ADVERTISERS INDEX

ADVERTISER	URL	PH	PG
ABLECOMMERCE	WWW.ABLECOMMERCE.COM/CFDJ/THANKS.CFM	360.253.4142	2-3
AFFINITY	WWW.AFFINITY.COM/RESELLER	800.646.0662	33
AIR2WEB	WWW.AIR2WEB.COM	877.238.3637	45
ALLAIRE CORPORATION	WWW.VUE.COM/ALLAIRE	877.460.8679	43
ALLAIRE CORPORATION	WWW.ALLAIRE.COM/USERGROUPS		47
ALLAIRE CORPORATION	WWW.ALLAIRE.COM	888.939.2545	49
ANDREWS TECHNOLOGY	WWW.ANDREWSTECHNOLOGY.COM	888.393.0159	29
BIZNIZWEB,INC	WWW.BIZNIZWEB.COM	281.367.4016	56
CF HOSTING	WWW.CFHOSTING.COM	770.982.8382	55
CFDJ CAREER CENTER	WWW.COLDFUSIONJOURNAL.COM		65
CFXHOSTING	WWW.CFXHOSTING.COM	866.CFXHOST	31
COLDFUSIONDB.COM	WWW.COLDFUSIONDB.COM/CFDJ11		54
CONCEPTWARE	WWW.CONCEPTWARE.COM		25
CORDA TECHNOLOGIES	WWW.POPCHART.COM	801.802.0800	35
CYBERSMARTS	WWW.CYBERSMARTS.NET		62
CYSCAPE	WWW.CYSCAPE.COM/BHFREE	800.932.6869	54
EVOLUTION B	WWW.EVOLUTIONB.COM	877.662.7551	71
FUSION AUTHORITY	WWW.FUSIONAUTHORITY.COM	718.252.2329	13
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM	877.215.HOST	68
INTELIANT	WWW.INTELIANT.COM	800-815.5541	21
INTELLIGENT ENVIRONMENTS	WWW.SCREENSURFER.COM	781.932.1700	11
INTERLAND	WWW.INTERLAND.COM	877.887.3864	9
INTERMEDIA	WWW.INTERMEDIA.NET	800.379.7729	72
INTERNET AUTOMATED	WWW.IAUTOMATED.COM		63
JAVADEVELOPERSJOURNAL.COM	WWW.JAVADEVELOPERSJOURNAL.COM	201.802.3020	70
JDJ STORE	WWW.JDJSTORE.COM	888.303.JAVA	64
MACROMEDIA	WWW.MACROMEDIA.COM/ULTRADEV	415.252.2000	36-37
MISTRAL DESIGN GROUP	WWW.PURPLEHOSTING.COM		57
NAVISITE	WWW.NAVISITE.COM	888.298.8222	59
NETMORF	WWW.NETMORF.COM	877.533.MORF	4
PAPERTHIN	WWW.PAPERTHIN.COM	800.940.3087	27
RSW SOFTWARE	WWW.RSWSOFTWARE.COM	781.993.8500	15
SITEHOSTING.NET	WWW.WEARECFHOSTING.COM	877.NTHOSTING	19
SYS-CON MEDIA INC.	WWW.SYS-CON.COM	800.513.7111	70
THESHORTLIST.COM	WWW.SHORTLIST.COM		18,39
VERIO	WWW.DEDICATEDSERVER.COM	877.399.0589	17
VIRTUALSCAPE	WWW.VIRTUALSCAPE.COM	877.VSCAPE4	23
WIRELESS BUSINESS & TECH	WWW.WIRELESS-MAGAZINE.COM.		65
WIRELESS DEVCON	WWW.WIRELESSDEVCON2000.COM		50-51
WOODBOURNE SOLUTIONS	WWW.WOODBOURNESOLUTIONS.COM	301.428.7620	56
XML DEVCON 2000-2001	WWW.XMLDEVCON2000.COM	800.513.7111	61

Please visit the Web sites of our advertising partners who make it possible for us to bring you this Digital Edition (PDF) of *CFDJ*

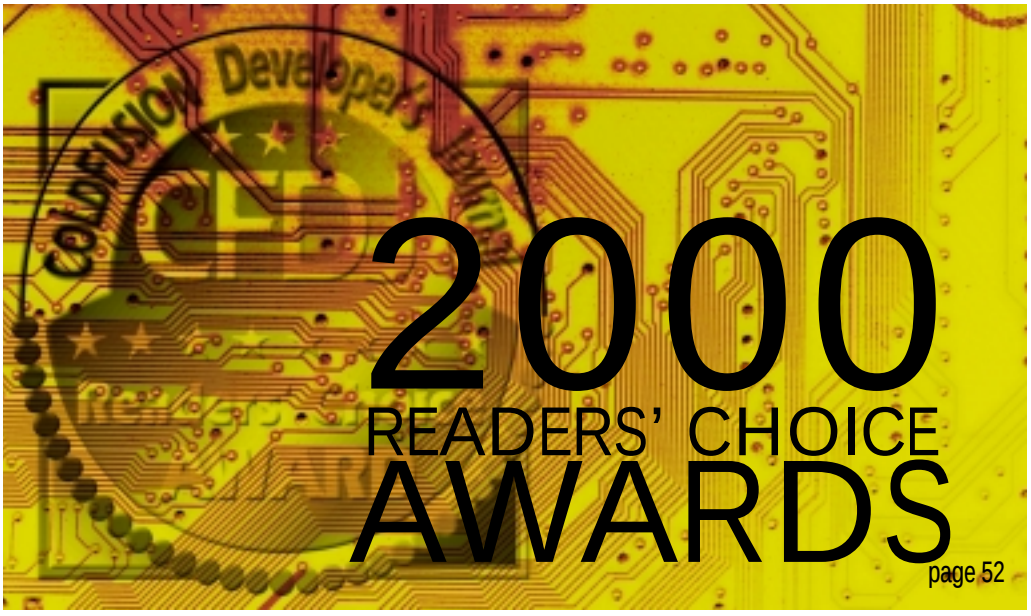
U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

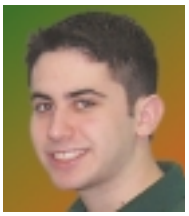
ColdFusionJournal.com

November 2000 Volume: 2 Issue: 11

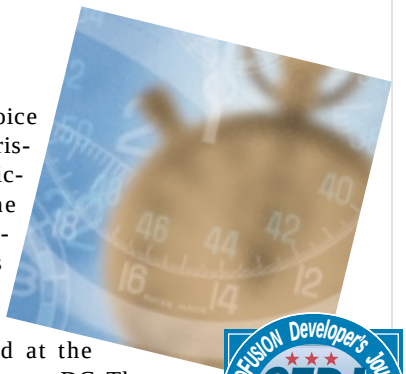
SYS-CON MEDIA



Readers' Choice Awards: The Results Are In



BY **ROBERT DIAMOND**



The first annual ***CFDJ*** Readers' Choice Awards are filled with lots of surprises. Overall, they paint a clear picture of the industry and what you – the reader – like and dislike. Modeled after the extremely successful ***Java Developer's Journal*** awards (one of ***CFDJ***'s sister publications), the established model proved to work quite well here. (How could it not? The whole system runs off ColdFusion!) Awards are being presented at the Allaire Developer Conference, November 5–8, in Washington, DC. They are presented in 13 categories: Best Book, Best Consulting Service, Best Custom Tag, Best Database Tool, Best Design Service, Best E-Business Software, Best Education and Training, Best Testing Tool, Best Web Development Tool, Best Web Hosting, Best Web Site, Best Web Application, and Most Innovative CF Application. For full coverage of the awards ceremony, including photos of the presentation, please visit www.coldfusionjournal.com.

Over 13,000 votes were logged – quite an impressive number, and many more than our initial estimates of the turnout. The number was cut down after our auditing process, however, as hundreds of duplicates and other bad votes were removed from the voting tables. Every vote went through an extensive auditing process – way too long and boring to get into here – and what we were left with was what I can confidently call the best set of results we could have hoped for. Knowing how closely companies were watching the voting process, the auditing was taken quite seriously, and every possible method was used to get a clean set of results. Throughout the contest, the leaders in several categories changed; some latecomers rose to the top, and a few companies even went out of business and asked to be removed! It was quite exciting for those following the awards day in and day out to watch the fluctuations in the live results. But enough of me rambling on about how great the results were – check them out yourself. (See page 48 for complete coverage including profiles of the winning companies.)

With this year's awards wrapped up, plans are already underway for next year's. The nominating process starts in a few months. If you have suggestions on how to improve the technical end or the user experience, please drop me a line at Robert@sys-con.com; we're always looking for new ideas and suggestions. Also, if you believe any categories were overlooked, let us know and we'll look into adding them. We've already got several new and exciting ideas of our own.

For those of you using XML along with ColdFusion on your development projects, I invite you to take a look at our first Readers' Choice Awards for ***XML-Journal***. You can access those at www.xml-journal.com/xml/readerschoice. By the time you read this, the awards will already be underway. While visiting www.xml-journal.com, you can also register for free access to the XML archives, a great resource for those working in or learning XML.

Till next month...

ABOUT THE AUTHOR

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

ROBERT@SYS-CON.COM

Instant Messaging for Your Apps

Using the session tracker in real-life implementations

BY CHRISTIAN SCHNEIDER

After my first article in **CFDJ** *(Vol. 2, issue 8) – "Live Monitoring of User Sessions" – I had some new ideas about using the session-tracking framework to implement a sample instant messaging application.*

By using the information of who is currently online with open sessions, I can provide users with an instant messaging system that enables them to send small text messages to each other while browsing the site. This would be a nice feature for community-based sites; for example, a site that offers books can enhance browsing with the add-on feature of instant messaging, which enables clients to assign themselves nicknames and send small instant messages to each other to discuss certain books. Sounds good, I thought, and implemented a prototype based on the live session monitor from my previous article.

