration, MVC and methodologies, best practices, database optimization, and more. Please join us.

**Register now.**

Classes sell out quickly, so don't wait. You can register online at **www.macromedia.com/go/max**

build.

Gold Sponsors

intel®

Microsoft®

The 2003 Macromedia® Conference

argument – the string representing the Person instance's name. That is set into the Variables scope within the CFC and persists for as long as the CFC does. I call the CFC this way:

```
<cfinvoke
        component="Person"
        method="Init"
        returnvariable="myPerson">
        <cfinvokeargument name="Name" value="Simon Templar">
</cfinvoke>
```

That's a pretty standard use of a constructor method to return the CFC instance, which is of the CFC type "Person". Now take a look at a CFC that chooses what CFC to return dynamically:

```
<cfcomponent displayname="Employee" extends="Person">
   <cfset Variables.Name = "">
   <cfset Variables.Title = "">
   <cfset Variables.EmployeeID = "">
   <cffunction name="Init" access="public" returntype="struct">
     <cfargument name="EmployeeID" type="numeric" required="Yes" default="">
    <cfquery name="EmpQuery" datasource="employee">
    SELECT EmployeeType, Name, Title
    FROM Employee
    WHERE EmployeeID = '#Arguments.EmployeeID#'
    </cfquery>
    <cfinvoke
        component="#EmpQuery.EmployeeType#Employee"
        method="Init"
        returnvariable="myEmployee">
        <cfinvokeargument name="Name" value="#EmpQuery.Name#">
        <cfinvokeargument name="EmployeeID"
         value="Arguments.EmployeeID#">
        <cfinvokeargument name="Title" value="#EmpQuery.Title#">
     </cfinvoke>
          <cfreturn myEmployee>
   </cffunction>
</cfcomponent>
```

This CFC uses an Init() method as a constructor that returns a CFC instance but it isn't a reference to that CFC. It takes one argument, which is an employee ID number. It queries an employee database to determine what type of employee this EmployeeID represents. It then takes the information from the query and dynamically calls the constructor of the type of CFC this employee represents. I call the CFC this way:

```
<cfinvoke
        component="Employee"
        method="Init"
        returnvariable="theEmployee">
        <cfinvokeargument name="EmployeeID" value="12345">
</cfinvoke>
```

This may return a "JanitorEmployee" CFC instance or a "WebDeveloperEmployee" CFC instance. This allows your code to remain independent by delegating the choice of which CFC or class to instantiate to another object and referring to the newly created CFC instance through a common interface. This allows an object to create customized objects without knowing their class or any details of how to create them.

## Stay Tuned

Next month we'll expand on the topic of object creation as well as work directly with several creational pattern examples. We will examine a small document-management system with multiple implementations that show all four of the previously mentioned creational design patterns. The document management system has at its core document objects, which are instances of document.cfc. Document.cfc is a composite object as it contains one or more "document parts" that are themselves CFC instances.

- If the document uses internal methods to return the objects that make up the parts of a document and you can use inheritance to override those methods to return different types of objects to represent those same "document parts," then we are probably using the Factory Method pattern.
- If the document object is passed as a parameter into an internal method to create the objects that make up the parts of a document and you can return different types of objects by passing in a different type of document object, then we are probably using the Abstract Factory pattern.
- If the document object has internal methods that can create each of the various objects that make up the parts of a document and you can use inheritance to override those methods to return different types of objects to represent those same "document parts," then we are probably using the Builder pattern.
- If the document management system starts with prototypical instances of each type of object that represents a part of the document, then you copy those into your document and you can alter the document structure by replacing those prototypical "document parts" with different ones, then we are probably using the Prototype pattern.

We'll take a look at each of these examples and see exactly how and when different creational patterns are appropriate. In addition, the Gang of Four design patterns book we talk about every month is described in detail by the folks at the Macromedia Web site. The article and detailed description of the book can be found at www.macromedia.com/devnet/mx/coldfusion/articles/design_patterns.html.


### About the Author

*Brendan O'Hara is one of the coauthors of* Advanced Macromedia ColdFusion MX Application Development, *published by Macromedia Press. Brendan has a Macromedia ColdFusion MX Developer Certification along with Java and Linux certifications from Penn State University. Brendan has just been named to Team Macromedia for ColdFusion. He is a ColdFusion, Java, and .NET software architect in the Philadelphia suburbs.*

*brendantohara@yahoo.com*

# 5th Annual CF_Underground Event

A Pre-Conference event to Macromedia MAX 2003

organized by TeraTech

## Monday, November 17, 2003
## 10 - 5pm

"Fantastic job guys!!" - Dominic Plouffe
"Best event so far."- David Epler
"Lots of fun!" - Amy Grothaus

## Salt Lake City, Utah

Register Today and Get ready for more fun and excitement -- ColdFusion style -- courtesy of TeraTech and MDCFUG. Learn a bunch of cool CF programming meta tricks to save you time and headaches.

Early Bird $69 until Sept. 30th

Speakers

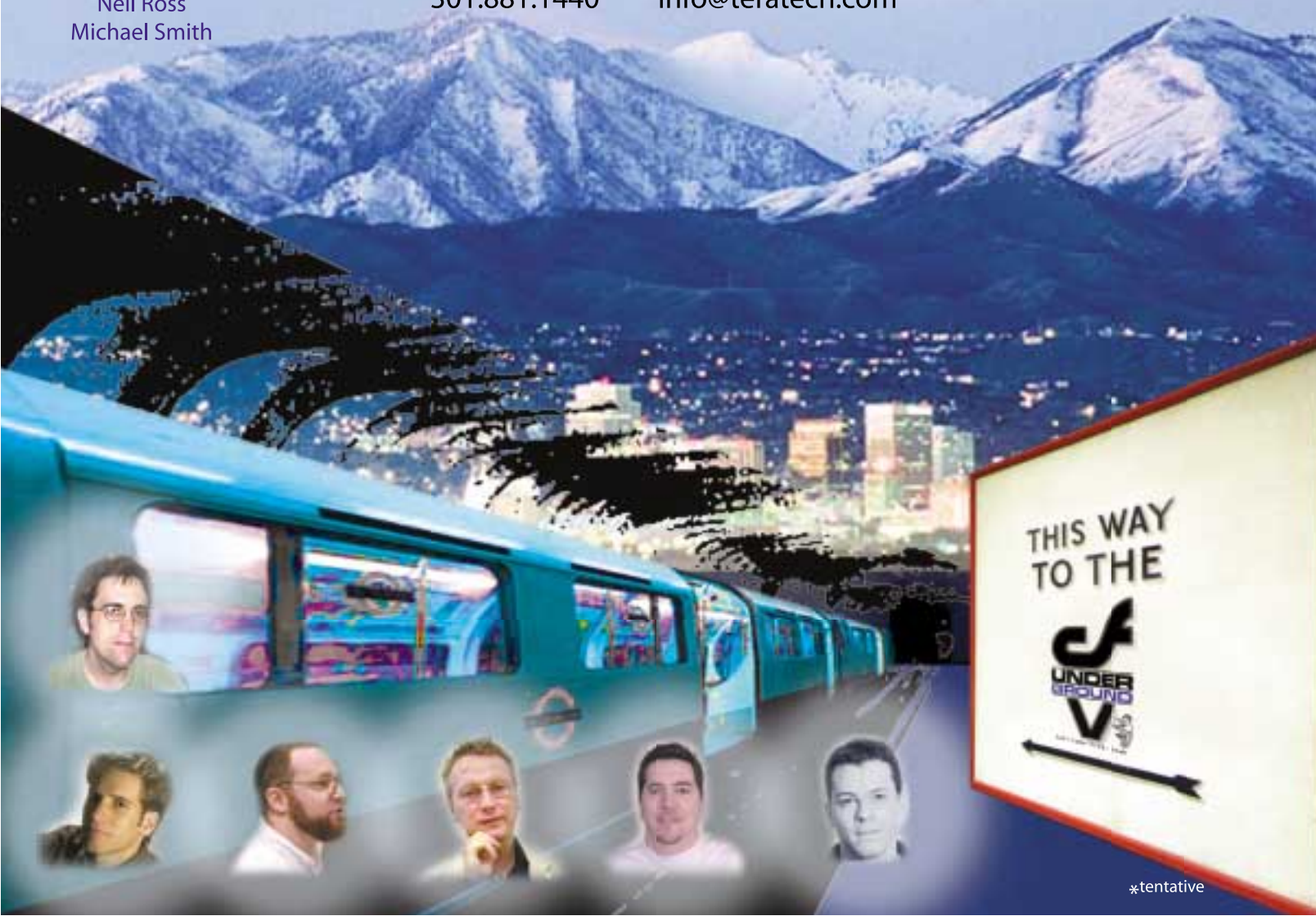
Ray Camden*
Philipp Cielen
Michael Dinowitz
Matt Liotta
Neil Ross
Michael Smith

Just $99 per person

For more information and to register, visit

www.cfconf.org/cf_underground5/

301.881.1440 info@teratech.com




*tentative

# Determining Accessibility

## Laws and guidelines PART 2

Last month, we covered the basics of accessibility, what accessibility is, and why you should care about creating Web sites and applications that are available to the disabled. This month, we'll go into much more detail about the laws and guidelines covering accessibility as well as validation programs and speech browsers to test your sites with.

By Sandra Clark

### Section 508

The United States federal government has a federal law mandating Web site accessibility (among other things). It is section 508 of the Rehabilitation Act Amendment of 1998. For our purposes, we will just call it 508. Paragraph 794d(a)(1)(A) – Development, procurement, maintenance, or use of electronic and information technology states:

> *When developing, procuring, maintaining, or using electronic and information technology, each Federal department or agency, including the United States Postal Service, shall ensure, unless an undue burden would be imposed on the department or agency, that the electronic and information technology allows, regardless of the type of medium of the technology --*
>
> *(i) individuals with disabilities who are Federal employees to have access to and use of information and data that is comparable to the access to and use of the information and data by Federal employees who are not individuals with disabilities; and*
>
> *(ii) individuals with disabilities who are members of the public seeking information or services from a Federal department or agency to have access to and use of information and data that is comparable to the access to and use of the information and data by such members of the public who are not individuals with disabilities.*

What this means to us as Web developers, is that any Web site or application that is either used by the federal government, provided through government funding, or provided by the federal government to the general public, must be accessible to people with disabilities. So for our purposes, this includes not only the Internet, but intranets and extranets as well as solutions such as content management that are used in the creation of Web sites.

There is an out: "unless an undue burden would be imposed on the department or agency, that the electronic and information technology allows, regardless of the type of medium of the technology."

However, if this section is invoked and comes to court, the "undue burden" would have to be proved, so it's not good enough simply to invoke this clause and hope you can get away with it. Section 508 does set forth the standard for identifying an undue burden, but it's a high standard. An undue burden may be considered due to fiscal constraints, but also may include security issues and the overall feasibility of making certain functionality accessible.

Fiscal constraints doesn't mean underbidding a contract and finding out that applying 508 costs more money than you thought; fiscal constraints more commonly applies to making pages that were completed before Section 508 compliant in a timely manner. The overall feasibility of making certain functionality accessible doesn't mean you use Flash as a rich Internet client on a contact form simply because it's cool. It means that if you have a complicated form that can only be designed as a grid using JavaScript that cannot be duplicated in any other way and that is required for the site you are working on that you may still do it.

The rest of Section 508 actually dictates what is necessary to make information technology (in our case, Web sites) conform to the law. You may read the entire law (and find out more information) at www.access-board.gov/508.htm.

Paragraph 1194.22(a-p) describes the accessibility that is required in a Web site.

The requirements are:

*1194.22(a)* *A text equivalent for every non-text element shall be provided (for example via alt or longdesc attributes, or in element content).*

*1194.22(b)* *Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.*

*1194.22(c)* *Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.*

*1194.22(d)* *Documents shall be organized so they are readable without requiring an associated style sheet.*

*1194.22(e)* *Redundant text links shall be provided for each active region of a server-side imagemap.*

*1194.22(f)* *Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.*

*1194.22(g)* *Row and column headers shall be identified for data tables.*

*1194.22(h)* *Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row and column headers.*

*1194.22(i)* *Frames shall be titled with text that facilitates frame identification and navigation.*

*1194.22(j)* *Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2Hz and lower than 55Hz.*

*1194.22(k)* *A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of these standards, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.*

*1194.22(l)* *When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.*

*1194.22(m)* *When a web page requires that an applet, plug-in, or other application be present on the client system, the page must provide a link to a plug-in or applet that complies with 1194.21(a).*

*1194.22(n)* *When electronic forms are designed to be completed online, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all direction and cues.*

*1194.22(o)* *A method shall be provided that permits users to skip repetitive navigation links.*

*1194.22(p)* *When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.*

## Web Accessibility Initiative

The W3C (World Wide Web Consortium) promotes the Web Accessibility Initiative (or WAI). Many governments have adopted the recommendations from this group when drafting their laws. The WAI has come up with the Web Content Accessibility Guidelines (WCAG). At the time of this article, WCAG 2.0 is in the proposal stage, so the information given here will be related to WCAG 1.0.

The WCAG has divided its guidelines into three priorities:

- **Priority 1:** *A Web content developer **must** satisfy items within this priority. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying these items is a basic requirement for some groups to be able to use Web documents.*
- **Priority 2:** *A Web content developer **should** satisfy items within this priority. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying these items will remove significant barriers to accessing Web documents.*
- **Priority 3:** *A Web content developer **may** address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document Satisfying this checkpoint will improve access to Web documents.*

In general, if you are applying the WAI to your Web development, you should aim at satisfying the Priority 1 and 2 elements. If you are developing to Section 508, you will need to make sure you satisfy all elements.

The WCAG is much more specific than Section 508. Each guideline is separated into one or more checkpoints that are also prioritized. For space purposes only, the guideline and a summary of the checkpoints are given here. For the full guidelines, please go to www.w3c.org/TR/WCAG10/.

1. **Provide equivalent alternatives to auditory and visual content.**
   • Checkpoints 1.1–1.5 cover text equivalents for any graphic, movies, multimedia, and client-side image maps.
   • Covers Priorities 1 and 3
2. **Don't rely on color alone.**
   • Checkpoints 2.1–2.2 cover making sure that information can be determined without reliance on color and that the foreground and background colors are able to be seen by people with color blindness.
   • Covers Priorities 1, 2, and 3
3. **Use markup and style sheets and do so properly.**
   • Checkpoints 3.1–3.7 cover using markup instead of images, create valid documents, use style sheets, use relative units, header elements, markup lists, and proper use of quotation markup.
   • Covers Priority 2
4. **Clarify natural language usage.**
   • Checkpoints 4.1–4.3 require making clear any changes in the language of a document's text, the expansion of abbreviations and acronyms in a document, and identifying the document's natural primary language.
   • Covers Priorities 1 and 3
5. **Create tables that transform gracefully.**
   • Checkpoints 5.1–5.6 cover identifying row and column headers, not using tables for layouts, no structural markup in tables used for layout, summaries for tables, and abbreviations for header labels.
   • Covers Priorities 1, 2, and 3
6. **Ensure that pages featuring new technologies transform gracefully.**
   • Checkpoints 6.1–6.5 cover creating documents that don't depend on style sheets, making sure that equivalents for

dynamic content is updated when dynamic content changes, making pages usable when scripts, applets, or other objects are turned off or not supportable, making sure that event handlers are device independent, and that dynamic content is accessible.
• Covers Priorities 1 and 2

7. **Ensure user control of time-sensitive content changes.**
• Checkpoints 7.1–7.5 include avoiding flickering, blinking, movement in pages, auto refreshing pages, and using sever-side redirects only.
• Covers Priorities 1 and 2

8 **Ensure direct accessibility of embedded user interfaces.**
• Checkpoint 8.1 requires scripts and applets to be either directly accessibile or compatible with assistive technologies
• Covers Priorities 1 and 2

9. **Design for device-independence.**
• Checkpoints 9.1–9.5 require using client-side rather than server-side image maps, making sure elements with their own interfaces can be operated device independently, specifying logical rather than device-independent event handlers for scripts, creating a logical tab order and providing keyboard shortcuts.
• Covers Priorities 1, 2, and 3

10. **Use interim solutions (until user agents will allow for the control of such).**
•Checkpoints 10.1–10.5 cover not using popups or other windows, make sure all form controls have a label, provide linear text as an alternative for all tables that lay text out in parallel word-wrapped columns, include default characters in edit boxes and text areas, and include non link-printable characters between adjacent links.
• Covers Priorities 2 and 3

11. **Use W3C technologies and guidelines.**
• Checkpoints 11.1–11.4 covers using the latest W3C technologies when supported, avoiding deprecated features, providing information for users to receive the document according to their preferences, and providing a link to alternative pages that are updated in conjunction with a page that cannot be made accessible.
• Covers Priorities 1, 2, and 3

12. **Provide context and orientation information.**
• Checkpoints 12.1–12.4 cover frames, form labels, and dividing large blocks of information into more manageable groups.
• Covers Priorities 1 and 2

13. **Provide clear navigation mechanisms.**
• Checkpoints 13.1–13.10 cover links, general site layout, consistency in navigation, grouping related links, search functionality for different skill levels, distinguishing information at the beginning of headings, paragraphs, lists, etc., document collections, skips.
• Covers Priorities 2 and 3

14. **Ensure that documents are clear and simple.**
• Checkpoints 14.1–14.3 cover writing the document in the simplest and clearest language appropriate for the content, adding in graphic or auditory presentations to help with page comprehension and creating a consistent style.
• Covers Priorities 1 and 3

## So What Does All That Mean?

If you want to ensure legal compliance with not only U.S. federal law, but also other countries, then you would be best served by working with WAI (WCAG) Priorities 1 and 2, which, together, cover all of Section 508 and then some. The guidelines cover all kinds of accessibility, from color blindness to keyboard access to making a Web site available to those who are blind. Many of these requirements can be quantified into business rules, such as stating that all graphics must have alt tags. These items can easily be determined via a validator. However, the same guidelines are also subjective since there's no way to determine via an automated tool whether the information within the alt tag successfully describes the graphic it is attached to.

## Validating Accessibility

So how can you determine whether the site you designed follows the quantifiable rules for Section 508 or WCAG? Well, the first thing to do is to run it through an accessibility validator. An accessibility validator will determine if you meet the objective requirements for validation.

There are two types of accessibility browsers: one that will look at open code and is useful for determining accessibility of static pages and another that will parse through your site and is more useful for validating dynamic documents. I find myself using both of them at different stages of my development process. Since I commonly do prototyping, I use the static pages first to make sure my entire design is accessible. At the end of the process I run my entire site through a site parser. I do have to note that I need to add extra items for testing purposes into my site at this time since most validators will not submit forms; however, they do work though URL links so I commonly add a URL to the form page link that goes to the same place as the form's action, for testing purposes, but is removed when my site goes live.

Rather than listing individual items here, The Web Accessibility Initiatives Web site (WAI) has an excellent listing of resources at www.w3.org/WAI/ER/existingtools.html.

Personally, I have used both the desktop Bobby tool ($299 from Watchfire) for validating my dynamic content and the UsableNet's Accessibility Suite extensions (free from usablenet.com) for checking my static prototyping. Other popular items are UsableNet's Lift ($299) for static pages and their Lift Machine ($7,200 for a 25-user license) for dynamic content.

## Speech Browsers and Other Subjective Testing

Even though validating your Web site is a good start, nothing helps more than actually running your site through a speech browser and other simulation tools to actually get a sense of whether or not your site is truly usable by someone who has a disability. An even better test of your site's accessibility is to have it tested by people who are disabled. Feedback from them is invaluable! By using these techniques you make sure your site is truly compliant with the spirit of the law.

Speech browsers range from full-featured screen readers (which work with all software applications including browsers) to Web content readers. They also vary in price from $895 for

Jaws (a full-featured screen reader) to free for Simply Web 2000. Their capabilities also vary widely. For instance, Flash MX has accessibility features built in, but they are only available to users of Window-Eyes (which only runs on Windows).

A listing of screen readers and speech browsers can be found at: www.w3.org/WAI/References/Browsing#1.

Since speech browsers and screen readers vary so much in how they deal with content that it simply isn't possible to predict how they will work, I tend to use two of them that are free. PwWebSpeak (www.soundlinks.com/pwgen.htm) has come close to providing the same sort of browsing experience as IBM Home Page Reader. At the other end of the spectrum, Simply Web 2000 (www.econointl.com/sw/) is such a nontraditional screen reader that I know if I can make my content readable in it, then it truly is accessible to all other browsers.

Other ways to make sure your site is accessible is to run it through a color blindness testing tool such as www.q42.nl/demos/colorblindnesssimulator/ or Vischeck (www.vischeck.com/vischeck/vischeckURL.php), which will render your page through a filter to let you visually see how it would look to someone with this disability.

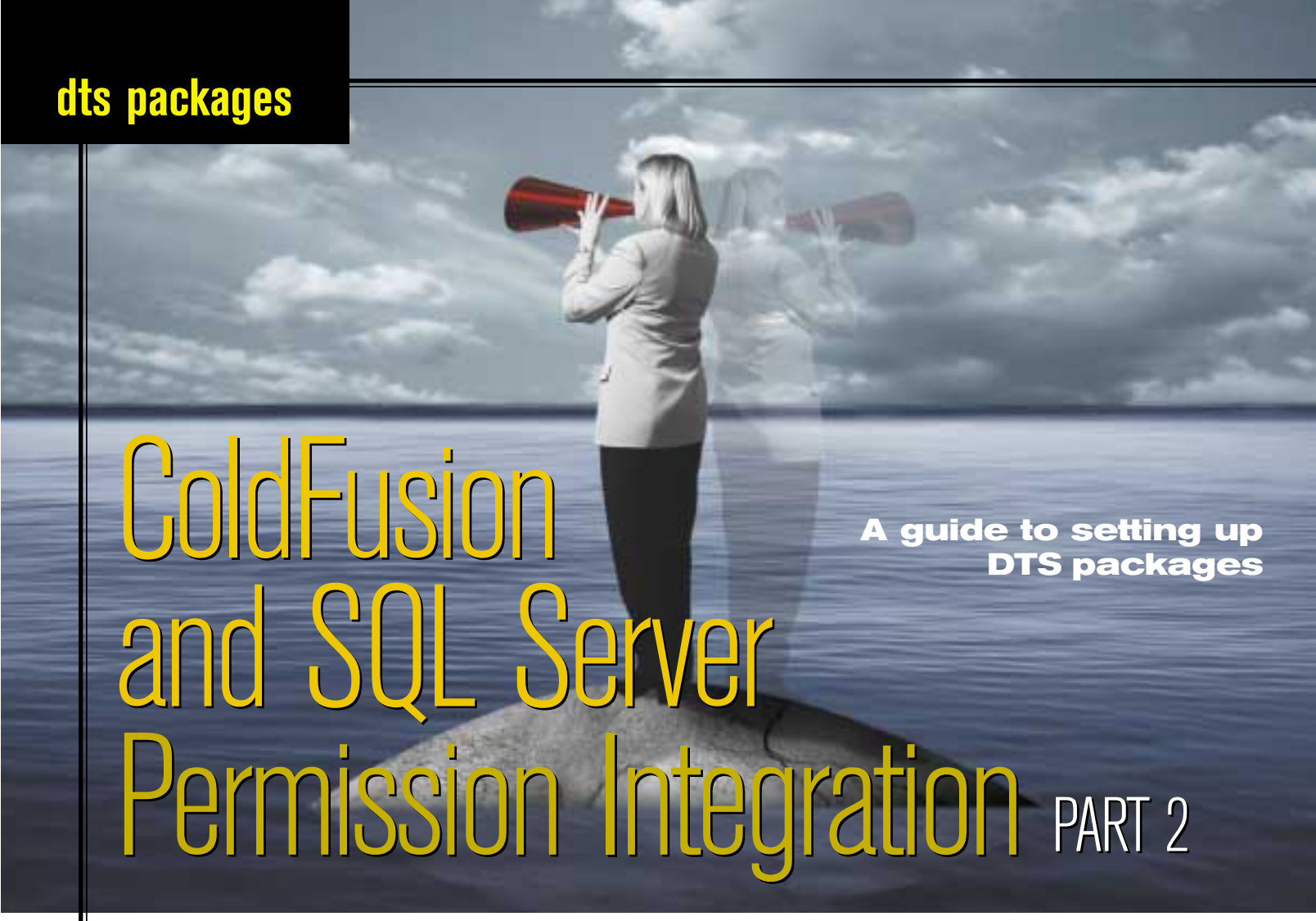
Can your user make the font larger on your site by using the text sizing command from the view menu? If so, how does it look when the larger font is applied? Is your Web site still usable with JavaScript and Style Sheets turned off? Can you use your Web site successfully by navigating through it only with your keyboard?

## Conclusion

This article provides a lot of technical information regarding Section 508 and the Web Accessibility Content Guidelines, including links to speech browsers. I recently came across an online book by Joe Clark called "Building Accessible Web Sites" at www.joeclark.org/book/sashay/serialization/. It's a good read with lots of relevant, down-to-earth information. Designing a Web site for accessibility is as much an art as a science. But fortunately, it isn't as hard as you might think. Next month, we'll discuss using Web standards as a way of easily creating accessible Web sites.


## About the Author

*Sandra Clark, a Macromedia Certified Advanced ColdFusion developer, is a senior software developer with the Constella Group in Bethesda, Maryland. She has contributed material to the* ColdFusion 5.0 Certified Developers Study Guide *published by Syngress Media/Osborne McGraw Hill and is an author on* Discovering Fusebox 4, *by Techspedition, Inc. She has also spoken at various User Groups and ColdFusion user conferences around the country.*
*slclark@shayna.com*



# ColdFusion and SQL Server Permission Integration PART 2

## A guide to setting up DTS packages

This article is a continuation of the how-to guide (Part 1 ran last month) for setting up a ColdFusion 5 server and a Microsoft SQL Server 7.0 to execute a DTS package through the ColdFusion server.

The main objective of the guide is to create a DTS package that will result in file output that will be delivered to a network UNC path or mapped drive using a set of stored procedures executed by ColdFusion. The configuration for this setup will also allow a ColdFusion server to properly propagate user rights across networked servers and domains for using shared access.

Although this guide focuses on the use of a ColdFusion 5 server and SQL Server 7.0, you can easily apply this information to a ColdFusion MX server and SQL Server 2000.

Let's review the key areas that were covered in Part 1:

- Description of a DTS package
- How to configure Windows services for ColdFusion and SQL Server
- How permissions are passed between servers
- Useful example code to test ColdFusion permissions and propagation of user rights
- How to configure SQL Server permissions
- How to create and execute a simple DTS package in the SQL Server Enterprise Manager

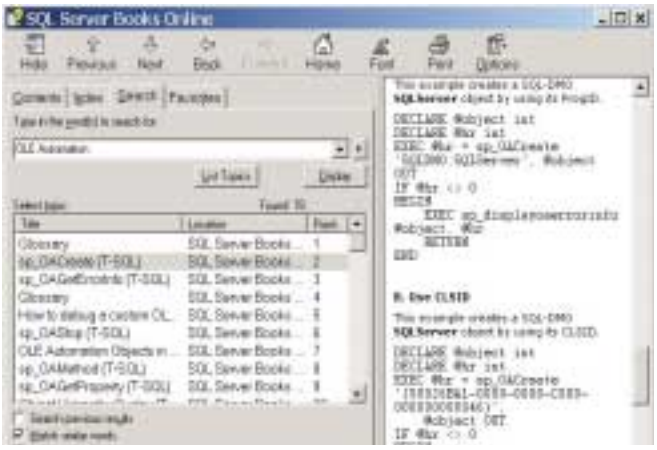
By James Blaha

### How to Install and Create the Stored Procedures in SQL Server for ColdFusion OLE Automation Execution

Complete this section logged onto the SQL Server using your ColdFusion ODBC login account.