How It Looks

As you can see from Figures 1 and 2, the screen is divided into a mainframe in which customers can browse the site (I use simple sample documents here), a lower frame in which the current list of logged-in users is visible (and autorefreshed twice every minute), and a text field to send an instant message to the users you've checked. This kind of implementation is just a sample I created for demonstration purposes. While browsing the site's content, users can send off instant messages (see Figures 2 and 3) that are delivered as popup boxes. Once delivered, the receiver can click Cancel to close the message or OK to reply to the sender (see Figure 4).

Since popup messages might be annoying to those users who logged in but didn't want to receive them, the implementation could easily be changed to present them as plain text in the lower frame. When implementing these features, clearly state to users that when they log in the instant messaging app, they may receive instant messages.

This article presents the implementation of the idea as well as the technical details behind it. When implementing this in the real world, other criteria should also be considered, such as providing users with the ability to pause instant messaging (temporarily going off the list) and to decide whether they'd like to receive the messages as popup boxes or plain text. As I only implemented this idea to demonstrate how to use the session-tracker framework, I'd greatly appreciate any feedback on design and implementation considerations from developers who bring it to reality.

How It Works

Let's look at how it was created. We have eight files; however, at least two shouldn't arouse any interest (Listings 1 and 2, index.cfm and welcome.cfm, respectively) because they only start the frameset and present some dummy data to browse.

The files login.cfm. (see Listing 3) and especially makeLogin.cfm (see Listing 4) let users set their nicknames, then insert the mappings of their Internet protocol (IP) addresses to these nicknames

into the application-scoped nickname structure. This structure is necessary because it references all users by their names, not only their IPs. It's indexed by the IP address with the user's nickname as the value. *Note:* This could also have been coded using cookies to reference each user's nickname, but I used a server-side structure to keep things simple.

ListUsers.cfm (see Listing 5) is similar to the session report in my previous session-tracker article. It just loops over the session-tracker structure and lists all online users (with their nicknames derived from the nickname structure) along with a checkbox to check the message recipients. Beneath that listing, a text box is shown with a submit button that enables users to send off their messages. Using the checkboxes to check the receivers lets the user send an instant message to multiple recipients.

The really interesting stuff happens in the CF-specific files, Application.cfm (see Listing 6) and OnRequestEnd.cfm (see Listing 7); these are executed by the ColdFusion server before and after any request. Each logged-in user is stored in the application-scoped session-tracker structure with his or her IP address and the current timestamp to check for timed-out users. (See my previous article for more details.) OnRequestEnd.cfm is especially interesting since it also checks the application-scoped message queue at the end of each request for messages for the current user. If any messages are found in the current user's queue, they're displayed with the ability to send a direct reply. This message queue consists of a simple application-scoped structure that's indexed by users' IP addresses. This structure holds a small array that acts as the user's queue for each user (Figure 5 shows the architecture). By looping over that array, all message packages for the current user are extracted and displayed. A message package consists of a mini-structure that holds the sender's name and IP address (to reply), as well as the message text itself.

The interesting thing about displaying the messages may be the direct-reply feature, which is realized client-sided using JavaScript. The delivered message is shown as a confirm box in which the user can click OK or Cancel.

```
// Present message
doReply = confirm("Message from #tmpPack-
age["from"]#:\n\n#tmpPackage["text"]#\n\n-
Reply?");
```

Using this information, the script determines whether to ask the user for a replytext (using a JavaScript prompt box).

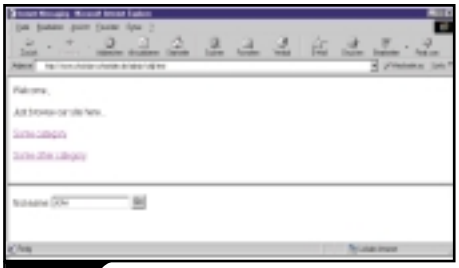


FIGURE 1 Start with nickname login

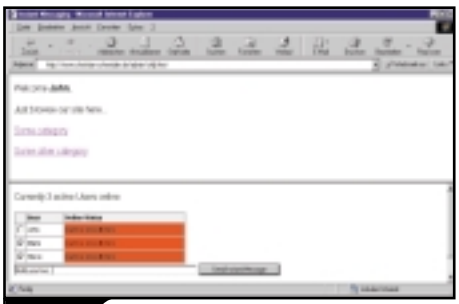


FIGURE 2 List of current online users

FIGURE 3 Receiving an instant message

FIGURE 4 The direct reply feature

```
if (doReply) {
  // Prompt for reply
  message = prompt("Your reply to #tmpPack-
age["from"]#:","");
  if ((message != "") && (message != null))
{
    // Send reply message

parent.frames.frameBottom.location.href="send
Message.cfm?message="+escape(message)+"&users=
#URLEncodedFormat(tmpPackage["sender"])#";
  }
```

If the user wishes to reply to a message and enters a short reply text, it's sent off unseen by invoking the sendMessage.cfm template in the lower frame (using JavaScript) with masked URL parameters that contain the message and its receiver.

SendMessage.cfm (see Listing 8) "sends" the message that's either submitted by the form in listUsers.cfm or handed over as URL parameters by the direct reply feature:

```
  <cfparam name="URL.from" default=#tmp-
Nickname#>
  <cfparam name="URL.sender"
default=#CGI.REMOTE_ADDR#>
  <cfparam name="URL.message" default="">
  <cfparam name="URL.users" default="">

  <cfparam name="FORM.from"
default=#URL.from#>
  <cfparam name="FORM.sender"
default=#URL.sender#>
  <cfparam name="FORM.message"
default=#URL.message#>
  <cfparam name="FORM.users"
default=#URL.users#>
```

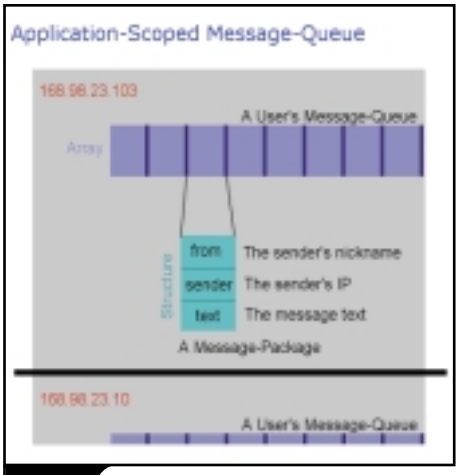


FIGURE 5: Diagram showing the message queue architecture

With these statements, the sendMail.cfm template can use FORM.parameters as well as URL.parameters to send off messages. As you may have already guessed, messages are simply sent in sendMessage.cfm by appending the message details (sender and text) as a message package at the receiver's queue using ArrayAppend( ). Remember that this application-scoped queue is checked for new messages in OnRequestEnd.cfm.

Final Thoughts

This article demonstrates how to use the session tracker in real-life implementations and discusses some of the interesting techniques behind it. As this article resulted from the idea of using the session tracker, I'd appreciate feedback on this topic. To fully understand this article, it's necessary to read my previous one about live monitoring user sessions. It can also be found online at www.sys-con.com/coldfusion/archives/0208/Schneider.

Additional ideas regarding the session tracker include extending this instant-messaging system into a chat system that could easily be developed based upon the concepts in this article.

About the Author

Christian Schneider is a ColdFusion developer with over three years of intensive experience in developing CF-based intranet applications for banks and logistic corporations.

mail@christian-schneider.de

Listing 1: index.cfm

```
<head>
 <title>Instant Messaging</title>
 <frameset rows="*,170">
  <frame name="frameMain" src="welcome.cfm">
  <frame name="frameBottom" src="Login.cfm">
 </frameset>
</head>
```

Listing 2: welcome.cfm

```
<!--- /// welcome.cfm /// --->

<cfset tmpNickname = "">
<cfif IsDefined("SESSION.nickname")>
 <cflock name="#SESSION.sessionID#" type="ReadOnly" time-
out="20" throwontimeout="Yes">
  <cfset tmpNickname = SESSION.nickname>
 </cflock>
</cfif>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">

<html>
<head>
 <title>Welcome</title>
</head>

<body><basefont face="Arial">

<cfoutput>
 Welcome <b>#tmpNickname#</b>,<p>
 Just browse our site here...
 <p>
 <a href="something_a.cfm">Some category</a>
 <p>
 <a href="something_b.cfm">Some other category</a>
</cfoutput>

</body>
</html>
```

Listing 3: Login.cfm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
 <title>Login</title>
</head>

<body><basefont face="Arial">

<form name="formLogin" action="makeLogin.cfm" method="post">
 Nickname: <input type="Text" name="nickname">
 <input type="Submit" value="Go">
</form>

</body>
</html>
```

Listing 4: makeLogin.cfm

```
<cflock name="#APPLICATION.applicationName#" type="Exclusive"
timeout="20" throwontimeout="Yes">
 <!--- Set up Nickname-Structure for mapping IPs to Nick-
names --->
 <cfparam name="APPLICATION.NickNames" default=#StructNew()#>
 <!--- Insert submitted Nickname into Nickname-Structure --->
 <cfset dummy = StructInsert(APPLICATION.NickNames,
CGI.REMOTE_ADDR, FORM.nickname, true)>
</cflock>