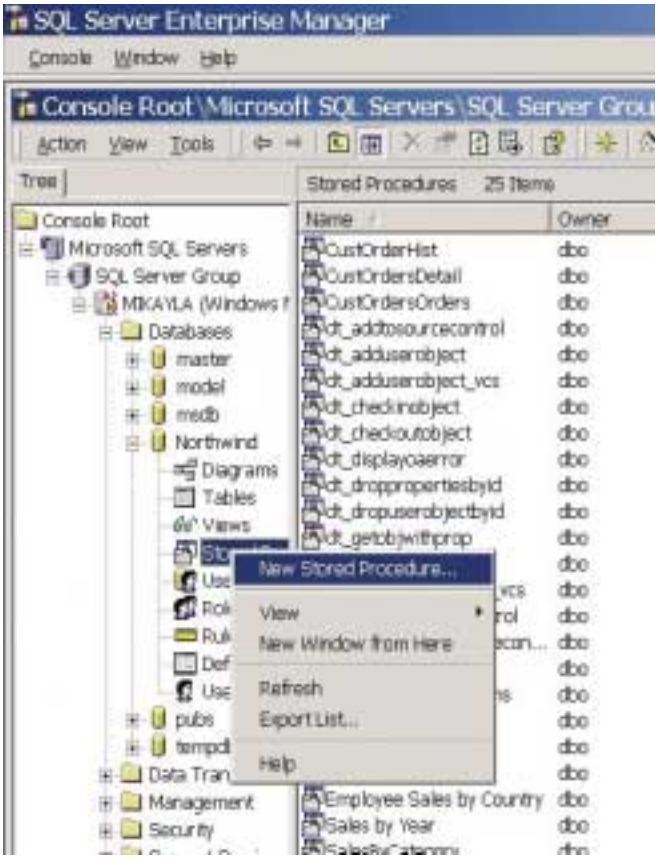
The information below is based on Dan G. Switzer's Pengoworks Web site. The origin of the code is actually based on the Microsoft Online Books for SQL Server. It provides information on running a DTS package through a stored procedure using OLE automation.

The Microsoft SQL Server Books Online is a great reference and you can easily look up any of the information contained below. Don't look in your Enterprise Manager window for it though. This information is typically located in Start>Programs> Microsoft SQL Server 7.0> Books Online, and is optional when SQL Server installs, but if you did a typical install you should already have this handy information. Two great Web sites to check for more information on this are www.pengoworks.com/index.cfm?action=articles:spExecuteDTS ("Executing a DTS Package from CF/ASP/PHP/SQL"); and www.databasejournal.com/features/mssql/article.php/1459181 ("Data Transformation Services").




## To Start:

1. You will need to create four stored procedures in your database (spDisplayPKGErrors, spExecuteDTS, sp_displayoaerrorinfo, and sp_hexadecimal). Place these stored procedures, aka SPROCs, in the desired database(s) in which you wish to execute your DTS packages from ColdFusion. These all have to be created under the same user account that you use for your ODBC connection. So if ACCOUNT1 is your ODBC user name, you must log into your SQL Server Enterprise manager with that ID when you set these SPROCs up.
2. Start by opening your database in the Enterprise Manager, right-click on "Stored Procedures", and click on "New Stored Procedure."




3. Once the new SPROC window is opened, remove the code highlighted below (CREATE PROCEDURE [PROCEDURE NAME] AS) and paste in one of the SPROCs below. Click APPLY and OK and repeat this process for the other stored procedures all listed below.


### Stored Procedure 1: spExecuteDTS

This is the main SPROC used for OLE automation. I went through and commented the code, so you'll know what's going on in the SPROC if you're interested in reading it.

```
-- Local variable(s) use: @
-- Global variable(s) use: @@
-- Note: There are over 30 global variables a.k.a. parameterless system
functions available for use.
-- E.g.@@VERSION returns the current version of your SQL Server.

-- Creating the stored procedure a.k.a. Sproc

-- Code usage below
-- CREATE PROEDURE|PROC <sproc name>
-- @parameter_name data_type> [= default|NULL] [VARYING] [OUTPUT]

-- Note: By adding [= NULL] or a value to a parameter it becomes an
optional variable from the user executing the procedure.

CREATE PROC spExecuteDTS
    @Server varchar(255),                -- Required: SQL Server name
    @PkgName varchar(255),               -- Required: Package Name
(Defaults to most recent version)
    @ServerPWD varchar(255) = Null,      -- Optional: Server Password if
using SQL Security to load Package (UID is SUSER_NAME())
    @IntSecurity bit = 0,                            -- Optional: 0 = SQL
Server Security, 1 = Integrated Security
    @PkgPWD varchar(255) = ''            -- Optional: Package Password


-- Below the AS statement is the code for the procedure.

AS
```

```
-- The SET statement is usually used for setting variables in the
fashion you see in most procedural languages.
-- Code Usage e.g. SET @Age = 21


SET NOCOUNT ON
/*
   Return Values
   - 0 Successful execution of Package
   - 1 OLE Error
   - 9 Failure of Package
*/

-- The DECLARE statement declares variable(s). Note: The default value
for the variable is always NULL.
-- Code Usage e.g. DECLARE @<variable name> <variable type>

DECLARE @hr int, @ret int, @oPKG int, @Cmd varchar(1000)


-- The EXEC|EXECUTE keyword is required since a call to a sproc wasn't
the first thing in the batch.
-- OUTPUT Parameter: Will return a value to the sproc.

-- Question: How do you run a DTS package from the Query Analyzer or
stored procedure?
-- Answer: By using OLE Automation objects such as sp_OACreate,
sp_OAGetProperty, sp_OAMethod


-- Create a Pkg Object

/* The main procedure, sp_displayoaerrorinfo, takes the return code
from any of the standard OLE Automation stored procedures,
    and displays the error Source and Description using
sp_OAGetErrorInfo. It also calls sp_hexadecimal to convert the integer
    error code into a string (char) representation of the true hexa-
decimal value. */

EXEC @hr = sp_OACreate 'DTS.Package', @oPKG OUTPUT

IF @hr <> 0                                          -- If variable
NOT EQUAL to ZERO continue.
BEGIN                                                -- Grouping the
code into code blocks BEGINS here.
   PRINT '***  Create Package object failed'
   EXEC sp_displayoaerrorinfo @oPKG, @hr
   RETURN 1

-- RETURN Values: Used to determine the execution of the sproc. FYI: 1
equals an OLE Error

END                                      -- Grouping the code into
code blocks ENDS here.

-- Evaluate Security and Build LoadFromSQLServer Statement

/* Syntax: Package.LoadFromSQLServer ServerName, [ServerUserName],
[ServerPassword],  [Flags],

[PackagePassword], [PackageGuid],

[PackageVersionGuid], [PackageName], [pVarPersistStgOfHost] */

--  ServerName = Server name.
--  ServerUserName = Server user name.
--  ServerPassword = Server user password.
--  Flags = Value from the DTSSQLServerStorageFlags constants indicat-
ing user authentication type.
--  PackagePassword = Package password if the package is encrypted.
--  PackageGuid = Package identifier, which is a string representation
of a globally unique identifier (GUID).
--  PackageVersionGuid = Version identifier which is a string repre-
sentation of a GUID.
--  PackageName = Package name.
--  pVarPersistStgOfHost = Screen layout information associated with a
package (for internal use only).


-- Syntax: SUSER_SNAME()
/* This is the user security identification number. server_user_sid,
which is optional, is varbinary(85). server_user_sid can be the secu-
rity identification number of any Microsoft SQL Server login or
Microsoft Windows NT user or group. If server_user_sid is not speci-
fied, information about the current user is returned. */

-- sp_OAMethod Calls a method of an OLE object.

--  Syntax:
--  sp_OAMethod objecttoken, methodname [ , returnvalue OUTPUT ] [ , [
@parametername = ] parameter [ OUTPUT ] [ ...n ] ]

IF @IntSecurity = 0
   SET @Cmd = 'LoadFromSQLServer("' + @Server +'", "' + SUSER_SNAME()
+ '", "' + @ServerPWD + '", 0, "' + @PkgPWD + '", , , "' + @PkgName +
'")'
ELSE
   SET @Cmd = 'LoadFromSQLServer("' + @Server +'", "", "", 256, "' +
@PkgPWD + '", , , "' + @PkgName + '")'

EXEC @hr = sp_OAMethod @oPKG, @Cmd, NULL

IF @hr <> 0
BEGIN
          PRINT '***  LoadFromSQLServer failed'
          EXEC sp_displayoaerrorinfo @oPKG , @hr
          RETURN 1
END

-- Execute Pkg
EXEC @hr = sp_OAMethod @oPKG, 'Execute'
IF @hr <> 0
BEGIN
          PRINT '***  Execute failed'
          EXEC sp_displayoaerrorinfo @oPKG , @hr
```

```
          RETURN 1
END

-- Check Pkg Errors
EXEC @ret=spDisplayPkgErrors @oPKG

-- Unitialize the Pkg
EXEC @hr = sp_OAMethod @oPKG, 'UnInitialize'
IF @hr <> 0
BEGIN
          PRINT '***  UnInitialize failed'
          EXEC sp_displayoaerrorinfo @oPKG , @hr
          RETURN 1
END

-- Clean Up
EXEC @hr = sp_OADestroy @oPKG
IF @hr <> 0
BEGIN
   EXEC sp_displayoaerrorinfo @oPKG , @hr
   RETURN 1
END

RETURN @ret
```

### Stored Procedure 2: spDisplayPKGErrors

```
-- display errors from spExecuteDTS execution
CREATE PROC spDisplayPKGErrors
   @oPkg As integer
AS

SET NOCOUNT ON

DECLARE @StepCount int
DECLARE @Steps int
DECLARE @Step int
DECLARE @StepResult int
DECLARE @oPkgResult int
DECLARE @hr int

DECLARE @StepName varchar(255)
DECLARE @StepDescription varchar(255)

IF OBJECT_ID('tempdb..#PkgResult') IS NOT NULL
          DROP TABLE #PkgResult

CREATE TABLE #PkgResult
(
   StepName varchar(255) NOT NULL,
   StepDescription varchar(255) NOT NULL,
   Result bit NOT NULL
)

SELECT @oPkgResult = 0
```

```
EXEC @hr = sp_OAGetProperty @oPkg, 'Steps', @Steps OUTPUT
IF @hr <> 0
BEGIN
          PRINT '***  Unable to get steps'
          EXEC sp_displayoaerrorinfo @oPkg , @hr
          RETURN 1
END

EXEC @hr = sp_OAGetProperty @Steps, 'Count', @StepCount OUTPUT
IF @hr <> 0
BEGIN
          PRINT '***  Unable to get number of steps'
          EXEC sp_displayoaerrorinfo @Steps , @hr
          RETURN 1
END

WHILE @StepCount > 0
BEGIN
   EXEC @hr = sp_OAGetProperty @Steps, 'Item', @Step OUTPUT,
@StepCount
   IF @hr <> 0
   BEGIN
                  PRINT '***  Unable to get step'
                  EXEC sp_displayoaerrorinfo @Steps , @hr
                  RETURN 1
   END

   EXEC @hr = sp_OAGetProperty @Step, 'ExecutionResult', @StepResult OUTPUT
   IF @hr <> 0
   BEGIN
                  PRINT '***  Unable to get ExecutionResult'
                  EXEC sp_displayoaerrorinfo @Step , @hr
                  RETURN 1
   END


   EXEC @hr = sp_OAGetProperty @Step, 'Name', @StepName OUTPUT
   IF @hr <> 0
   BEGIN
                  PRINT '***  Unable to get step Name'
                  EXEC sp_displayoaerrorinfo @Step , @hr
                  RETURN 1
   END

   EXEC @hr = sp_OAGetProperty @Step, 'Description', @StepDescription OUTPUT
   IF @hr <> 0
   BEGIN
                  PRINT '***  Unable to get step Description'
                  EXEC sp_displayoaerrorinfo @Step , @hr
                  RETURN 1
   END

   INSERT #PkgResult VALUES(@StepName, @StepDescription, @StepResult)
   PRINT 'Step ' + @StepName + ' (' + @StepDescription + ') ' + CASE
WHEN @StepResult = 0 THEN 'Succeeded' ELSE 'Failed' END

   SELECT @StepCount = @StepCount - 1
   SELECT @oPkgResult = @oPkgResult + @StepResult
END

SELECT * FROM #PkgResult

IF @oPkgResult > 0
BEGIN
   PRINT 'Package had ' + CAST(@oPkgResult as varchar) + ' failed step(s)'
   RETURN 9
END
ELSE
BEGIN
   PRINT 'Packge Succeeded'
   RETURN 0
END
```

### Stored Procedure 3: sp_displayoaerrorinfo

This information is directly from the MS Online Book. You can use the following sp_displayoaerrorinfo stored procedure to display OLE Automation error information when one of the OLE Automation procedures returns a nonzero HRESULT return code. This sample stored procedure also uses sp_hexadecimal.

```
CREATE PROCEDURE sp_displayoaerrorinfo
    @object int,
    @hresult int
AS
DECLARE @output varchar(255)
DECLARE @hrhex char(10)
DECLARE @hr int
DECLARE @source varchar(255)
DECLARE @description varchar(255)
PRINT 'OLE Automation Error Information'
EXEC sp_hexadecimal @hresult, @hrhex OUT
SELECT @output = '  HRESULT: ' + @hrhex
PRINT @output
EXEC @hr = sp_OAGetErrorInfo @object, @source OUT, @description OUT
IF @hr = 0
BEGIN
SELECT @output = '  Source: ' + @source
PRINT @output
SELECT @output = '  Description: ' + @description
PRINT @output
END
ELSE
BEGIN
    PRINT "  sp_OAGetErrorInfo failed."
    RETURN
END
```

### Stored Procedure 4: sp_hexadecimal

This information is also directly from the MS Online Book.

```
CREATE PROCEDURE sp_hexadecimal
    @binvalue varbinary(255),
    @hexvalue varchar(255) OUTPUT
AS
```

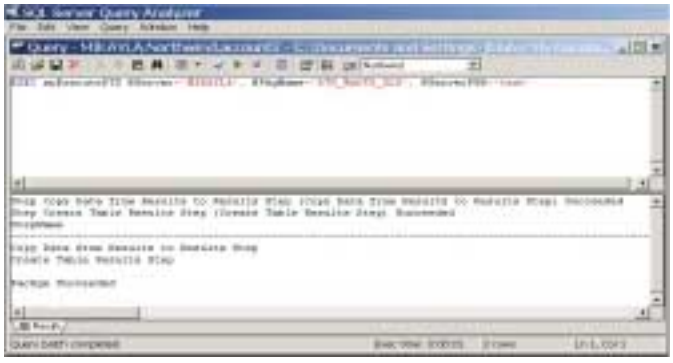
```
DECLARE @charvalue varchar(255)
DECLARE @i int
DECLARE @length int
DECLARE @hexstring char(16)
SELECT @charvalue = '0x'
SELECT @i = 1
SELECT @length = DATALENGTH(@binvalue)
SELECT @hexstring = '0123456789abcdef'
WHILE (@i <= @length)
BEGIN
DECLARE @tempint int
DECLARE @firstint int
DECLARE @secondint int
SELECT @tempint = CONVERT(int, SUBSTRING(@binvalue,@i,1))
SELECT @firstint = FLOOR(@tempint/16)
SELECT @secondint = @tempint - (@firstint*16)
SELECT @charvalue = @charvalue +
SUBSTRING(@hexstring, @firstint+1, 1) +
SUBSTRING(@hexstring, @secondint+1, 1)
SELECT @i = @i + 1
END
SELECT @hexvalue = @charvalue
```

**Your configuration is now complete.**

## How to Test Configurations in SQL Server

Complete this section logged onto the SQL Server using your same ColdFusion ODBC login account.