<cflock name="#SESSION.sessionID#" type="Exclusive" time-
out="20" throwontimeout="Yes">
 <!--- Insert submitted Nickname into Session-Scope (just for
remembering the user's nickname) --->
 <cfset SESSION.nickname = FORM.nickname>
</cflock>

<cflocation url="listUsers.cfm">
```

Listing 5: listUsers.cfm

```
<cflock name="#SESSION.sessionID#" type="ReadOnly" time-
out="20" throwontimeout="Yes">
 <cfset tmpNickname = SESSION.nickname>
</cflock>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">

<html>
<head>
 <title>Currently Active Users</title>
 <style type="text/css">
  body {
   font-family: Arial, Helvetica, Verdana;
  }
  td {
   font-size: 8pt;
  }
```

```
  input {
   font-size: 9pt;
  }
 </style>
</head>

<body onLoad="window.setTimeout('location.reload()',
30000);">

<cfoutput>

Currently #StructCount(APPLICATION.SessionTracker)# active
Users online

<cflock name="#APPLICATION.applicationName#" type="Exclusive"
timeout="20" throwontimeout="Yes">

 <form name="formUsers" action="sendMessage.cfm"
method="post">
  <input type="Hidden" name="from" value="#tmpNickname#">
  <input type="Hidden" name="sender"
value="#CGI.REMOTE_ADDR#">

  <table border="1" cellspacing="0" cellpadding="1"
width="40%">
  <tr bgcolor="##eeeeee">
   <td width="20"> </td>
   <td width="*"><b>User</b></td>
   <td width="*"><b>Online-Status</b></td>
  </tr>

  <!--- Loop through Session-Tracker and list current online
users --->
  <CFLOOP collection=#APPLICATION.SessionTracker#
item="aUser">
   <cfset onlineSince = StructFind(APPLICATION.SessionTrack-
er, aUser)>

   <!--- Check for time-outs --->
   <CFIF DateCompare(onlineSince+theTimeout, Now()) EQ 1>
    <!--- User's last activity lies within session-timeout,
so his session is active --->
    <cfset inactiveSince = DateDiff("n", onlineSince, Now())>
    <!--- The threshold for coloring the report may be set
individually --->
    <cfif inactiveSince LTE 2>
     <cfset theColor = "Red">
    <cfelseif inactiveSince LTE 5>
     <cfset theColor = "Yellow">
    <cfelse>
     <cfset theColor = "Cyan">
    </cfif>
    <!--- Output of current user in list --->
    <tr>
     <td><input type="Checkbox" name="Users"
value="#aUser#"></td>
     <td><cfif
StructKeyExists(APPLICATION.NickNames,aUser)>#StructFind(AP-
PLICATION.NickNames, aUser)#<cfelse>#aUser#</cfif></td>
     <td bgcolor="#theColor#">inactive since <b>#inac-
tiveSince#</b> mins</td>
    </tr>
   <CFELSE>
    <!--- User's session has timed-out, so we can delete his
IP from the Session-Tracker and Nickname-Structure, as well
as the Message-Queue --->
    <cfscript>
     StructDelete(APPLICATION.SessionTracker, aUser);
     StructDelete(APPLICATION.NickNames, aUser);
     StructDelete(APPLICATION.MessageQueue, aUser);
    </cfscript>
   </CFIF>
  </CFLOOP>

 </table>
  <input type="Text" name="message" size="70"
maxlength="200">
  <input type="Submit" value="Send Instant Message">
 </form>

</cflock>

<p>
<a href="welcome.cfm">Go back</a>

</cfoutput>
</body>
</html>
```

Listing 6: Application.cfm

```
<!--- Set this Variable in local scope: can be accessed from
the report-page directly --->
<cfset theTimeout = CreateTimespan(0,0,15,0)>

<!--- Set Application name and options --->
<CFAPPLICATION  NAME="My_Test_App"
    CLIENTMANAGEMENT="YES"
    SESSIONMANAGEMENT="YES"
    SESSIONTIMEOUT=#theTimeout#>
```

Listing 7: OnRequestEnd.cfm

```
<!--- Session-Tracker Code --->

<cflock name="#APPLICATION.applicationName#" type="Exclusive"
timeout="20" throwontimeout="Yes">

  <!--- If Session-Tracker does not exist, generate it... --
->
 <cfparam name="APPLICATION.SessionTracker" default=#Struct-
New()#>

 <!--- Instead of using the clients IP-address we can also
log the client's user-name
    or similar session-scoped login info here --->
 <!--- It's important to specify TRUE here to allow the
overwriting of existing entries (= updating) --->
 <CFSET dummy = StructInsert(APPLICATION.SessionTracker,
CGI.REMOTE_ADDR, Now(), true)>

 <!--- ***** Message-Queue Code ***** --->

 <cfif
CompareNoCase(GetFileFromPath(CGI.SCRIPT_NAME),"sendMessage.
cfm") NEQ 0>

  <!--- If Message-Queue does not exist, generate it... --->
  <cfparam name="APPLICATION.MessageQueue" default=#Struct-
New()#>

  <!--- Check Message-Queue for messages for current user --
->
  <cfif StructKeyExists(APPLICATION.MessageQueue,
CGI.REMOTE_ADDR)>
   <cfoutput>
    <!--- Fetch current user's queue --->
    <cfset tmpQueue = StructFind(APPLICATION.MessageQueue,
CGI.REMOTE_ADDR)>
    <!--- Loop through messages in that queue and display
each message --->
    <cfloop index="i" from=1 to=#ArrayLen(tmpQueue)#>
     <cfset tmpPackage = tmpQueue[i]>
     <SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
               <!--
          // Present message
                    doReply = confirm("Message from #tmpPack-
age["from"]#:\n\n#tmpPackage["text"]#\n\nReply?");
          if (doReply) {
                    // Prompt for reply
                    message = prompt("Your reply to #tmpPack-
age["from"]#:","");
                    if ((message != "") && (message != null)) {
                              // Send reply message
                              parent.frames.frameBottom.loca-
tion.href="sendMessage.cfm?message="+escape(message)+"&users=
#URLEncodedFormat(tmpPackage["sender"])#";
                    }
          }
                //-->
                </SCRIPT>
    </cfloop>
    <!--- Delete message from the queue --->
    <cfset dummy =
ArrayClear(APPLICATION.MessageQueue[CGI.REMOTE_ADDR])>
   </cfoutput>
  </cfif>

 </cfif>

</cflock>
```

Listing 8: sendMessage.cfm

```
<cfset tmpNickname = "">
<cfif IsDefined("SESSION.nickname")>
 <cflock name="#SESSION.sessionID#" type="ReadOnly" time-
out="20" throwontimeout="Yes">
```

```
  <!--- Remember user's nickname --->
  <cfset tmpNickname = SESSION.nickname>
 </cflock>
</cfif>

<!--- Use either URL or FORM parameter to get the message to
send --->
<cfparam name="URL.from" default=#tmpNickname#>
<cfparam name="URL.sender" default=#CGI.REMOTE_ADDR#>
<cfparam name="URL.message" default="">
<cfparam name="URL.users" default="">

<cfparam name="FORM.from" default=#URL.from#>
<cfparam name="FORM.sender" default=#URL.sender#>
<cfparam name="FORM.message" default=#URL.message#>
<cfparam name="FORM.users" default=#URL.users#>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
 <title>Instant Messaging</title>
 <SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
    <!--
  // for going back to the user-listing template
     function gotoListUsers() {
   location.href="listUsers.cfm";
  }
    //-->
    </SCRIPT>
</head>
<body onLoad="window.setTimeout('gotoListUsers()', 15000);">

<cfif IsDefined("FORM.users")>

 <cflock name="#APPLICATION.applicationName#" type="Exclu-
sive" timeout="20" throwontimeout="Yes">
  <cfoutput>
   Sent the message &amp;quot;<b>#FORM.message#</b>&amp;quot; to<p>

   <!--- Loop through all receivers of the message --->
   <cfloop index="aUser" list=#FORM.users#>

    <!---  Append message at the receiver's queue --->
    <cfscript>
     // Generate Message-Queue for receiver if it does not
already exist
     if (NOT StructKeyExists(APPLICATION.MessageQueue,
aUser)) {
        StructInsert(APPLICATION.MessageQueue, aUser,
ArrayNew(1), true);
     }
     // Setting up the message package to send
     tmpInsert = StructNew();
     tmpInsert["from"] = FORM.from;
     tmpInsert["sender"] = FORM.sender;
     tmpInsert["text"] = FORM.message;

     // Send the package (= append it at receiver's Message-
Queue)
     ArrayAppend(APPLICATION.MessageQueue[aUser], tmpInsert);
    </cfscript>

    <!--- Print current receiver as sent --->
    <cfif StructKeyExists(APPLICATION.NickNames,aUser)>
     #StructFind(APPLICATION.NickNames, aUser)#
    <cfelse>
     #aUser#
    </cfif><br>
   </cfloop>