Open your SQL Server Enterprise Manager and click on Tools>SQL Server Query Analyzer. You'll need to test and execute the DTS package here, so you'll see any errors that may occur in the results window. This is where you'll go to help troubleshoot a failing DTS package.



If you're using SQL Server Security Authorization, which is what you are using if you're logged in with the ODBC user's ID, you can use one of the following two commands to execute the SPROC, which will execute the DTS package:

```
EXEC spExecuteDTS @Server='YourSQLServer', @PkgName='DTS_xxx',
@ServerPWD='YourLogInPassword'
```

or

```
EXEC spExecuteDTS @Server='YourSQLServer',
@PkgName='DTS_xxx',
@ServerPWD='YourLogInPassword', @IntSecurity =
0
```

If you're using SQL Server Integrated Security Authorization, you can use the following command to execute the SPROC, which will also execute the DTS package. (This would mean you're logged in to the Enterprise Manager with your Windows login account.)

```
EXEC spExecuteDTS @Server='YourSQLServer',
@PkgName='DTS_xxx',  @IntSecurity = 1
```

**Your testing is complete!**

Okay, so you're getting anxious to see these new methods work aren't you? Before we try our new SPROC method, let's first test the DTS package execution an easier way by executing it through a COM Object. You probably are wondering why I'm pushing the SPROC method over COM. For starters, MX seems to have some COM issues that I'm not sure have been resolved yet. So when we decide to migrate to MX, I want all my code to work. There are also some security concerns when using COM. And finally, the most important reason to use a SPROC over COM – speed! There is a performance gain from executing your packages through a SPROC that uses OLE automation over using the COM object to execute the package.

Cut and paste the code below into a template and execute it.

**Example for DTS Execution via a COM Object**

```
<CFTRY>
    <CFOBJECT TYPE="COM" NAME="objDTS"
     CLASS="DTS.Package" ACTION="CREATE">
    <CFCATCH TYPE = "Object">
        <CFSET error_message = "The DTS
       Package Object Could Not Be Created.">
    </CFCATCH>
</CFTRY>

<CFTRY>
    <CFSET r =
objDTS.LoadfromSQLServer("YOUR_SQL_SERVER_NAME
","YOUR_ODBC_USER_NAME"," YOUR_ODBC_PASS-
WORD",0,"","","","YOUR_DTS_Package_NAME","")>
<CFCATCH>

<CFSET error_message = "The DTS Package Could
Not Be Loaded From the SQL Server at this
time.">

</CFCATCH>
</CFTRY>

<CFIF IsDefined("error_message")>
    <CFOUTPUT>#error_message#</CFOUTPUT>
</CFIF>

<CFSET p = objDTS.Execute()>
```

There are two examples, first calling it via COM (via CFOBJECT) and then via a stored procedure (using CFPROC-PARAM). You will need to change the four options listed below:

A. YOUR_SQL_SERVER_NAME
B. YOUR_ODBC_USER_NAME
C. YOUR_ODBC_PASSWORD
D. YOUR_DTS_Package_NAME

If you're wondering what the options are for the objDTS.LoadfromSQLServer that you can use, they are listed below with a full example of the available options:

```
ObjectName.LoadFromSQLServer("ServerName",
     "ServerUserName",
     "ServerPassword",
     "Flags",
     "PackagePassword",
     "PackageGuid",
     "PackageVersionGuid",
     "PackageName",
     "pVarPersistStgOfHost")
```

Now here's what you've been waiting for, the code to call a SPROC, which will execute your DTS package from ColdFusion.

**Example for DTS Execution via a SPROC**

```
<CFSTOREDPROC PROCEDURE="spExecuteDTS"

DATASOURCE="YOUR_ODBC_USER_NAME"
               debug="Yes"
       returnCode = "Yes">

<cfprocresult name="importData">

<cfprocparam type="In"
             cfsqltype="CF_SQL_VARCHAR"
             dbvarname= "@Server"
             value= "YOUR_SQL_SERVER_NAME"
             null="no">

<cfprocparam type="In"
             cfsqltype="CF_SQL_VARCHAR"
             dbvarname= "@PkgName"
             value= "YOUR_DTS_Package_NAME"
             null="NO">

<cfprocparam type="In"
             cfsqltype="CF_SQL_VARCHAR"
             dbvarname="@ServerUserID"
             value="YOUR_ODBC_USER_NAME"
             null="NO">

<!--- <cfprocparam type="In"
                  cfsqltype="CF_SQL_VARCHAR"
                  dbvarname="@ServerPWD"
                  value="YOUR_ODBC_PASSWORD"
                  null="NO"> --->

<!--- Note: If you run this with a local SA
account or you're using the SQL Server
Integrated Security you'll need the line
below. If your code doesn't work, comment out
the line below. --->

<cfprocparam type="In"
                  cfsqltype="CF_SQL_VARCHAR"
                  dbvarname="@IntSecurity"
                  value="1"
                  null="NO">

</CFSTOREDPROC>

<hr>

<cfoutput>

The return code for the stored procedure was:
'#cfstoredproc.statusCode#' <br><br>
The execution time of the stored procedure, in
milliseconds was:
#cfstoredproc.ExecutionTime#.

</cfoutput>
```

## From the Top Down Troubleshooting

- ***Windows Server tasks completed?***
  1. Did you edit the Windows service for ColdFusion? *(ColdFusion Application Server/ ColdFusion MX Application Server)*
  2. Did you edit the Windows service(s) for the SQL Server? *(MSSQLServer & SQLServer Agent)*

- ***SQL Server completed tasks performed under the "SA" account?***
  - ***3.*** Does your SQL Server database login ID have "db_owner" rights used for the ODBC ColdFusion connection?
  - ***4.*** Do the extended stored procedures have execute rights for the needed SPROCs in the master database? (Execute rights either have to go to the ID account or to public.)
    - *A. sp_OACreate*
    - *B. sp_OADestroy*
    - *C. sp_OAMethod*
    - *D. sp_OASetProperty*
    - *E. sp_OAGetErrorInfo*
    - *F. sp_OAGetProperty*
- ***SQL Server completed tasks performed under the "login ID for the CF ODBC"?***
  - ***5.*** Create DTS package.
  - ***6.*** Did you successfully test your DTS package execution manually?
  - ***7.*** Did you create the four needed database SPROCs in your database?
    - *A. spExecuteDTS*
    - *B. spDisplayPKGErrors*
    - *C. sp_displayoaerrorinfo*
    - *D. sp_hexadecimal*
  - ***8.*** Did you test the SQL Server Enterprise Manager SPROC execution successfully?
- ***ColdFusion Server completed tasks?***
  - ***9.*** Test the ColdFusion CFOBJECT COM method for execution?
  - ***10.*** Test the ColdFusion CFSTOREDPROC method for execution?

## Still Having Problems?

Unfortunately even if you follow this guide step by step there are no guarantees for success! You may still have problems getting your DTS packages to execute properly from COM, SPROCs, or just from manually executing your DTS package in general. In most cases your issues will have to do with permissions someplace in your system. For others using COM, you may need to update your Windows OS with the latest patches and version of Internet Explorer on the ColdFusion server. However, in the end, the information contained here should provide you with enough information on where to begin looking for your problem areas and the order to follow. Good troubleshooting skills and a step-by-step testing approach are the best ways to resolve any issues.


### About the Author

*James Blaha is an Internet programmer/analyst at Pace University. His experience ranges from Internet application development to hardware and operating systems support. He also serves as the committee chairperson for HASUG, the Hartford Area SAS User Group, located in Connecticut.*

*jblaha@pace.edu*

# Growing a Business with BlueDragon Server

## It's free!

How many times have you heard the question: "Why use CFML when you can use PHP, ASP, or JSP, which are all free?" With BlueDragon Server, one of three editions of BlueDragon from New Atlanta Communications, you can finally answer that question. It's a free – yes, really free – server for CFML.

By Vince Bonfanti

And if you've wanted to build and distribute CFML applications as products, with a bundled implementation of a CFML server, you can do that with BlueDragon as well, and for far less than the cost of CF5 or CFMX.

In this month's column, I'd like to identify some of the business opportunities in store for those who consider this relatively new alternative.

### When Is Free Really Free?

In the December 2002 issue of ***CFDJ***, Ben Forta wrote a helpful article, "But It's Free" (www.sys-con.com/coldfusion/article.cfm?id=541), making the case for why the many free alternatives really don't stack up as entirely free when you consider all that CFML brings to the table. I make the point often as well: CFML is more than just a language, it's implemented in both ColdFusion and BlueDragon as a framework, providing features that might otherwise have to be added to those other alternatives.

Still, we all know of projects (or people) who just won't be motivated to ante up the $1,299 needed to get started with deploying applications on ColdFusion Server. They can be pointed to the free BlueDragon Server edition, available at www.newatlanta.com/bluedragon/.

I am often asked: "So, about this free version of the Server, do you mean free for production?" Yes, BlueDragon Server is available free for production. Go to town! Install it on your server and sell your widgets, tell folks about your charity, run a blog. Do whatever you'd like (subject to the terms and conditions of the licensing agreement shown when you download it).

### Is It 'Crippled' like CF Express Was?

Veteran CFML developers may have a bad taste toward "free CF servers." Allaire once offered CF Express, back in the 4.5 timeframe, which was (sadly) quite limited in terms of the number of tags and functions that you were allowed to use.

The old CF Express didn't even allow such things as CFMAIL, CFLOCK, CFHTTP, CFTRANSACTION, CFSCRIPT, CFMODULE, CFSTOREDPROC, CFFORM, CFERROR, CFFILE, CFDIRECTORY, CFCONTENT, CFCACHE, and lots more. (You can read more about what CF Express didn't offer by viewing the CFML Language Reference from that edition, available online at Macromedia's site: www.macromedia.com/v1/documents/cf4/acrobatdocs/40langrefe.pdf. See Chapter 1, page 7.)

BlueDragon Server offers all those tags and lots more, and clients have found that their CFML applications run very effectively in this alternative environment. For more information on our compatibility with CF5 and CFMX, see the BlueDragon Web site.

Indeed, by the time you read this, New Atlanta should have announced substantial increased compatibility, including many (if not most) of the CFMX-level language features people have been looking for.

### What's the Catch?

So what's not allowed in the free BlueDragon Server edition that is allowed in the other editions? Have you got a hand free to count along? The free Server edition doesn't support CFEXECUTE, CFGRAPH, CFOBJECT, CFWDDX, CFSCHEDULE, and CFSERVLET. (By the time you read this and more MX-level language features are announced, it's possible that some of those new features will be held back from the free edition.) Okay, you'd need six fingers to count that, but CFSERVLET is deprecated in CFMX anyway.

Speaking of servlets, the free Server edition also does not offer servlets/JSP integration, nor CFX Java custom tags. Also, the number of included DB drivers is limited, and you can only run it with Apache on Linux and IIS on Windows.

Still, for free, you're getting an awful lot. And the Server JX edition, which adds those features and more, is still at its introductory price of just $549 (regularly

$999). The J2EE edition, starting at $2,499, enables you to deploy your CFML as a standard J2EE Web app on any J2EE server, including WebSphere, WebLogic, Sun ONE, JRun, Oracle, Borland, JBoss, Tomcat, and more.

A complete grid of supported and unsupported tags in the three editions is at www.newatlanta.com/products/bluedragon/product_info/cfml_tag_support.cfm. See also www.newatlanta.com/products/bluedragon/product_info/features.cfm#FCMatrix for other feature comparisons.

## Making Money Versus Spending It

What if you're a reseller interested in using the free Server edition for your application? The argument that "another $1,300 won't kill you in the grand scheme of things" may be valid on custom programming projects (where the cost of development and management of the server far outweighs the cost of that server), but the picture changes if you want to deploy your application on many servers.

Let's say you've built a "packaged" solution written in CFML that you'd like to sell for $300/server. Guess what? Prior to knowing of BlueDragon Server you'd have to add another $1,300 to the cost of your product for the CF5/MX license, more than quadrupling the end cost to your customer (minus whatever discounts you might be able to negotiate with Macromedia).