  </cfoutput>

 </cflock>

<cfelse>
 No users specified
</cfif>

<p>
<a href="listUsers.cfm">Go back</a>

</body>
</html>
```

CODE LISTING

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

The Screwtape **E-Mails**

Devilish advice for the age of the Internet

BY
HAL HELMS

From: screwtape@fourthcircle.hell
To: wormwood@e-hell.com

April 4, 1997

My dearest W, let me join so many others in congratulating you on your promotion to our new Internet division. I don't pretend to understand such things – I leave such newfangled enterprises to you young devils. In fact, my belated congratulations owe to the fact that I simply could not figure out how to send this e-mail thing. My "e-mail client" (as Stenchfoot calls it) has such a bewildering set of options (I have yet to understand acronyms such as BCC and SMTP) that I remained for the longest time perplexed by its dizzying choice of buttons and options. "Who designed this infernal thing?" I asked one day – and was delighted to find that one of our own had provided inspiration for many of the user interfaces in programs today.

Say what you will of the inability of large enterprises to respond quickly to changes (and no one knows better than I just what a bureaucracy hell is), I think that while the Internet may have caught us all off guard, our response has been swift and strategic. Of course, I was delighted and indulged in a certain pride to see you, my own protégé, in such a position of prominence. Ah, there I go – lecturing when I really only meant to write a quick note to say "Bravo." Keep up the good work.

April 14, 1997

W, I got your e-mail and am glad to see you're settling into your new post. I confess that I'm a bit concerned about your notion that "The Internet changes everything." Wormwood, we're operating on principles that have been tested in the very fires of hell – that have served us well over these many millennia. I hardly think they'll be set aside by a new technology. I think you'll have to agree that the century just past showed how effectively technology can be put to our own ends. So stay the course, W. Remember, they may be very clever, but they are just humans. Never lose site of the fact that your job is to do everything in your power to disempower them. What they want is a little humbling – that will dry up that nonsense about the Internet "democratizing" things.

May 28, 1997

W, I confess that I had not until now realized how dangerous this Internet thing is. Your last e-mail pointed me to developments that I had not realized had so quickly developed. Possibly, I haven't been as attentive as I might have been. Whatever. We need to act – and act quickly. When exactly did the notion that people could determine their own destinies achieve such widespread currency? I need hardly point out just how dodgy a position this puts us in. Still, we have some valuable allies "on the surface," as you so colloquially put it. It's time to call in a few markers. If programmers are building this Internet, then it's clear enough – we have to stop them.

June 19, 1997

Now, Wormwood, it doesn't pay to start panicking – I can assure you that such weakness is not looked on favorably at headquarters. But perhaps new times call for new tactics. You say that these programmers work on "projects." Why then do you not apply yourself in helping make these projects go bad? Surely an enterprising demon such as you will have many ideas on this. Just remember this, W: all bad human behavior is motivated by fear and greed. These are your two great allies.

June 23, 1997

I must confess my surprise at your questions. (What do they teach young devils, these days?) Let's examine the situation more closely. You ask "Fear…of what?"

Well, my boy, how are these for starters? Fear of change, fear of appearing foolish, fear of – well, let me give you a concrete example. A client of mine was recently working on a project – not a software project, but I think you'll see the basics remain the same – and I must say he was making alarming progress. However, what I had working on my side was that my client wasn't aware of how close he was to a major breakthrough. It was a fairly simple thing to slip some thoughts in. I began with "This isn't the way things are done." I followed up with "I should probably check to see if others are doing it the same way." Then the merest whiff of "What will others say?" was enough to derail what might have otherwise come to a most unfortunate conclusion.

[*Ed. Note:* In 1942 the English writer C.S. Lewis published a slim volume titled *The Screwtape Letters*, a set of letters written by an experienced demon of the underworld, Screwtape, to his nephew, Wormwood, providing hellish advice on how to ensnare humans in nefarious schemes. One cannot read these letters and remain unmoved by Screwtape's avuncular concern for his aspiring charge, and many readers have longed for an update on how Wormwood got along in the ensuing decades. Their wishes have finally come to fruition: new correspondence from Screwtape to Wormwood has been found, and *CFDJ* presents for the first time selections from this newly extant collection.]

You see, Wormwood, what you must do – I hope I don't need to stress how painful the consequences of failure in this would be – is to get humans to freely give up on themselves and look to others for guidance and direction. I confess that it's not as easy these days as it used to be. Great edifices of authority have been overthrown. Still, one wants results and not excuses. Get cracking.

July 3, 1997

Haven't I said it clearly enough? Derail their projects and these software engineers will change their tune quickly enough. In this your greatest ally is *complexity*, Wormwood. Appeal to their pride, boy. Suggest to them that "Real developers don't need some silly methodology." You'll want allies with vested interests to drum in this message. I don't give a fig who it is – surely there are people and companies out there who turn a pretty penny by taking advantage of obfuscation and convolution. Find them. Use them.

July 19, 1997

I'm glad to hear about your recent success, W. Still, you want to remain ever vigilant. These humans are a wretchedly tricky lot. You remember that client I wrote of – the one who I thought I had carefully navigated to a happy conclusion? Well, all my hard work was undone recently. It began innocently enough – he was reading a biography of that troublemaker Einstein. To make an ugly story short, the book got my client thinking that approaching problems with a "beginner's mind" was actually the true sophistication. While I was busy with other matters, the wretch began experimenting with new techniques and making his own decisions. The results were disastrous for our side. I fear this was a case of counting my chickens before they hatched.

While I'm on the subject, the idea of "gurus" is a two-edged sword. To the degree that gurus are seen as the unchallengeable experts, they undermine that most unhappy tendency of humans to discover for themselves. That, of course, is most desirable. But let my uncharacteristic slip instruct you: if you let your clients get too close to experts, they may find themselves inspired to become their own gurus. That, most assuredly, is not what we want. Always remember your ultimate goal: disempowerment.

August 12, 1997

I underestimated you, Wormwood. The notion that you would take my admission of a tiny slipup and use it to advance your own career by reporting me to headquarters was something I had not thought you worthy of. But I am not without resources, W. A fellow demon in HQ, with whom I had planned several successful campaigns, called to advise me of your report. Of course, I had it squelched, but I must say I was touched by your treachery, W. Why not take that admirable enthusiasm, energy, and ambition and use it against your clients?

Has it not occurred to you that one way to waylay programmers is to keep them from programming? How? I have always found meetings, my dear Wormwood, to be a demon's best friend. Not just any meetings, but preferably interminably long ones that have no stated agenda. Don't think this beneath you, W. The time spent in these meetings is time developers are not spending developing. That can only be good for our side.

Nor should pettiness be overlooked. It's highly effective – and can be great fun as well. I was consulting with a large phone company some years ago and made the suggestion that they place a clock on the door above the restroom to track how long employees spent in the restroom. A giant stopwatch. Even now, I chuckle thinking about how this company stated that their "people were their most important assets" while simultaneously tracking to the second how much time these "assets" were spending on bodily functions. There's no depth, W, to which some people will not sink in order to advance their own careers. For this, we can only be grateful.

September 11, 1997

How glad I was to get your last e-mail, W. You're exactly right: in an organization many people can say No, but only a few can say Yes. Your job is to keep these developers from those people who can say yes. I have always found spreading suspicion among groups to be most effective. Convince the developers that their customers are "them" and the rankest novice demon can follow up with how "they don't understand."

Then you can work on management, pushing the idea that developers are an untrustworthy sort. Having created the "problem," you can provide the "solution" by suggesting they install "filtering" software to keep employees from availing themselves of the full resources of the Internet. Present it to management as keeping workers on the straight and narrow. It's wonderfully disempowering. The results? Gridlock for the organization and another victory for us!

December 5, 1997

Well, again, congratulations are due, Wormwood! I must say, I always knew you had it in you. Your idea of making sure that the project requirements were "detailed" in 100+ pages of written specifications while preventing even a single prototype that might have brought to the fore the many details that, unresolved, went on to break the project – well, it's enough to make an uncle very, very proud.

And yes, I completely understand what you mean – once you have a true fiasco, there's so much blame to spread around that it can go on for months and undermine whatever trust (read "effectiveness") your clients may have been building. I hope you'll forgive an old demon's desire to point out what you obviously understand – that the important thing is to get the people involved to blame each other rather than to look at the system that created the results.

Well, I think you're in for a well-deserved vacation. I understand that the Balkans are lovely at this time of year. Why don't you take a little time and see what you can stir up?

Affectionately,
Screwtape

ABOUT THE AUTHOR

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

HAL.HELMS@TEAMALLAIRE.COM

Java for CFers Part 1 of 3

How to make Java (via JSP) a little less intimidating

BY BEN FORTA

Java is a reality. But for many CFers the buzz and hype surrounding Java is cause for much concern. For those of us who love ColdFusion because of its simplicity, Java can indeed seem intimidating. Much of that concern is legitimate. The fact of the matter is, there's no way you'll learn Java as quickly as you learned ColdFusion, and you'll definitely not be as productive as quickly.

Having said that, Java does have lots to offer. It's fast, it scales well, it runs on a wide range of platforms, it supports all sorts of services and extensions. There's a seemingly endless supply of Java-related resources out there and – perhaps most important – Java is respected by the development community (considerably more so than ColdFusion is). And, as I've mentioned in previous columns, a Java/CF convergence of sorts is on the way (not that you'll have to learn Java, but you'll be able to leverage it if you so desire).

In this column I'd like to start exploring Java (and JSP – JavaServer Pages – specifically) from a ColdFusion developer's perspective – and the best place to start is with some simple comparisons.

Note: If you'd like to tinker with JSP, and I strongly recommend your doing so, download an evaluation version of JRun from the Allaire Web site. JRun is a top-notch J2EE server with superb JSP support, and it also integrates nicely with ColdFusion (as you'll see in a future issue).

Introducing JSP

Java has lots to offer, but it's a pain to learn. What to do? The answer is JSP. JSP is essentially a tag-based scripting-style interface to Java development. With JSP you can write Java-based applications much like you'd write code in ColdFusion (or ASP and Perl, for that matter).

As I explained in a previous column ("ColdFusion and Java – A Match Made in E-Heaven," ***CFDJ***, Vol. 2, issue 4), Java is both a platform and a language, and the two are often confused. The Java platform is used for more than Web applications, although Java-based Web apps are what's of interest to us now. One way to create these apps is to use servlet technology – a servlet is a special form of Java application designed specifically for Web-based applications. In fact, you can think of a servlet as a compiled script (imagine compiling your CFM pages into binary code). Servlets provide simplified access to everything from GET and POST data to cookies and a whole lot more. And servlets are written in Java.

Or at least they were until JSP came along. Using JSP you can create servlets using tags and scripting. The JSP pages are then compiled and eventually become servlets, all without your doing anything special to make that happen. JSP is actually very similar to ColdFusion in that it processes scripts and generates output. But unlike ColdFusion, JSP generates compiled code that can be executed far quicker on subsequent requests. In other words, JSP provides you with a path to Java without your actually having to learn much Java at all.

Some Basics

Let's start with some syntax basics. CFML developers use tags, <CFSET>, <CFOUTPUT>, and so on. JSP developers also use tags, all of which start with <% and end with %>.

<%! %> is used to declare or define variables or functions. You'd use <%! %> wherever you'd have used a <CFSET> tag. And like <CFSET>, <!% %> is not used to display any output.

<% %> is used to delimit a script block (perhaps in Java or JavaScript). You can think of <% %> as being kind of like <CFSCRIPT> and </CFSCRIPT>.

<%= %> is used to evaluate expressions, much like <CFOUTPUT>, and for the most part you'd use it wherever you'd use <CFOUTPUT> tags.

<%@ %> has no real ColdFusion parallel (although it does the job of lots of different ColdFusion tags and options); it's used to specify directives – page-wide properties that affect the generated servlet. For example, you'd use a directive to set the output MIME type (equivalent to <CFCONTENT>) and to turn on session-state management (equivalent to <CFAPPLICATION>).

Note: JRun Studio is the editor of choice for JSP development. But if you have ColdFusion Studio already, you'll be pleased to know that it comes with basic JSP support and color coding built right in.

Using Variables

Let's start with something simple – variable assignment. The following ColdFusion code creates a variable and assigns a value to it:

```
<CFSET name="Ben">
```

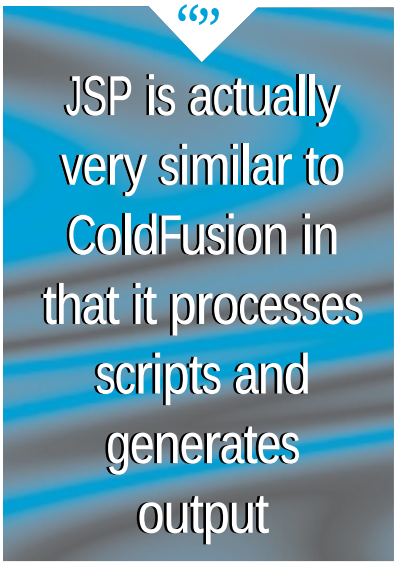
This is the equivalent JSP code:

```
<%! String name="Ben"; %>
```

The big difference here is that the JSP code also defines the variable type. Actually, the type could have been left out, but most JSP developers like to state the types explicitly to avoid possible confusion or ambiguity. Another difference is that a single tag set can be used to define multiple variables, like this:

```
<%! String fname="Ben", String
lname="Forta"; %>
```

Now that you know how to create variables, let's look at what it takes to use them. Here's a simple ColdFusion code snippet:




```
<CFOUTPUT>Hello #name#</CFOUTPUT>
```

And here's the JSP equivalent:

```
Hello <%=name%>
```

In truth, the examples aren't the same. The line of JSP code I showed you is more like this (with the <CFOUTPUT> tags just around the variable, not including the word Hello):

```
Hello <CFOUTPUT>#name#</CFOUTPUT>
```

Unlike ColdFusion, which assumes that all text within an expression is literal text unless explicitly flagged as not (by using pound signs), JSP assumes that all text within an expression is an expression to be processed. If you want to include literal text within your expression, you'll need to let the JSP processor know that it is literal text by enclosing it within quotes, like this:

```
<%="Hello "+name%>
```

Conditional Processing

Conditional processing in JSP is also quite similar to CFML. Here's a typical CFML <CFIF> statement – it evaluates an expression and displays one of two possible outputs depending on whether or not the expression is zero:

```
<CFIF cart_contents GTE 1>
   <CFOUTPUT>#cart_contents# items in
cart</CFOUTPUT>
```

```
<CFLESE>
   Your cart is empty
</CFIF>
```

And here's the JSP equivalent:

```
<% if (cart_contents >= 1) { %>
   <%=cart_contents%> items in cart
<% } else { %>
   Your cart is empty
<% } %>
```

This JSP code contains three script blocks (each enclosed within <% and %> tags). One script block could have been used, but then the code to generate the output would have been a little more complex.

Using Loops

Loops are no more complex than if statements. Here's a CFML loop – it loops from 1 to 10, displaying the current loop count as it goes:

```
<CFLOOP FROM="1" TO="10" INDEX="i">
   <CFOUTPUT>#i#</CFOUTPUT><BR>
<CFLOOP>
```

Here's the JSP equivalent:

```
<% for(int i=1; i<10; i++) { %>
   <%=i%><BR>
<% } %>
```

The JSP for() loop takes three parameters: the first is the start value (here a variable is defined and assigned the start value in one step); the second is the condition that must be true for the loop to continue (here the loop will continue as long as i is less than 10); the third is an action to be performed on each iteration (here i++ is used to increment i on each loop; to loop backwards, i could have been used). This same for() loop can be used to check for any condition (like <CFLOOP CONDITION=""> in CFML).

Using Includes

ColdFusion has supported includes for a long time via the <CFINCLUDE> tag:

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

Here's the JSP equivalent:

```
<%@ include file="file.jsp" %>
```

Simple. Enough said.

Commenting Your Code

All code is commented (or should be). Here's a CFML comment:

```
<!--- This is a comment --->
```

And here's a JSP comment:

```
<%-- This is a comment --%>
```

Your JSP code, like your CFML code, can also use simple HTML comments if you want the comment to be sent to the client.

Summary

We've just scratched the surface – but hopefully Java (via JSP) is looking just a bit less intimidating. We'll continue this topic in the next issue with a discussion on forms, URLs, database access, and more. We'll also look at how to invoke Java code from within CFML. Stay tuned.

ABOUT THE AUTHOR

Ben Forta is Allaire Corporation's product evangelist for the ColdFusion product line. He is the author of the best-selling ColdFusion 4.0 Web Application Construction Kit *and its sequel,* Advanced ColdFusion 4.0 Development, *as well as books on* Spectra, HomeSite, and SQL. *He recently released* WAP Development with WML and WMLScript *and* JavaServer Pages Application Development.

BEN@FORTA.COM

Introducing UltraDev
by Macromedia, Inc.

Dreamweaver does data

REVIEWED BY
JERRY BRADENBAUGH

Drag-and-drop Web development – it's not a new concept, really. It dates back a few years to the first releases of NetObject's Fusion and Microsoft's FrontPage. It enables developers and designers (and administrative assistants, janitors, and just about anyone else) to drag and drop their way to Web page content quickly and easily – no real Web coding experience necessary.

Macromedia's latest development in this arena is Dreamweaver UltraDev. The product is built on the standard Dreamweaver product (the company's GUI Web development tool). UltraDev raises the bar for GUI Web tools by letting coders view live data while they work. This feature significantly reduces the development process. Here's why.

One of the pains of Web dev is the somewhat cumbersome page creation process. It goes like this:

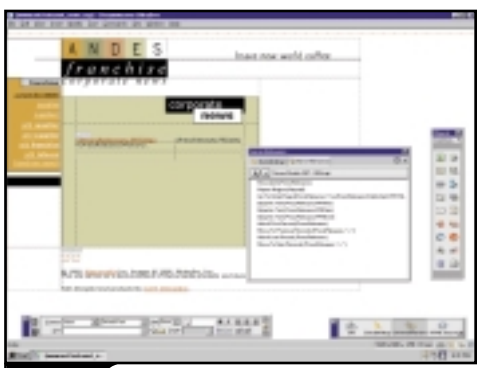


FIGURE 1: Viewing the recordset references

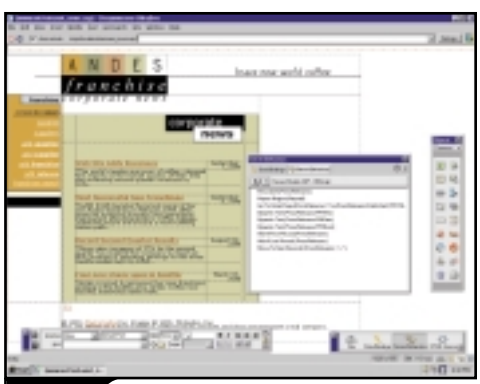


FIGURE 2: Live Data Preview in action

1. Write code or drag and drop content.
2. Save the changes.
3. Call the script from a browser to make sure everything works correctly (e.g., database queries) and looks right.
4. Repeat entire process until page is finished.

That's all fine and dandy until you have to run through the process a few dozen times to get everything right. Saving your code and refreshing your browser each time begins to wear.

If only you could view and modify the content from the database as you work, you'd save countless minutes toggling back and forth between applications. Dreamweaver UltraDev has pushed the envelope to give developers just that. In addition to its superior features, this product offers Live Data Preview, which lets you view and make changes to your database content as you develop. It's pretty easy, too.

Live Data Preview

You need three components to make it happen: a site defined in UltraDev, a database connection, and a recordset defined in UltraDev to reference your content output.

Defining a new site is a snap. You have to provide UltraDev with information for three elements: Local Info, Web Server Info, and App Server Info. Local Info lets you give the site a name and specify where the files for the site are stored on your computer. Web Server Info is where you specify how to access the site (locally, through your computer's network or via FTP) to update the file changes you make. Finally, App Server Info allows you to choose an application server environment (ASP, ColdFusion, or JSP). Whichever you select will be responsible for generating dynamic content.

Next is the database connection. You can define an ODBC data source using the ODBC administrator in Windows or a JDBC database connection (Mac users have to use JDBC). You can define a JDBC database connection using UltraDev's Connection dialog box. You need the proper Java classes installed on your machine (a driver is provided in the bundle). Macromedia provides the RmiJdbc driver for Mac users. You can download or purchase other drivers depending on the database you use.

VITALS

Product Name: Dreamweaver UltraDev
Vendor: Macromedia, Inc.
600 Townsend Street
San Francisco, CA 94103
Tel: 415 252-2000 **Fax:** 415 626-0554
Web: www.macromedia.com
Functionality: Web development tool
Price: $599 (Drumbeat 2000 upgrade: $99; standard Dreamweaver 3 upgrade: $299)

What's Hot
Preview live data during development
Supports multiple application-serving environments
Good site management capabilities
Great extensibility via JavaScript, C, and other APIs

What's Not
Interface clutter
Lame text editor
Steeper learning curve

All that remains is creating a recordset to contain the results of your database query, and UltraDev's Data Binding Inspector will create it for you. Once you've named a recordset and associated it to a database connection, just drop in the recordset data where you want it to go on the page.

Figure 1 displays a page containing items {PressReleases.PRTitle}, {PressReleases.PRDate}, and {PressReleases.PR Short}. These refer to columns in a recordset called PressReleases. PRTitle refers to the database column containing the name of the press release, PRDate refers to the date, and so on.

Figure 2 shows the same Web page, but this time in Live Data Preview mode. The recordset references are replaced with actual data, and there are four rows. UltraDev runs the query and displays the results directly into your development environment. Now you can change the way the content looks with fewer hassles. Change the color of the press release title to green, then make it a link. Format the date. Italicize the short description. Whatever you want. Any way you slice it, your life just got easier.

Going Further

Want even more functionality from UltraDev? Add it yourself. The product includes Dreamweaver's big fat JavaScript programming API and Document Object Model, which adds up to serious extensibility. You can create your own inspectors, server behaviors, server models, object palettes, menu commands, and more. You can even download and install extensions that others have created from the Dreamweaver UltraDev exchange site (check out www.macromedia.com/exchange/).

The extensibility doesn't stop there. You can also tap into the HTTP API for extensibility outside the realm of the local file system. The product's C-level extensibility enables you to push the envelope in areas that JavaScript can't reach, such as file input/output.

What I Didn't Like

Ain't that just like a product reviewer? Always something not to like. Nothing here is that bad, but hey, I'm all about full disclosure.

Too Much Interface Clutter

Earlier versions of Dreamweaver are just as guilty of this – floating palettes, property inspectors, menus, and documents. While working on only a single document it's easy to have six or seven items on the screen that block the view of the page you're designing. You'll probably find yourself moving a lot of things around or doing a lot of hiding and showing. You can reduce this problem if you increase your monitor's resolution – especially if you can set it to 1280 x 1024, or the like.

Light on the Text Editor

The text editor is pretty lame (not feature rich) and adds to the clutter mentioned above. Fortunately, HomeSite and BBEdit are both great text editors and are included in the application bundle.

Steeper Than Normal Learning Curve

With so many great features in the application, first-time users might be intimidated by the seemingly massive amount of menu options. Once they overcome the shock and use it a few times, they'll appreciate how deep this thing goes. Just make sure you keep your nose buried in the tutorials.

Availability and Pricing

Dreamweaver UltraDev is available in Windows 95/98/2000 and NT4. It's also available for the Mac (PowerMac). A full boxed or downloaded version costs $599. As far as upgrade pricing is concerned, those upgrading from Drumbeat 2000 will pay $99. Those upgrading from the standard Dreamweaver 3 will pay $299.

Wrapping Up

In the end, Dreamweaver UltraDev is a solid and feature-rich application that lets users quickly create dynamic Web content while viewing live data. This eliminates repetitive code saving and browser refreshing, and hence speeds the time to product delivery. Since it supports ASP, ColdFusion, and JSP, developers can work with the technology they're most comfortable with. Despite some small drawbacks, the product is a winner.

ABOUT THE AUTHOR
Jerry Bradenbaugh is a Web architect for CynicProductions.com.

JERRY@CYNICPRODUCTIONS.COM

RAD++

Propel Your ColdFusion Projects with Application FRAMEWORKS

A comparison of Fusebox, Spectra, and iiFramework

BY
PATRICK STEIL

What is an *application framework*? This article defines the term and compares three current application frameworks or methodologies. I'll also explain how they're used to implement "foundational" functionality, which can be reused in other applications with little or no programmer involvement.

What Is an Application Framework?

I've often used the metaphor of building a house when discussing the Web application development process with my clients. When building a house, different subcontractors come in to complete separate but equally important building tasks. An application framework is like having the lot cleared, the slab poured, and the frame raised. I call this part the *foundation*. The soundness of the foundation is critical. If it's not well constructed, you might be better off tearing down the house and starting from scratch rather than adding that second story. We all know it would have cost less in the long run if the foundation were strong enough to handle your additional growth needs.

Once the foundation of the house is complete, the interior walls go up, the painters paint, the carpet's laid, and so forth. The painter doesn't have to know how the slab was poured or how the frame was put together. The painter needs to know only that the kitchen should be painted peach and the dining room green.

When building a Web-based application with an application framework in place, you want this framework to handle as much of the foundational functionality as possible. Then, rather than reinventing the proverbial wheel, you can focus on customizing the functionality of the Web site. The foundational functionality may differ slightly depending on the application you're building, but in general the areas shown in Table 1 should be considered foundational functionality for any Web-based application.

Category	Description
Content Management	Provides an interface so anyone can easily edit and manage static and dynamic content such as Web site text, formatting, and pictures
Authentication and Session Management	Allows for identification of user and tracking of that user's session
Role-Based User Management	Lets you define the types of users and their access to the system. These permissions should have support for dynamic assignment and runtime changes
Exception Handling	Handles the errors in the application; reports and logs these errors in a consistent manner
Navigation Control with Graphics Integration	Manages the links a user may access, based on evaluation of the user role and profile
Application Variable Management	Provides a Web-based method to configure and reconfigure variables for an application
Application/Site Creation and Management	Provides a Web-based method to create a new database, directory structure, and foundational functionality for each application
Search Engine Support	Supports page titles, meta tag keywords, meta descriptions, and the like; enables search engine marketing of the Web site
Personalization	Enables the system to dynamically tailor content layout, based on the user's preference and past use of the site

TABLE 1: Generic Web application framework functionality

What Should an Application Framework for Web-Based Applications Accomplish?

In an ideal world an application framework would have all the features mentioned in Table 1 and the developer wouldn't be required to learn how they're implemented. Requiring an application framework to live up to the promise of better developer productivity and robust, secure applications with a rapid application development time without a tremendous learning curve is a lot to ask.

Let's examine three separate technologies and see how they compare.

Fusebox – Framework or Methodology?

What is Fusebox? Many developers have heard about Fusebox but haven't had an opportunity to use it and see what it accomplishes. The idea behind Fusebox architecture is, as stated in the Fusebox Web site (www.fusebox.org), "to increase teamwork efficiency, productivity, and reduce the time it takes to build a Web application."

I've seen several presentations and implementations of the Fusebox architecture, and have read through the Fusebox Web site. My determination is that Fusebox is a developer's application methodology. It's a set of agreed-to standards on how a Web application should be written. By following the Fusebox standards, a project that's utilizing multiple developers would be more productive because it addresses issues such as file-naming conventions. What Fusebox doesn't address are standards for encapsulating commonly used functionality.

Fusebox addresses some fundamental standards that, until recently, hadn't been addressed in ColdFusion application development. For this reason I think that Fusebox has a place in the developer's world as a methodology. To be fair, Fusebox doesn't claim to be an application framework. I'm discussing this technology to emphasize and contrast an application framework versus an application methodology.

When to Use Fusebox

If your development group doesn't have any standards in place for building applications, Fusebox can help, although I have some concerns with the Fusebox methodology. It's a programming methodology that can help enhance your team development.

When Not to Use Fusebox

If you're creating a public "Internet" site, Fusebox is not ideal because, among other issues, it lacks support for search engines. Fusebox application pages can't be dynamically indexed, which is important to the search engines. The reason for this is, there's only one root-level file for the entire application – the index.cfm. To provide custom functionality, a fuseaction URL variable must be passed. Passing parameters in the URL will prevent many search engines from "spidering" the pages within the site.

Table 2 provides a comparison of the Fusebox methodology with the application framework requirements.

Spectra – What Is It?

Spectra is the latest ColdFusion-based product offering from Allaire. Allaire calls it "the first packaged system to address the three critical areas of every successful online business – content management, e-commerce, and personalization." I'm not sure what a "packaged system" means, but if you look deep enough, you'll see that Spectra is indeed both a programming methodology and an application framework.

Spectra frees the programmer from the responsibility of role-based security and authentication by allowing nontechnical people to administer security and authentication. It handles content management, and its workflow capabilities appear to be quite robust. You also get personalization capabilities, if you need them. Spectra includes a site design tool as well.

Application Framework Functionality	Description
Content Management	Not supported
Authentication and Session Management	Uses ColdFusion "out of the box" session tracking; no authentication model defined
Role-Based User Management	Not supported
Exception Handling	Not supported
Navigation Control with Graphics Integration	Provides standards for how to provide sitewide common navigation
Application Variable Management	Not supported
Application/Site Creation and Management	Not supported
Search Engine Support	None; in fact, it deters search engines from indexing your site
Personalization	Not supported

TABLE 2: Comparison of Fusebox and application framework functionality

The Spectra application framework requires you to adopt its ColdFusion programming methodology. This methodology is an object-based paradigm requiring you to plan and design your objects, methods, and properties carefully. This is not unusual and is the case for any object-oriented system or any application development methodology. Each "object" in the system then has common services that can be used for functionality, such as logging, granting permissions, and supporting metadata. Inheritance is not supported. Figure 1 shows a content management screen in Spectra v1.0.1.