Consider FuseTalk, makers of a leading discussion forum solution written in CFML. Their lowest priced Professional edition sells for $289. It doesn't make a whole lot of sense to expect their customers to spend $1,300 for a CFMX license (if they don't already have one) if all they want to do is run a $289 product. But, FuseTalk Professional runs on the free BlueDragon Server, and is supported by both FuseTalk, Inc., and New Atlanta.

## Redistribution: Opening New Doors of Opportunity

Many people are getting excited about the prospect of selling their CFML apps with BlueDragon as the runtime engine. Would you like to bundle BlueDragon Server along with your product, so that one installation implements both the BlueDragon Server and your CFML application (and whatever other configuration information is needed)? Would it be useful that the user doesn't even need to know about an underlying "server," nor have to use the administrator (you can remove it if you'd like)?

Would you like to bundle your CFML application in a binary format that can't be easily decoded (using our BlueDragon archive, or BDA, format)? These aren't features usually available to Server and Server JX customers, but New Atlanta offers many options for vendors wanting to license BlueDragon for redistribution.

The idea of selling your CFML application bundled with BlueDragon is what we call "redistribution" (as opposed to just pointing them to our Web site to download the free Server). If you read the licensing agreement for all of our products (which you agree to before you can download them), it states that redistribution of BlueDragon requires a separate OEM licensing agreement. It's not unusual. MySQL works similarly for commercial use and redistribution.

## Offering Your Clients a Solution: Not a Technology

If you have to tell your customer to go download the free BlueDragon Server (or indeed to buy CFMX or one of the other BlueDragon editions), you could be raising a "technology" issue. What if they have a bias against ColdFusion? Bundling the BlueDragon Server with your CFML application protects against this issue. You're offering a solution, not a technology.

Indeed, if you've got a product that you've created for the CFML community (be it a discussion forum, a content management system, a timesheet system, a help desk app, whatever), and you're currently offering it only to folks who have ColdFusion already, you're missing out on a much larger audience. By redistributing your application with BlueDragon bundled and transparent to them, you can change that.

With the free Server edition and our redistribution options, we're working to make CFML continue to be your Web application language of choice. And we're working to solve key challenges to help you make the most of it. Most of all, we want to help you grow your business and be successful.


### About the Author

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

*vince@newatlanta.com*

# Sorting Multidimensional Arrays

## Make the end users of your Web app happy PART 2

In my last article we looked at how to sort multidimensional arrays by creating a second single-dimensional array that is used as a key. The focus of this article is how to sort multidimensional arrays by creating a query object that can be sorted in the same way you would an ordered result set from a database. In truth, this article will demonstrate two methods: one for ColdFusion 5.0 and a second, far superior method using the <CFFUNCTION> tag introduced with ColdFusion MX.

The first step is to build a small multidimensional array. The example code below shows daily sales for a fruit stand:

```
<cfset Session.masDailySales = arraynew(2)>
<cfset Session.masDailySales[1][1]  = "Apples">
<cfset Session.masDailySales[1][2]  = "1">
<cfset Session.masDailySales[1][3]  = "9.95">
<cfset Session.masDailySales[1][4]  = "Michael">
<cfset Session.masDailySales[1][5]  = "Cash">

<cfset Session.masDailySales[2][1]  = "Oranges">
<cfset Session.masDailySales[2][2]  = "1">
<cfset Session.masDailySales[2][3]  = "6.95">
<cfset Session.masDailySales[2][4]  = "Joanne">
<cfset Session.masDailySales[2][5]  = "Check">

<cfset Session.masDailySales[3][1]  = "Peaches">
<cfset Session.masDailySales[3][2]  = "4">
<cfset Session.masDailySales[3][3]  = "8.95">
<cfset Session.masDailySales[3][4]  = "Michael">
<cfset Session.masDailySales[3][5]  = "Credit">
```

For each of the two new methods three variables will need to be created:

```
<cfset colArray = session.masdailysales>
<cfset colNames = "product,quantity,price,name,type">
<cfset colSort  = #sort_column#>
```

The colArray variable will be a local copy of the original array, colNames will be the field names assigned to each column of the table being created, and colSort (passed as a URL variable, named sort_column, from the column header hyperlinks described below) will be the number of the column that the completed table will be sorted on.

By Richard Gorremans

### Defining the colSort Variable

Each of the column titles is displayed to the user as a hyperlink. The hyperlinks will each call the page again and will pass the column number as sort_column. The code below shows how it will default to one (1). Each time the page is called it looks for this information and sets the sort_column URL parameter that will be used to create the colSort variable.

As an example, the title for the Quantity would be coded as:

```
<a href="MultArraySort.cfm?&sort_column=2">Quantity</a>
```

When the page is reloaded, the following code will set the column number that is to be sorted. The IsNumeric() function is used to ensure that a numeric value is used. This defaults to 1 if not defined, blank, less than zero, greater than the number of columns (5 in this example), or non-numeric.

```
<cfif IsDefined("url.sort_column") AND IsNumeric(url.sort_column) AND
   url.sort_column GT 0 and
   url.sort_column LT 5>
  <cfset sort_column = url.sort_column>
<cfelse>
  <cfset sort_column = 1>
</cfif>
```

### ColdFusion 5 Coding

Using the string colNames, an array will be created that contains the field names for each of the elements of the original array:

```
   <cfset s_order = arraynew(1)>
   <cfloop from="1" to="#ListLen(colNames)#" index="i">
                <cfset s_order[i] = ListGetAt(colNames,i)>
                </cfloop>
```

This coding will create the array to the right.

| Array | |
|---|---|
| 1 | Product |
| 2 | Quantity |
| 3 | Price |
| 4 | Name |
| 5 | Type |

The first step in querying arrays is creating a query object. Query objects are data structures that hold data, such as the information from our original array. Using the QueryNew(), QueryAddRow(), and QuerySetCell()

functions, the following code illustrates how to create a query object called tempquery:

```
<cfset tempquery = QueryNew(#colNames#)>
<cfset qRow  = QueryAddRow(tempquery, #ArrayLen(colArray)#)>
<cfloop from="1" to="#ArrayLen(colArray)#"
 index="i">
      <cfloop from="1" to="#ArrayLen(s_order)#" index="j">
                <cfset temp = QuerySetCell(tempquery,
                 "#s_order[j]#",
                 colArray[i][j], i)>
      </cfloop>
</cfloop>
```

The parameter for the QueryNew() function is the string colNames. This string contains the equivalent of field names in a database table or column headers. Next we'll add the same number of rows to the tempquery array as there are in the original array. The QueryAddRow() function is used to add these rows with tempquery and the length of the original array used as parameters.

Using a nested loop sequence and the QuerySetCell() function, the tempquery array is populated with the data from the original array, only now we have a query object:

**query**

| | NAME | PRICE | PRODUCT | QUANTITY | TYPE |
|---|---|---|---|---|---|
| 1 | Joanne | 6.95 | Oranges | 1 | Check |
| 2 | Michael | 8.95 | Peaches | 4 | Credit |
| 3 | Michael | 9.95 | Apples | 1 | Cash |

The final step is executing a <CFQUERY>, selecting from tempquery and setting the order to the column specified by the colSort variable:

```
<cfquery dbtype="query" name="sortquery">
     select * from tempquery
     order by #s_order[colSort]#
</cfquery>
```

Less code is required, execution time is slightly faster, and you get the same results as you would using the coding in my previous article. With a little experimentation this code can be easily converted to a custom tag.

## ColdFusion MX Coding

ColdFusion MX introduced <CFFUNCTION>, making it no longer necessary to use <CFSCRIPT>. With <CFFUNCTION> you can now create user-defined functions that use ColdFusion tags in their body, including <CFQUERY> and <CFSTOREDPROC>.

By changing the local variables to parameters and using the sortquery object as a return value, the following code will create a custom function that can be utilized in any ColdFusion MX application. The coding is shown below:

```
 <cffunction name="MDArraySort" Returntype="query">
<!--- Create an arguments collection for arguments passed to function
       colArray - array being sorted
       colNames - column/field names for the table being created
       colSort - which column to sort on --->
     <cfargument name="colArray" type="array" required="true">
     <cfargument name="colNames" type="string" required="true">
     <cfargument name="colSort" type="string" required="true">
     <!--- Declare variable collection used in function as local --->
     <cfset var s_order = arraynew(1)>
     <cfset var tempquery = "">
     <cfset var qRow = "">
     <cfset var temp = "">
     <!--- Create array that contains the column/field names --->
     <cfloop from="1" to="#ListLen(arguments.colNames)#" index="i">
          <cfset s_order[i] = ListGetAt(arguments.colNames,i)>
     </cfloop>
     <cfset tempquery = querynew(#arguments.colNames#)>
     <cfset qRow  = QueryAddRow(tempquery,
          #ArrayLen(arguments.colArray)#)>
      <!--- Populate the query table --->
     <cfloop from="1" to="#arraylen(arguments.colArray)#" index="i">
          <cfloop from="1" to="#arraylen(s_order)#" index="j">
               <cfset temp = QuerySetCell(tempquery, "#s_order[j]#",
                    arguments.colArray[i][j], i)>
          </cfloop>
     </cfloop>
     <!--- perform a select that sorts the new table --->
     <cfquery dbtype="query" name="sortquery">
          select * from tempquery
               order by #s_order[arguments.colSort]#
     </cfquery>
     <cfreturn sortquery>
</cffunction>
```

With the function created and made available to all applications, the following code will create a query object that can be used to display the information to the user, in the order they select.

```
<cfset t1 = MDArraySort(session.masdailysales,"product,quantity,price,
            name,type","#colOrder#")>
```

Far superior to the previously described routines, the ColdFusion MX solution is generic, and can be utilized in any ColdFusion MX application without modification.

## Conclusion

As shown in these two articles, working with arrays can add flexibility, increase the speed of a dynamic Web page by reducing the number of calls to a data source, add power to your Web application, and ultimately make the end user of your Web application very happy. In closing, remember, when old man Murphy comes to visit, always give him a warm welcome and learn from the lessons he strives to teach.




### About the Author

*For the past three years Richard Gorremans has been working for EDFUND, the nonprofit side of the Student Aid Commission, located in Rancho Cordova, California. As a senior software engineer with over 12 years in the business, he has been a technical lead, producing Web-based products that enable borrowers, lenders, and schools to view and maintain student loan information via the Web.*

*xbase@volcano.net*

# International Conference & Expo

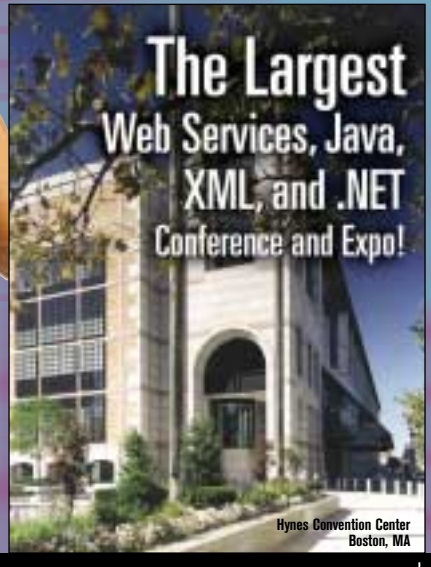
# Edge 2004 EAST

## Development Technologies Exchange

## February 24-26, 2004

## Hynes Convention Center, Boston, MA

**ARCHITECTING JAVA, .NET, WEB SERVICES, OPEN SOURCE, AND XML**




Addressing:

- Application Integration
- Desktop Java
- Mobility
- ASP.NET
- VS.NET
- JavaServer Faces
- SOA
- Interoperability
- Java Gaming

Register By November 21, 2003 SAVE Up To $400

**For more information visit**
**www.sys-con.com**
**or call**
**201 802-3069**

Over 200 participating companies will display and demonstrate over 500 developer products and solutions.

Over 3,000 Systems Integrators, System Architects, Developers, and Project Managers will attend the conference expo.

Over 100 of the latest sessions on training, certifications, seminars, case-studies, and panel discussions promise to deliver real world benefits, the industry pulse and proven strategies.

**Full Day Tutorials Targeting**

- Java • .NET • XML
- Web Services • Open Source

**Conference program available online!**
**www.sys-con.com/edgeeast2004**

**Contact information: 201 802-3069 • events@sys-con.com**

OWNED BY SYS-CON MEDIA

PRODUCED BY SYS-CON EVENTS

LinuxWorld Magazine · LinuxWorld.com · Java Developer's Journal · JavaWorld · WebServices Journal · Linux Business Week · XML Journal · OASIS · SD Times · eaindustry Consortium · PerfectXML · .NET Developer's Journal · asp.netPRO · WebSphere Developer's Journal · WebLogic Developer's Journal · wireless business technology · GE Advisor · ColdFusion Developer's Journal · ASPstreet.com · HSPstreet.com · PowerBuilder Developer's Journal · MX developer's journal

# Build a Simple Mach-II Application

## Consider this when flexibility and maintainability are important goals of your system

In the August edition of ***CFDJ***, Ben Edwards and I presented a first look at Mach-II, a new, object-oriented framework for building software applications. In this issue, I offer a tutorial (with commentary) on building a simple Mach-II application.

By Hal Helms

My choice of subjects for our Mach-II application may have been influenced by spending the last two and a half weeks in Las Vegas training students on Java, Mach-II, and Fusebox 4, culminating in the Fusebox conference. What inspired me for the application was watching otherwise intelligent people trying to outdo each other in throwing away money. Blackjack and poker, I understand; there the player does have some possible positive expectation, but craps and roulette? Watching players eagerly give up 5% or more to the house – with predictable results – I decided that the new Las Vegas slogan, "What happens here stays here," should be amended to "What you bring here stays here."

In this Mach-II application, we'll build a roulette game. While our game is admittedly not as much fun as the real thing, it has the distinct advantage that you can cheat to make yourself a winner without risking a late-night meeting with someone named "Vinny."

Let's start by quickly reviewing key Mach-II concepts. The architecture of Mach-II is known as *event-based, implicit invocation*. Each request to the framework spawns an event, an encapsulation of the request that includes such things as form and URL variables. The system then looks for software components known as *listeners* that are registered at configuration as listening for certain events. At configuration time, the system architect registers the listener, the way the listener should be called (using an *invoker*), and the method to be called. The framework itself knows nothing about what the listeners are up to, nor does any listener correspond with any other. The system is said to be *highly cohesive* (listeners have a highly focused mission) and *loosely coupled* (listeners are unaware of each other).

To determine the events the system needs, let's create a narrative that describes the roulette system. It might look something like this:

*A player logs into the system, providing his or her name and starting balance. The player is then shown a roulette table from which to select numbers to bet on and an amount to be bet on each number. When the betting is done, the wheel is spun and the dealer pays off any winning bets according to the traditional American-style odds.*

Since there are 38 numbers on an American wheel (the integers 1–36 accompanied by a zero and a double-zero), the true odds on any number being chosen are 38:1. The payoff, however, is only 35:1. The house edge is therefore 5.26%. If this doesn't seem too onerous, consider that the 5% edge is not on the *bankroll*, but on the *action*. Action is defined as the total amount risked: each new time "through" the bankroll represents that much more action. A $100 bankroll can easily generate $1000–2000 in action. Take 5% of that action and you can see why Las Vegas can afford to air-condition the desert, provide free drinks, and put on some nifty entertainment. (Several years ago, a casino executive was asked by a talk-show host what the casino did with "system players" – people who had developed a system whereby they thought they could beat the house odds. The executive's response? "We send a limo to the airport to pick them up.")

What events can we discover from our narrative? The ones I've identified are shown below in Table 1:

| Narrative | Event |
|---|---|
| Player is shown login form | newPlayer |
| Player logs into system | newPlayerRegistered |
| Player is shown the roulette table | showTable |
| Player places bet | placeBet |
| The system settles the play, paying the player according to the house odds. | settlement |

**Table 1**

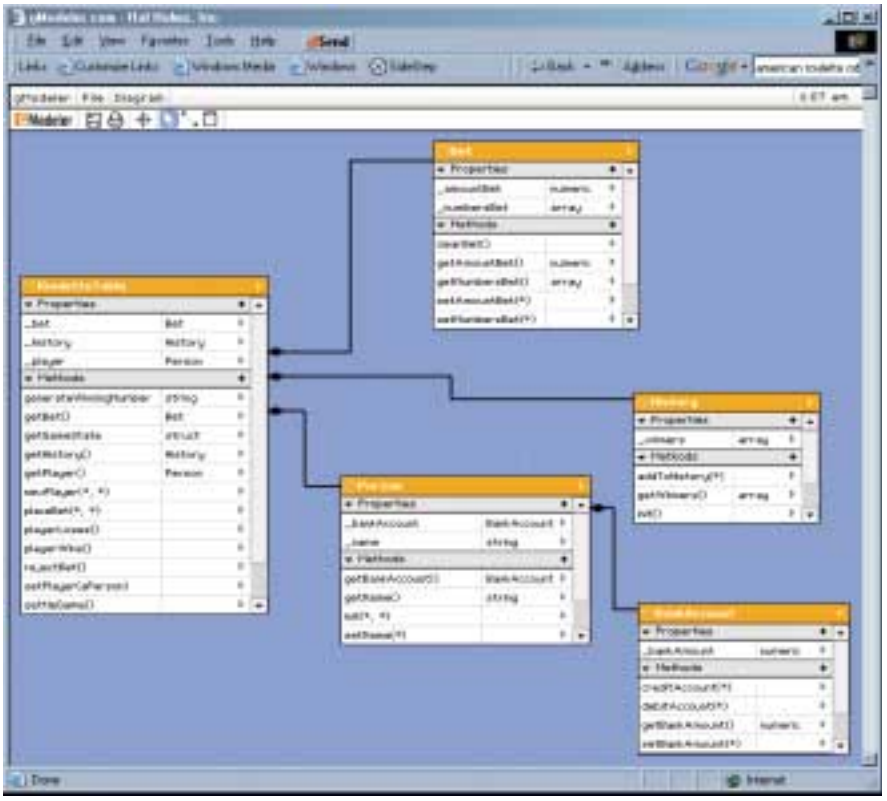


**Figure 1:** Class diagram: Roulette application


**Figure 2:** Player login form: NewPlayerForm.cfm

Why is there no event for WheelSpin, PlayerWins, or PlayerLoses? Understanding this is key to understanding event-based systems in which not all actions are events and not all components are listeners.

Consider that an object-oriented application is made up of many components that comprise an *object model*. The object model identifies the components with their associated methods that form a scale model of the system under study. The object model should be independent of the application using it.

If you're familiar with the Unified Modeling Language (UML), you'll recognize that the object model is presented in the form of a class diagram. While CFCs aren't strictly classes, they exhibit many similarities, so much so that a class diagram can be of great benefit. There are many excellent UML modeling tools available, but for our purposes we recommend the excellent Web-based gModeler created by Grant Skinner and built on Flash (www.gModeler.com).

What components and what methods do we need for roulette? Answering this question goes to the heart of how you choose to model the portion of the world under study. There are no right/wrong answers here, only better or worse judgments. The ones I've used for our sample application are shown in Figure 1.

All components are modeled as CFCs. The RouletteTable CFC is the key component; three of the other four components relate to it through the mechanism of *composition*, also known as "has-a" relationship – as in "a RouletteTable *has a* History." The other component, BankAccount, also relates to the Person CFC through composition. (For more on class relationships, see my book, *Discovering CFCs*, from techspedition.com.)

These CFCs can and should be built independently of our Mach-II application. We can create test scripts to ensure that each component is doing its job properly. For example, an extremely simple (and inadequate) test script to see that RouletteTable's generateWinningNumber() is working properly might look like this:

```
<cfset table = CreateObject('component',
'RouletteTable') />
<cfset winningNumber =
table.generateWinningNumber() />
<cfoutput>
#winningNumber#
</cfoutput>
```

When we're done, we should have a working, consistent model of a roulette game. We don't, though, have an application. For that, we turn to Mach-II. We need to find a way "into" our object model. This is accomplished by having a system component act as a listener for application events. The RouletteTable is an ideal candidate for a listener, since all other components relate directly or indirectly to it.

With our events and our listener determined, it's time to create the view pages that the player will see. There are only two: the initial player login page and the roulette table itself (see Figures 2 and 3).


**Figure 3:** Roulette table:RouletteTable.cfm

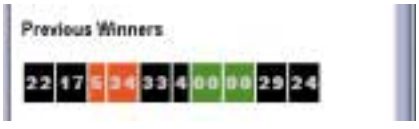



**Figure 4:** Previous winners section of RouletteTable.cfm

Since roulette players (like many gamblers) have a poor concept of odds, the house indulges their fantasy by showing previous winning numbers so that players can judge which numbers are "due." After several plays, the "Previous winners" section of the table is populated (see Figure 4).

With the scale model built and the view pages created (the code for this application can be downloaded from www.mach-ii.com), we're ready to tie everything together.

Start by creating a roulette directory structure under your webroot that looks like that shown in Figure 5.

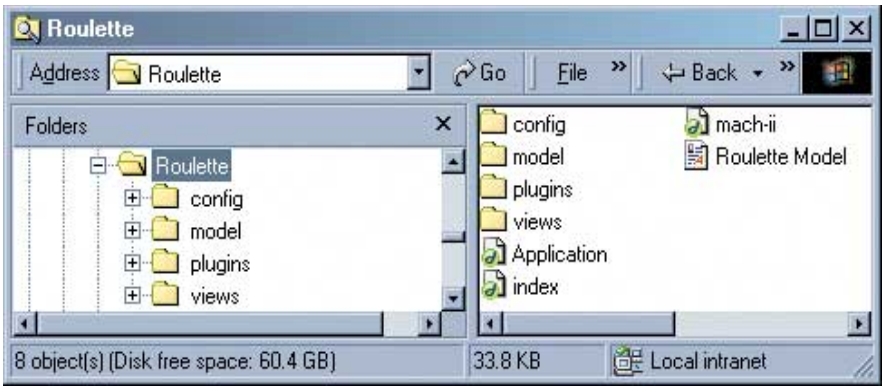



**Figure 5:** Roulette application directory structure

The **model** directory should house all of the CFCs shown in our class diagram. The **views** directory should hold the two view pages identified. If you recognize the terms, *model* and *view*, from the Model-View-Controller design pattern, you may be wondering where the **controller** directory is. In Mach-II, the framework itself is the controller.

The configuration of the system is the work of **mach-ii.xml**, found in the **config** directory. For this article, I show only the germane sections of the file dealing with events and listeners (see Listing 1).

The only listener specified is **rouletteTable**, which is invoked using a built-in Mach-II invoker, **MachII.framework.invokers.CFCInvoker_EventArgs**. While we don't have space to go into invokers, we can say that this invoker will turn all URL and form variables into a structure that is then passed to the listener.

The **<event-handlers>** section holds individual **<event-handler>** elements. Each event-handler defines an event and the listeners and/or views that are affected by it.

At runtime, player requests trigger events. The Mach-II framework reacts to these events, calling the prescribed method for each listener in the order in which each was registered, passing a reference to the event to the affected listener. Listeners can return information to the system through the **resultKey** attribute of the event-handler. This variable can then be read by the view pages.

Events and listeners can themselves announce new events. We can see that the **settlement** event announces a **showTable** event. When you examine the full code, you can find listeners (CFCs) that also announce events.

Events can be provided with event mappings. Since listeners are part of the object model and since many applications may be served by a single object model, there is no assurance that events announced by listeners will match the event handlers shown in the application. In order to provide as loose a coupling between components as possible, the event-mapping element maps events announced by listeners to events present in the application's configuration file. This sort of flexibility is a hallmark of Mach-II with its reliance on ease of application maintainability.

To fully understand the roulette application, download and study the code from www.mach-ii.com. Whether Mach-II fits your application framework needs is, of course, something you must decide. But if flexibility and maintainability are important goals of your system, Mach-II is definitely worth your consideration.


## About the Author

*Hal Helms (www.halhelms.com) is a Team Macromedia member who provides both on-site and remote training in ColdFusion, Java, and Fusebox. Hal is cofounder of the Mach-II project.*

*hal.helms@teamallaire.com*

## LISTING 1

```
<listeners>
    <listener
            name="rouletteTable"
            type="Roulette.model.RouletteTable">
            <invoker

type="MachII.framework.invokers.CFCInvoker_EventArgs" />
    </listener>
</listeners>

<event-handlers>
    <event-handler event="newPlayer">
            <view name="newPlayerForm" />
    </event-handler>

    <event-handler event="newPlayerRegistered">
            <notify
                    listener="rouletteTable"
                    method="newPlayer"
                    resultkey="" />
            <announce event="showTable" copyEventArgs="false" />
    </event-handler>

    <event-handler event="showTable">
            <notify
                    listener="rouletteTable"
                    method="getGameState"
                    resultkey="request.gameState" />
            <view name="table" />
    </event-handler>

    <event-handler event="placeBet">
            <event-mapping event="rejectedBet" mapping="showTable" />
            <event-mapping event="acceptedBet" mapping="settlement" />
            <notify
                    listener="rouletteTable"
                    method="placeBet"
                    resultkey="" />
    </event-handler>

    <event-handler event="exception">
            <view name="exception" />
    </event-handler>

    <event-handler event="settlement">
            <notify listener="rouletteTable" method="settleGame" />
            <announce event="showTable" />
    </event-handler>
</event-handlers>

<views>
    <view name="newPlayerForm" page="/views/NewPlayerForm.cfm" />
    <view name="table" page="/views/RouletteTable.cfm" />
    <view name="exception" page="/views/Exception.cfm" />
    <view name="test" page="/views/GotHere.cfm" />
</views>
```

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

# Starting a User Group

## Advantages for CF developers

The life of a Web developer can be a solitary one. Many developers work alone on projects or with a team of nondevelopers. Many work from home, or wear multiple hats, doing development work only part-time. If that describes your current work situation, then there probably aren't many opportunities to feel like you're part of a local ColdFusion community unless you're a member of an area ColdFusion User Group (CFUG).

By Brian Meloche

If you're a CF developer, regardless of your current work situation or skill level, you most likely don't know everything there is to know about CF. Even the most experienced CF guru doesn't know every Web technique and technology available – no one does. However, I am reasonably confident someone in your area knows more about something you work with (or would like to) than you do, and I doubt you know every developer in your area.

Most of us want to learn, grow, and become better and more effective CF developers. For these reasons, and many more, you owe it to yourself to join your local ColdFusion and/or Macromedia UG (MMUG). Many of you reading this are already members of your local MMUG, but how involved are you? What if there isn't a MMUG in your area? Consider starting your own. That's what I did when I recently founded the West Virginia Macromedia User Group (WVMUG).

Why form your own MMUG? There are a lot of rewards, both personal and professional, but, as with most things that are worth doing, it's not easy.

There are many things that need to be considered before getting your group up and running. Macromedia has a Web page with information about starting your own MMUG at: www.macromedia.com/cfu sion/user groups/start_a_ usergroup.cfm.

First of all, when starting a MMUG, you have to decide what kind you are going to create. There are three types:

- ***Community:*** Open to members within a region or metropolitan area
- ***Champions program:*** Open to employees at specific corporations, government agencies, and educational institutions
- ***Educational:*** Open to students of the college or high school where the group is located, and usually supervised by faculty members

You also need to decide on the product scope of your group. Some MMUGs cover ColdFusion topics only, while other groups, including the one I manage, also cover other Macromedia products. In smaller metropolitan areas, corporate, or educational groups, this is a good way to go, as the topics covered may be of interest to a wider range of Web professionals, students, and hobbyists. In larger metropolitan areas, there may be other MMUGs focusing on other Macromedia products, such as Flash. This may be beneficial when trying to partner for larger events, as well as finding other non-Macromedia UGs and on-campus organizations.

Before becoming an official Macromedia UG, you have to create the group Web site. The Web site must show the date, time, location, and agenda of the first meeting. New members should be able to join online, and you will want to set up a mailing list for the group to facilitate communication with the members. Once ready, fill out the online application to become official: www.macromedia.com/cfusion/user groups/user-group_apply.cfm. Macromedia will review your application and, if everything is ready, you'll have to sign and fax a nondisclosure agreement form (NDA) to be granted full user group status.

Be prepared for a lot of work when starting out. Many of the tasks involved in starting the group will fall to you and you alone, including marketing the group, seeking out corporate sponsors, finding and scheduling speakers, communicating with members, setting up/maintaining the group's Web site, administration, managing finances (if you decide to charge member fees), and being a liaison to Macromedia. Initially, I underestimated how much of the work would fall to me alone when starting WVMUG. However, as the group starts building a core, some members will come forward to help out. As user groups mature, many elect a board of directors to share some of the management duties.

I have always been a firm believer that the best way to learn how to do

something is to do it, which is one of the purposes of WVMUG. With this in mind, I've been offering members with little or no experience in Web development a chance to build parts of the WVMUG Web site to allow them to gain experience. For example, novice ColdFusion developers have been developing certain sections of the Web site to collect and report on membership and speaker surveys. Over time, this approach will be expanded to give more experienced ColdFusion developers chances to work on smaller projects that allow them to use less common CFML elements, Flash, and other Web development languages – using Dreamweaver MX 2004, of course!

Members are also encouraged to speak on topics on which they are knowledgeable, but that is not a prerequisite. Many MMUG managers say that some of their best meetings have occurred when the speaker didn't have a lot of expertise on the topic before preparing to present it. This gives members a chance to gain knowledge they might not get any other way. Also, MMUG meetings should not only be informative, but fun. Social activities should be encouraged, such as getting together after the meeting at a local bar or pub. This helps make newcomers to the group and those less experienced in CF, who might feel reluctant to contribute, feel more relaxed and confident.

Once you've set up the MMUG, some resources you'll have access to include:

- ***Outside speakers:*** You'll occasionally be able to bring in great speakers. Many of the regular writers in ***CFDJ***, as well as Macromedia employees and other user group managers, regularly speak at ColdFusion/Macromedia UG meetings.
- ***Group listing on Macromedia Web site:*** Your group is listed on the User Group Locator page of macromedia.com. This helps you find new members.
- ***Macromedia asset portal:*** Assets, presentations, and information help you manage your group more effectively.
- ***Monthly newsletter:*** Contains the latest updates on products, as well as Macromedia news.
- ***Macromedia user group manager's list:*** Macromedia has a mailing list that allows managers to communicate with fellow managers throughout the world. This is also the primary source of communication between Macromedia and the managers. I can attest that this list is one of the nicest perks of being a manager. You feel like your group is a part of a much larger community.
- ***Promotional items/not for resale (NFR) products/giveaways:*** When available, Macromedia sends NFR products for demonstration purposes. These cannot be used commercially. Macromedia may also send your MMUG full licensed copies of its software to use as promotional giveaways. These items come twice a year.
- ***Other promotional sponsors:*** There are sponsorship programs available with several book publishers and software companies that also offer promotional items to MMUGs.
- ***Special events at Macromedia conferences:*** For example, there will be a MMUG manager day on November 18 at Macromedia MAX 2003, taking place November 18–21 in Salt Lake City, Utah. For more information, go to: www.macromedia.com/macromedia/conference.

I have outlined some of what you have to deal with when starting your own MMUG. There's a lot of work involved. So why do it? Well, here's the secret: it doesn't feel like work. I have loved every second I have spent getting WVMUG off the ground. If you're like me, and you love Macromedia's products, what better way to spend your time than creating a community of people that share that love? If you are in an area that already has a user group, become a member, attend meetings, and become involved. If you don't have a group in your area, I strongly encourage making the effort to start one. I can tell you personally, that, after all this work, setting up WVMUG has been one of the most rewarding things I have ever done.

## About the Author

*Brian Meloche is a computer systems analyst with CDI Professional Services, currently working on contract in Nitro, WV. Brian has 11 years' experience in the IT industry, and has been doing Web development, design, and server administration for the past 7 years. He is the founder and manager of the West Virginia Macromedia Users Group (WVMUG), and a Certified Macromedia Advanced ColdFusion 5.0 Developer. You can learn more about the group by going to www.wvmug.org.*

*brian@wvmug.org*

# SYS-CON MEDIA

## 304,187 of the World's Foremost IT Professionals

### DIRECT MAIL, EMAIL OR MULTI-CHANNEL

**Target CTOs, CIOs and CXO-level IT professionals and developers who subscribe to SYS-CON Media's industry leading publications**

**Java Developer's Journal...** The leading publication aimed specifically at corporate and independent java development professionals


**LinuxWorld Magazine...** The premier monthly resource of Linux news for executives with key buying influences


**Web Services Journal...** The only Web Services magazine for CIOs, tech, marketing & product managers, VARs/ISVs, enterprise/app architects & developers


**XML-Journal...** The world's #1 leading XML resource for CEOs, CTOs, technology solution architects, product managers, programmers and developers


**PowerBuilder Developer's Journal...** The only PowerBuilder resource for corporate and independent enterprise client/server and web developers



**WebSphere Developer's Journal...** The premier publication for those who design, build, customize, deploy, or administer IBM's WebSphere suite of Web Services software



**ColdFusion Developer's Journal...** The only publication dedicated to ColdFusion web development


**WebLogic Developer's Journal...** The official magazine for BEA WebLogic application server software developers, IT management & users


**.NET Developer's Journal...** The must read iTechnology publication for Windows developers & CXO management professionals

**Wireless Business & Technology...** The wireless magazine for key corporate & engineering managers, and other executives who purchase communications products/services

Recommended for a variety of offers including Java, Internet, enterprise computing, e-business applications, training, software, hardware, data back up and storage, business applications, subscriptions, financial services, high ticket gifts and much more.

**NOW AVAILABLE!**
**The SYS-CON Media Database**
**304,187 postal addresses**

e-POSTDIRECT® Your iMarketing Source

*For email information:*
*contact Frank at 845-731-3832*
*frank.cipolla@epostdirect.com*
epostdirect.com 800-409-4443 fax845-620-9035

*For postal information:*
*contact Kevin at 845-731-2684*
*kevin.collopy@edithroman.com*
edithroman.com 800-223-2194 fax845-620-9035

List Brokerage & Management
EDITH ROMAN
Data Processing · Email Marketing

# ColdFusion User Groups

**For more information go to...**
**http://www.macromedia.com/cfusion/usergroups**

## New England

**New Hampshire**
Northern N.E. MMUG
www.mmug.info

**Massachusetts**
Boston, MA CFUG
www.cfugboston.org

**Rhode Island**
Providence, RI CFUG
www.ricfug.com

**Vermont, Montpelier**
Vermont CFUG
www.mtbytes.com/dfug/index.htm

## Midatlantic

**New Jersey, Raritan**
Central New Jersey CFUG
www.freecfm.com/c/cjcfug/index.cfm

**New York**
Albany, NY CFUG
www.anycfug.org

**New York**
Long Island, NY CFUG
www.licfug.org

**New York**
New York, NY CFUG
www.nycfug.org

**New York**
Rochester, NY CFUG
www.roch-cfug.org

**New York**
Syracuse, NY CFUG
www.cfugcny.org

**Pennsylvania, Harrisburg**
Central Penn CFUG
www.centralpenncfug.org

**Pennsylvania**
Philadelphia, PA CFUG
www.pacfug.org

**Pennsylvania**
Pittsburgh, PA CFUG
www.orbwave.com/pghcfug

**Pennsylvania**
State College, PA CFUG
www.cfug-sc.org

## Southern

**Alabama**
Birmingham, AL CFUG
www.bcfug.org

**Alabama**
Huntsville, AL CFUG
www.nacfug.com

**Delaware, Dover**
Delaware CFUG
www.decfug.org

**Delmarva, Dover**
Delmarva CFUG
www.delmarva-cfug.org

**Florida**
Gainesville, FL CFUG
http://plaza.ufl.edu/aktas

**Florida**
Orlando, FL CFUG
www.cforlando.com

**Florida**
Tallahassee, FL CFUG
www.tcfug.com

**Florida**
Tampa, FL CFUG
www.tbcfug.org

**South Florida**
South Florida CFUG
www.cfug-sfl.org

**Georgia, Atlanta**
Atlanta, GA CFUG
www.acfug.org

**Georgia, Atlanta**
Georgia CFUG
www.cfugorama.com

**Georgia**
Columbus, GA CFUG
www.vcfug.org

**Kentucky**
Louisville, KY CFUG
www.loulexcfug.com

**Louisiana**
New Orleans, LA CFUG
www.nocfug.org

**Maryland**
Annapolis, MD CFUG
www.ancfug.com

**Maryland**
Baltimore, MD CFUG
www.cfugorama.com

**Maryland**
Broadneck H. S. CFUG
www.cfug.broadneck.org

**Maryland, Bethesda**
Maryland CFUG
www.cfug-md.org

**Maryland**
California, MD CFUG
http://smdcfug.org

**North Carolina**
Charlotte, NC CFUG
www.charlotte-cfug.org

**North Carolina**
Fayetteville, NC CFUG
www.schoollink.net/fcfug/

**North Carolina**
Raleigh, NC CFUG
www.ccfug.org

**Oklahoma**
Oklahoma City, OK CFUG
http://idgweb4.ouhsc.edu/cfug

**Oklahoma**
Tulsa, OK MMUG
www.tulsacfug.org

**Tennessee**
Memphis, TN CFUG
http://cfug.dotlogix.com

**Tennessee**
Nashville, TN CFUG
www.ncfug.org

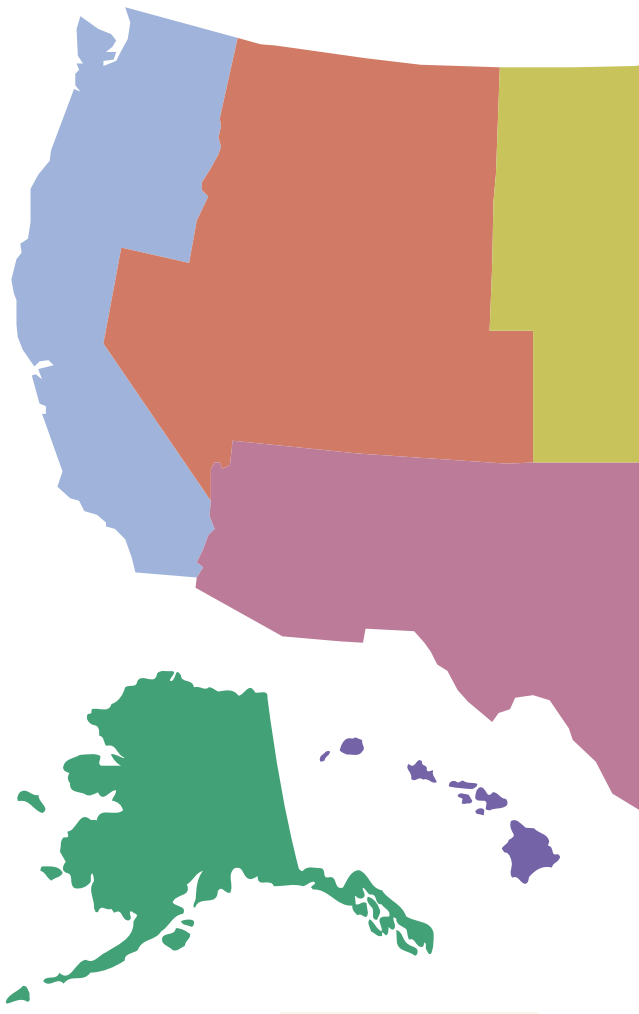
**Virginia**
Hampton Roads CFUG
www.hrcfug.org

**Virginia**
Northern Virginia CFUG
www.cfugorama.com

**Virginia**
Richmond, VA CFUG
http://richmond-cfug.btgi.net

**Virginia, Roanoke**
Blue Ridge MMUG
www.brmug.com

**Washington D.C.**
Washington, D.C. CFUG
www.cfugorama.com



## Midwest

**Northern Colorado**
Northern Colorado CFUG
www.nccfug.com

**Colorado** Hamilton M.S. CFUG
http://hamilton.dpsk12.org/teachers/team-c/intro.html

**Illinois, Champaign**
East Central Illinois CFUG
www.ecicfug.org

**Illinois, Chicago**
Chicago, IL CFUG
www.cfugorama.com

**Indiana**
Indianapolis, IN CFUG
www.hoosierfusion.com

**Indiana, South Bend**
Northern Indiana CFUG
www.ninmug.org

**Iowa**
Des Moines, IA CFUG
www.hungrycow.com/cfug/

**Michigan, Dearborn** Mmaniacs CFUG http://ciwebstudio.com/mmania/mmania.htm

**Michigan, East-Lansing**
Mid Michigan CFUG
www.coldfusion.org/pages/index.cfm

**Minnesota, Minneapolis**
Twin Cities CFUG
www.colderfusion.com

## Rockies

**Montana, Helena**
Montana CFUG
http://montanacfug.site-o-matic.com

**Nevada**
Las Vegas, NV CFUG
www.sncfug.com

**Utah**
Salt Lake City, UT CFUG
www.slcfug.org

**Wyoming, Jackson**
Wyoming MMUG
www.wycfug.org

## West Coast

**California**
Bay Area CFUG
www.bacfug.net

**California, Fresno**
Central California CFUG
www.cccfug.com

**California, Inland Empire,**
Inland Empire CFUG
www.sccfug.org

**California**
Los Angeles CFUG
www.sccfug.org

**California**
Orange County CFUG
www.sccfug.org

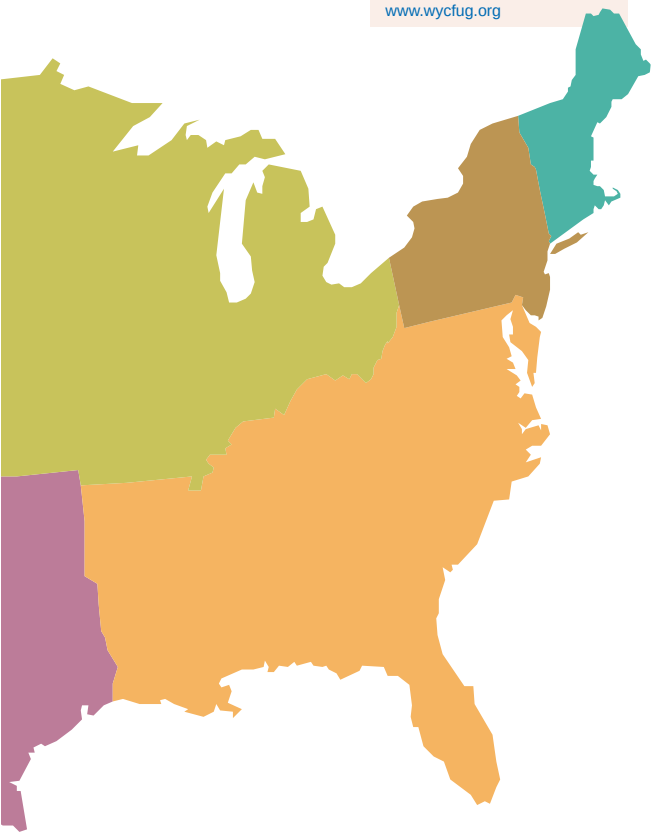
**California**
Sacramento, CA CFUG
www.saccfug.org

**California**
San Diego, CA CFUG
www.sdcfug.org

**Southern California**
Southern California CFUG
www.sccfug.org

**Oregon**
Eugene, OR CFUG
www.EugeneCFUG.org

**Oregon**
Portland, OR CFUG
www.pdxcfug.org



## Alaska

**Alaska**
Anchorage, AK CFUG
www.akcfug.org

## Hawaii

**Hawaii, Honolulu**
Hawaii CFUG
http://cfhawaii.org

**Missouri**
Kansas City, MO CFUG
www.kcfusion.org

**Missouri**
St. Louis, MO CFUG
www.psiwebstudio.com/cfug

**Nebraska**
Omaha, NE CFUG
www.necfug.com

**Ohio, Columbus**
Ohio Area CFUG
www.oacfug.org

**Ohio**
Mid Ohio Valley MMUG
www.movcfug.org

**Wisconsin**
Milwaukee, WI MMUG
www.metromilwaukee.com/usr/cfug

## Southwest

**Arizona**
Phoenix, AZ CFUG
www.azcfug.com

**Arizona**
Tucson, AZ CFUG
www.tucsoncfug.org

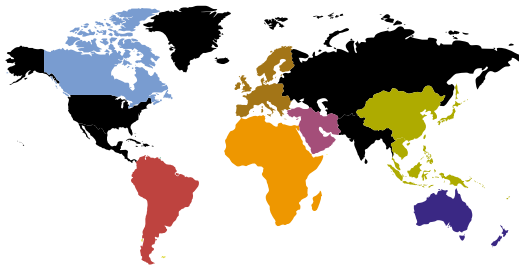
**Texas**
Austin, TX CFUG
http://cftexas.net

**Texas**
Dallas, TX CFUG
www.dfwcfug.org

**Texas**
San Antonio, TX CFUG
http://samcfug.org

## About CFUGs

ColdFusion User Groups provide a forum of support and technology to Web professionals of all levels and professions. Whether you're a designer, seasoned developer, or just starting out – ColdFusion User Groups strengthen community, increase networking, unveil the latest technology innovations, and reveal the techniques that turn novices into experts, and experts into gurus.



## Africa

**South Africa, Cape Town**
Cape Town, South Africa CFUG
www.coldfusion.org.za

**South Africa, Johannesburg**
Johannesburg, South Africa CFUG
www.coldfusion.org.za

## Asia

**Malaysia, kuala lumpur**
Malaysia CFUG
www.coldfusioneer.com

**Thailand**
Bangkok, Thailand CFUG
http://thaicfug.tei.or.th

**Japan, Urayasu-city** Japan CFUG
http://cfusion.itfrontier.co.jp/jcfug/jcfug.cfm

## Australia

**Australia-Melbourne**
Australian CFUGs
www.cfug.org.au

## Canada

**Canada**
Edmonton, AB CFUG
http://edmonton.cfug.ca

**Canada**
Kingston, ON CFUG
www.kcfug.org

**Canada**
Montreal, QC CFUG
www.kopanas.com/cfugmontreal

**Canada**
Ottawa, ON CFUG
www.cfugottawa.com

**Canada, Ottawa (HS group)**
Ecole Secondary CFUG
www.escgarneau.com/gug

**Canada**
Toronto, ON CFUG
www.cfugtoronto.org

**Canada**
Vancouver, BC CFUG
www.cfug-vancouver.com

**Canada**
Vancouver Island CFUG
www.cfug-vancouverisland.com

## Europe

**Belgium, Brussels**
Belgium CFUG
www.cfug.be

**Central Europe, Munich**
Central Europe CFUG
www.cfug.de

**Finland, Helsinki**
Finland CFUG
www.cfug-fi.org

**France, Valbonne**
France CFUG
www.cfug.fr.st

**Germany, Frankfurt**
Frankfurt CFUG
www.cfug-frankfurt.de

**Ireland**
Belfast, Ireland CFUG
www.cfug.ie

**Ireland**
Cork, Ireland CFUG
http://viewmylist.com/cork

**Italy, Bologna**
Italy CFUG
www.ingenium-mmug.org

**Sweden**
Gothenburg, Sweden CFUG
www.cfug-se.org

**Switzerland**
Martin Bürlimann, Switzerland CFUG
www.cfug.ch

**United Kingdom, London**
UK CFUG
www.ukcfug.org

**United Kingdom**
Northern England CFUG
www.cfug.org.uk

## Middle East

**Pakistan Edu, Lahore Cantt**
Pakistan Educational CFUG
www.cfeugpakistan.org

**Pakistan, Lahore**
Pakistan CFUG
www.cfugpakistan.org

**Saudi Arabia, Riyadh**
CFUG Saudia
www.cfugsaudia.org

**Turkey, Izmer**
Turkey CFUG
www.cftr.net

## South America

**Brazil**
Rio de Janerio CFUG
www.cfugrio.com.br

### SYS-CON Media Announces *MX Developer's Journal*

(*Montvale, NJ*) – SYS-CON Media, the world's leading *i*-technology publisher, has released a special preview issue of its latest highly anticipated new title, *MX Developer's Journal*. The publication debuted in conjunction with Macromedia's release of their MX 2004 product line, featuring new versions of Dreamweaver, Flash, and Fireworks, as well as the new Flash MX Professional 2004, which provides additional features for advanced Flash developers.


The premier issue will be available at Macromedia MAX 2003, Macromedia's user conference (see following news item).

Each issue of *MXDJ* will include:

- Hands-on tutorials covering every aspect of Flash MX, Dreamweaver MX, Fireworks MX, FreeHand MX, ColdFusion MX, and DirectorMX
- Full and constantly updated information on the overall Macromedia MX architecture
- Experience Reports on leading sites that already use Rich Internet Apps (RIA) to engage customers
- Insightful technical articles and feature stories on how best to harness the MX family of technologies
- Success stories – showcasing Fortune 500 companies already using Macromedia MX to deliver RIA
- Product reviews and sneak previews of upcoming releases within the Macromedia family
- How-to programming tips/code listings, and more...

A special U.S. charter subscription rate is being offered to *CFDJ* readers, with substantial savings off newsstand prices. For more information, see www.sys-con.com/mx/charter.cfm, or call 1 (888) 303-5282.

### Announcing MAX: the 2003 Macromedia User Conference

(*San Francisco*) – MAX, a new conference that combines DevCon and UCON, is Macromedia's annual professional conference for Macromedia developers and designers. It's scheduled for November 18–21 at the Salt Palace Convention Center in Salt Lake City, Utah.

This year, the Macromedia conference has a new name and a new venue, reflecting an exciting evolution for the conference and the company's customers. The name change is designed to express their goal – that MAX appeals to the broadest range of Macromedia customers. Stay tuned for details. www.macromedia.com/macromedia/conference

### Macromedia Delivers Public Beta Version of Macromedia Central

(*San Francisco*) – On "One Unwired Day," sponsored by Intel, Macromedia, Inc., unveiled a public beta release of Macromedia Central, providing a new way for people to interact with Internet information without relying on a continuous connection. Macromedia made the public beta available on One Unwired Day to showcase the possibilities of life without wires. For more information on Macromedia Central, and to download the public beta, go to www.macromedia.com/go/central/.

Central delivers a new kind of Internet experience, bringing the information you're most interested in directly to you. It offers a clean, unified place for Internet applications, including a customizable console for your most important information and a host of other features that help you save time and work faster.

The Central public beta includes two initial beta applications: Movie Finder and AccuWeather. These applications will be part of a larger application library being created by third parties for Central. Since the start of the initial beta program this summer, close to 1,000 developers have joined the application development community for Central.

## cf community *—continued from page 7*

ColdFusion can take advantage of resources already built into the J2EE Server (Java), that it will make separating business logic from the display layer easy as opposed to countering it, etc. His e-mail made it more clear that what Mario needed to do was not to show ColdFusion as an alternative to or replacement for Java, but as an enhancement to it. I suggested he state something like the following to his boss:

*ColdFusion is a J2EE application that sits on top of a J2EE Server along with any other J2EE application deployed on the server. ColdFusion Markup Language, the scripting language interpreted by the ColdFusion Application Server application, is just an alternative method for developing Java applications. It's easier to debug, faster to develop with, offers out-of-the-box support for a variety of functional engines such as the Verity text-search engine, a charting and graphing engine, etc. Learning curves and development times are shorter with CFML than with Java or JSP, and developers with Java knowledge can easily integrate applications written in CFML with existing Java applications or functionality. CFML offers the same architectural benefits as Java and other object oriented languages and it tends to be easier to integrate with a display tier than with these other languages. Bottom line, it is the ideal method for developing Java applications.*

This last sentence really sums up what ColdFusion is: "the ideal method for developing Java applications." No, it doesn't require knowledge of Java, but it does create Java byte code behind the scenes, and developers familiar with Java can take full advantage of its features in their ColdFusion applications. This is the ideal way to "pitch" ColdFusion MX, if not to everybody, certainly to companies and individuals who are either pro-Java or who just like buzzwords (you can throw in a "Web services this" and an "XML that" for these people while you're at it – and don't forget "cross platform," too). It's important that developers begin getting more comfortable and familiar with this way of explaining ColdFusion, as ColdFusion MX sometimes requires a very different "sales approach" than prior versions did.

Obviously, all of the great selling points of ColdFusion in the past still hold true, and should also be pointed out when convincing people to purchase or upgrade their servers. In order for us as developers to be successful, we must make the product successful. This can only happen if you take it upon yourself to learn the new features each time a new release is made available, and continue to persuade companies to use the product by showing your enthusiasm and educating them about just how cool ColdFusion is and continues to become.

Mobile Entertainment WiFi 3G Handheld Computing Wireless Wearables Enterprise Applications WISP

for over seven years CTIA has fostered and guided the wireless developer community. Today, the CTIA WIRELESS I.T. & ENTERTAINMENT 2003 show has become the one place where developers for all platforms can meet to learn, exchange ideas and form crucial partnerships.

# your 0000 0001 to success

Bluetooth WLAN Consumer Applications 802.11

# CTIA WIRELESS I.T.® & ENTERTAINMENT 2003

Come see what is new in wireless applications at CTIA WIRELESS I.T. & ENTERTAINMENT 2003 and forge ahead in the NEW wireless data industry.

**Exhibit Show Floor Showcasing all of the Major Leaders in Wireless Data**

**Comprehensive Educational Programming**

- Executive Smart Pass Program on WiFi
- 16 in-depth sessions dedicated to developments in Mobile Entertainment and Enterprise Solutions

**Key Networking Receptions**

- Exhibit Hall Preview
- Invite-only Mobile Entertainment Reception

**Developer Sessions**

- AT&T Wireless Business Partner Forum
- The Wireless LBS Challenge
- UIQ Development for Sony Ericsson Smartphones

The NEW™ WIRELESS DATA INDUSTRY

## OCTOBER 21–23, 2003

Sands® EXPO and CONVENTION, Venetian Hotel, Las Vegas, NV, USA

**www.ctiashow.com**

**OCTOBER 20, 2003** Pre-conference Seminars **OCTOBER 21, 2003** CTIA Educational Sessions, Special Interest Seminars & Exhibit Hall Preview Reception **OCTOBER 22-23, 2003** Exhibits, CTIA Educational Sessions [10/22 only] & Special Interest Seminars

DAY ONE

**Lucy Hood,** *SVP, Content,* **News Corporation**

**Kent Thexton,** *Chief Data & Marketing Officer,* **mmO2 PLC**

**Nada C. Usina,** *GM of Entertainment & Media, North & South America,* **Nokia**

DAY TWO

**Brian Bonner,** *VP & CIO,* **Texas Instruments Incorporated**

**Tony Scott,** *CTO,* **General Motors Information Systems & Services Organization**

**John V. Dolan,** *VP, Wireless & Voice Product Strategy* **Oracle Corp.**

**Dr. Franz Fink** *VP & GM, Wireless & Mobile Systems Group, Semiconductor Products Sector (SPS)* **Motorola, Inc.**

DAY THREE

**Juha Christensen** *Corp. VP, Mobile Devices Marketing Group* **Microsoft Corporation**

**Reed Hundt,** *Former Chairman* **FCC**

**Sean Maloney,** *EVP, GM, Intel Communications Group,* **Intel Corporation**

*Host*

CTIA

*Partners*

*Sponsors*