A thorough overview article of Allaire Spectra appeared in the October 1999 issue of ***ColdFusion Developer's Journal*** (Vol. 1, issue 5) located online at www.sys-con.com/coldfusion/archives/0105/krinsky/index.html. This article makes some very good observations regarding the advantages of Spectra, but also addresses some of the shortcomings of version 1.0.

To use Spectra for development, the programmer is required to adopt the OO methodology, and my experience and investigation show there's a big learning curve.

When to Use Spectra

Spectra is well suited for very complex enterprise-level applications in which you may have many, many different types of users, each with very complex access or authority rules.

When Not to Use Spectra

Due to its high price tag and the time investment necessary to learn how to adopt the Spectra programming methodology and

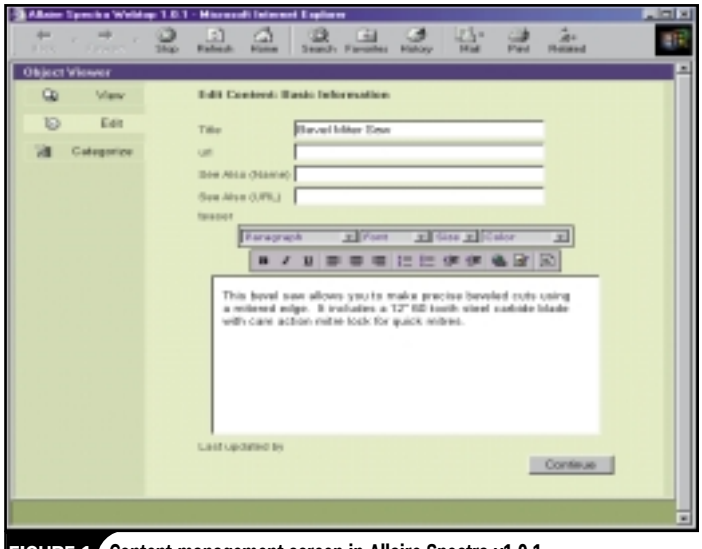


FIGURE 1: Content management screen in Allaire Spectra v1.0.1

API, Spectra rather negates the whole rapid application development mantra that has made ColdFusion so popular in the first place. I don't see smaller departments and companies, or the "let's get this application up and running by this Friday" type dot-com businesses, widely using Spectra. Table 3 provides a comparison of Spectra with application framework functionality. Table 4, which originally appeared in the October 1999 issue of ***CFDJ*** (Vol. 1, issue 5), provides an overview of Spectra.

Category	Description
Content Management	Yes, Spectra uses container-based content management, and provides content scheduling and syndication
Authentication and Session Management	Yes, Spectra depends on ColdFusion's underlying CFID/CFTOKEN user session tracking. It does provide automatic user authentication, when needed
Role-Based User Management	Yes, Spectra provides a robust system for managing access, down to the object level. A trained Spectra security administrator must administer this
Exception Handling	None. No exception handling framework in Spectra
Navigation Control with Graphics Integration	None
Application Variable Management	Spectra has a fair number of variables that could be thought of as application-level variables, and because of the heavy use of structures internally, permits access to system-level object properties and the properties you define for your objects via the structure elements
Application/Site Creation and Management	Yes, you must create a new database; then Spectra will set up the necessary tables to support your application
Search Engine Support	You must provide the programming/management for search engine meta tag keywords
Personalization	Built-in support for personalization

TABLE 3: Comparison of Spectra and application framework functionality

Characteristic	Plus	Minus
COAPI	Faster, easier, less error-prone development. More predictable development time lines. Enables widespread code-reuse.	Substantial learning curve. Reduced control and flexibility. Deeply nested custom tags reduce performance.
Globally Unique Identifiers (GUIDs)	Facilitates synchronization, syndication.	Marginally slower data access. Longer IDs make for heavier pages and JavaScript.
Central Object Table	Allows retrieval of collections containing arbitrary types of information. Facilitates provision of common services such as permissions and logging for all objects.	Complicates data extraction. Does not natively support relational data models, integrity, or dependencies.
Object-Relational Mapping	More intuitive, applicable object-oriented development. Supports traditional relational databases. Allaire takes care of optimization and query building.	Mapping incurs substantial performance overhead. Substantial performance penalties for complex functions (can be mitigated through caching).
WDDX Object Storage	Enables persistence of virtually infinitely complex objects. Extremely fast retrieval of complex objects (one "get").	Incurs WDDX deserialization overhead with every "get."
Object-Level Permissions	Context-insensitive retrieval and presentation functions.	Error-prone input. Incredibly slow without permission caching.
"Black Box" DataStore	Data integrity. Data and application portability and syndication. Facilitates upgrades and updates.	Limits customization and optimization. Accentuates product deficiencies (e.g., relationships, synchronization).
GUI System Configuration	Point-and-click data modeling and creation. Rapid system setup and deployment.	Point-and-click data model modification and destruction. Facilitates ad hoc, chaotic development.
Site Modeling Services	RAD site modeling framework.	Limited functionality may not accommodate menuing and layout requirements.

TABLE 4: Significant product characteristics

iiFramework – What Is It?

iiFramework from Infrastructure Inc. is a ColdFusion add-on product for Web application developers. It's a true application framework that implements most of the features we're discussing in this article (see Table 5).

iiFramework was developed through an iterative process of providing common functionality over and over again in real-world Web-based applications. Most of the code that makes up the iiFramework is actually two to four years old. Infrastructure took all the components of the iiFramework and rolled them into a product that allows an application to be built at a rapid rate. With iiFramework the application developer doesn't have to be aware of or need to know how to design and program foundational functionality. It requires very few changes to your programmer's current coding practices.

iiFramework uses the Site Manager Module to create a new site. This site can be based on any other site you may have previously created in the iiFramework. This allows reuse of an entire application with full customization capability. You can code the business logic into one site, then easily replicate that logic to any other site. The new sites will automatically inherit the look and feel of the base site. iiFramework separates business logic from presentation. To take full advantage of these features, Infrastructure provides 15 to 20 custom tags that you should become familiar with but are not required to use.

Where Spectra is object-centric, iiFramework is page-centric. Everything revolves around the model of the "page" or "template." Every page in the site must be registered with the iiFramework system (see Figure 2). If a page isn't registered, the iiFramework won't allow it to be displayed. If it is registered, the iiFramework will ensure that the current user has access to this page and that it's active before displaying it. Any page can be cloned and customized.

The iiFramework manages everything about this page including the static content, the ColdFusion code, the help content, and even the WML content. Yes, iiFramework supports WAP devices, even if you don't provide specific WML content. iiFramework manages permissions to the page by allowing you to specify which groups have access to it. With iiFramework, content management is supported by an integrated WYSIWYG editor.

Graphics layout of the site is handled via sitewide variables. Navigational control is handled by the iiFramework so the programmer doesn't have to code a menu of links. The system automatically reformats your navigational links, based on your current login and the pages your group has authority to access. If a user doesn't have access to a page, its links aren't shown. If the user knows the URL for the page, the system will still verify access authority and keep that user out.

All exceptions are automatically handled by the system and formatted within the same sitewide graphics layout as the rest of the site. The errors can be automatically e-mailed to a programmer for attention.

When to Use iiFramework

iiFramework works well with your typical ColdFusion Internet, intranet, and extranet applications that need to be developed rapidly. Any project requiring development time of one to 16 weeks will fit well into the iiFramework system. It's been used for 25,000-part catalog stores and program management systems, and Infrastructure uses it for all its company Web sites.

If you plan to build a CF application, iiFramework will help you build it better and make it more secure, supportable, and scalable.

When Not to Use iiFramework

If you're looking to build a complete reusable class library for an enterprise-wide development strategy, iiFramework may not be the best tool because it doesn't force an object-oriented methodology. While you can still reuse code through standard ColdFusion custom tags, by default you can't use inheritance and other OO properties that can be very powerful.

FIGURE 2: Page-level administration screen for an iiFramework Web site

Other Programming Methodologies

As I mentioned, Fusebox was chosen for this article as an example of a methodology versus an application framework. I also chose Fusebox because it's been around the longest and gets more press to date than any other such methodology. Two other ColdFusion application development methodologies are worth checking out (see www.smart-objects.com and www.cfobjects.com). These are two object-oriented methodologies that may complement your coding practices. They certainly can be used in conjunction with the iiFramework.

Resources

1. *Fusebox:* www.fusebox.org
2. *Spectra:* www.allaire.com/products/spectra
3. *iiFramework:* www.iiframework.com.

Application Framework Functionality Provides	What iiFramework Provides
Content Management	Yes, HTML, Help and WML content can be administered by any end user. No scheduling of content
Authentication and Session Management	Yes, iiFramework uses a proprietary session-tracking system that's more secure and scalable than the ColdFusion CFID and CFTOKEN. Provides automatic user authentication when needed
Role-Based User Management	Yes, iiFramework has a robust system for managing access at the page level. Can be administered by any end user. Every page must be registered with the system
Exception Handling	Yes, iiFramework will handle any exception not handled directly by your code. It will intelligently decide whether or not to e-mail the exception to your programmer
Navigation Control with Graphics Integration	Yes, iiFramework handles all navigation management for public and private links
Application Variable Management	Yes, iiFramework provides complete global and per site variable management through administrative pages.
Application/Site Creation and Management	Provides Web-based interface for creating new sites, making them active/inactive, and managing sitewide variables. New sites created from templates in minutes. New database is not necessary for each new site
Search Engine Support	iiFramework allows you to administer meta tag keywords for every page in the system through the administrator
Personalization	No built-in support for personalization.

TABLE 5: Comparison of iiFramework and application framework functionalities

About the Author

Patrick Steil is president and chief technology officer of Infrastructure Inc., a Texas-based software development company. As cochairman of the DFW ColdFusion User Group, he's very involved with the ColdFusion development community.

patrick@infranet.com

Build a Better Help Screen

Use JavaScript to add a professional touch

BY
RANDY L. SMITH

The U.S. Fish & Wildlife Service had lived with their ColdFusion-based, accomplishment-reporting Web site for about a year. The application allowed employees to report field accomplishments to area managers who then edited and released the reports for internal-only or public review on their Web site, or via e-mail or fax to the media or Congress.

I had used a fairly standard frame layout with a header, logo corner, left menu, and main area. Additionally, there was a narrow strip under the title where a drop-down selector and a user identifier were loaded when field reps signed in. I added a small panel at the bottom where help information would be displayed whenever a help icon was clicked.

The help table is simple, consisting primarily of a short, text-based, key-term field and a longer memo field that holds the help information.

Can't See the Trees

Even with six frames, the main viewing area was adequate on a 17-inch or larger monitor, but smaller screens were a tangled confusion of scroll bars and mini-panels. Employees (field reps) might be using the latest computer equipment or antiquated computers with 14-inch monitors. The field reps with the smaller monitors found it difficult to view the tiny main panel, and the vertical and horizontal scroll bars made even more of a mess (see Figure 1).

My client gave me free reign to do anything to provide more room for the main display area. I tightened the frames and moved the menu onto the user information strip by using smaller buttons (see Figure 2).

I had removed the narrow help strip at the bottom, but still wanted the ability to display help information to users without causing them to lose their place on the Web site – especially if they were in the middle of filling out a form. I could have used the <A title="help text here" > approach, but the help information is pulled from a database to allow it to be updated over time. I didn't want to add to the page-load overhead by querying the help table on every page that had a help button. I could have used a target= "_blank" anchor to display help information in a new browser window, but I wouldn't have had control over its size or location. I felt it would confuse users if some or all of the main window was hidden when the new browser window was spawned.

JavaScript Limitations?

JavaScript seemed to be the way to go, but I hadn't taken the time to learn it yet. I picked up a beginner's book and began to read in earnest. My first attempt at solving the problem was to write a function that would dynamically load the results of a ColdFusion query and display it in an alert box. My theory was that I could use the <IMG> tags' OnClick() function with a parameter to initiate a query, load the information into the alert box, and display the help information to the user. Very clean, very professional looking, and I get to keep my new-found screen real estate.

JavaScript refused to cooperate. I tried several methods, but couldn't get the information into the alert box unless I queried the help table every time I loaded the page.

One Step Forward...

Back to square one. I found an example where JavaScript was used to create a new window and dynamically write an HTML page. I created a working CF script and copied it into a JavaScript function.

FIGURE 1: Hard-to-view main panel

FIGURE 2: Making more room for the main display area

FIGURE 3: Query results

Clicking the button was supposed to execute the function that was, in turn, to write the code into the new window.

Still no go. JavaScript executed the embedded ColdFusion instructions as soon as I loaded the parent page. Trying to fool it with """ or any other workaround didn't get me anywhere because the resulting dynamically written file was an HTML page, not a CF script.

Never Say Die

I'm not easily deterred by a few minor setbacks, but this problem was looking an awful lot like my JavaScript Waterloo. Suddenly, I had a revelation: What if the program I wrote called a static ColdFusion script that then handled the query and generated the results?

Even better, I discovered that the first parameter of the JavaScript "open" function would call a URL. The solution was simple! Use OnClick() to call a function with a parameter. The function would create a new window and call a ColdFusion script with the help query parameter. The results of the CF program would populate the new JavaScript box.

First I needed a ColdFusion program that would search the help table for a parameter and display the results of the query. HelpLine.cfm (see Listing 1) searches for the key term and outputs the query results (see Figure 3). The Close button (also JavaScript) provides a clean method of exiting, and the topic is displayed in light gray at the bottom of the screen. Displaying the topic name helps you find that specific item later when you need to update the help text.

To make this help-display capability available to all the CF programs in that application, I loaded the HelpWindow( ) into the Application.cfm file. This function is to be called with a parameter. HelpWindow( ) creates a new, nicely formatted browser window, then executes HelpLine.cfm, which populates the window:

```
<script language="JavaScript">
function HelpWindow(TheTopic) {

window1=window.open("helpline.cfm?topic=
"+TheTopic,
"NewWindow1","toolbar=no,directo-
ries=no,menubar=no,scrollbars=yes,
width=600,height=300,left=200,top=100
");
 }
</script>
```

The final piece of this puzzle is to enable a call to HelpWindow() with the appropriate parameter when the user clicks on a help button; we do this by placing the function name in the OnClick method of the <IMG> tag. Note the location of the parameter GEOREGION. We'll pass it through HelpWindow() to HelpLine.cfm where it can be used in the query to find the appropriate help information:

```
<img src="../images/btnHelp.gif"
width=15 height=15 alt="Click on the
question mark to display information
about this item. It will appear in a
new window in the middle of your
screen." border=0 align="middle"
onclick="HelpWindow('GEOREGION')">
```

Recyclable

This code can easily be reused, enabling you to quickly provide professional-looking help displays with minimal effort. To recap, here are the four elements necessary:

- *HelpLine.cfm*
- *HelpWindow()* in Application.cfm file or the CF script it's called from
- *Help database*, populated with data, and a datasource link
- *OnClick() functions* of <IMG> tags programmed with the appropriate help parameter, as in OnClick="HelpWindow ('GEOREGION')"

Summary

Dynamic help screens are an excellent use of a temporary JavaScript window. Application users can quickly and easily get the information they need, close the window with little fuss, and return to exactly where they left off. This is especially beneficial on forms, where taking users away can cause problems and confusion.

ABOUT THE AUTHOR
Randy L. Smith is president/CEO of MCPI, LLC, an Internet/intranet database solution provider based in Hudson, Wisconsin.

RANDY@MCPI.COM

Listing 1: HelpLine.cfm

```
<cfparam name="topic" default="intro">
<cfquery name='HelpDisplay' datasource='govsite'>
SELECT description_of_item FROM help WHERE key_term='#topic#'
</cfquery>
<html>
<head>
 <title>System Help</title>
</head>
<body bgcolor="White" leftmargin=50 topmargin=10>
<h3 align=center>System Help</h3>
<p>
<table align=center width=80% cellspacing=2 cellpadding=2
border=0>
 <tr>
  <td align=LEFT valign=middle>
   <cfoutput query='HelpDisplay'>
    <font size=-
1>#ParaGraphFormat(description_of_item)#</font>
   </cfoutput>
  </td>
 </tr>
 <tr>
  <td align=center>
   <br><br>
   <form>
    <input type="button" value="Close"
onClick="window.close()">
   </form>
  </td>
 </tr>
 <tr>
  <td align=right>
   <font color="#C0C0C0"><cfoutput>#topic#</cfoutput></font>
  </td>
 </tr>
</table>
<br clear='all'>
<p>
</body>
</html>
```

CODE LISTING

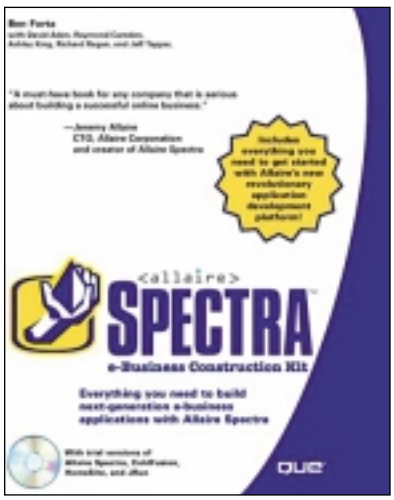
The code listing for this article can also be located at www.ColdFusionJournal.com

Next-Generation **E-Business Apps**

Allaire Spectra E-Business Construction Kit

By Ben Forta
900 pages, Que

REVIEWED BY ANNETTE KUNOVIC



It's always good when books can supplement or even replace application guides that come with a product. Ben Forta's latest book, *Allaire Spectra E-Business Construction Kit*, provides an in-depth view into Allaire's newest product, Spectra.

So you've decided that Spectra is the right fit for your project, and now you want to know if it's worth buying. Yes, it definitely is. (It's from Ben Forta, right?) Will you learn Spectra from this book? Well, that depends on you. Spectra is powerful but complicated. It requires you to really know ColdFusion as that's what Spectra's built on. You also need to know how to manipulate arrays, structures, structures of structures, structures of arrays, and arrays of structures.

Spectra is not for ColdFusion beginners and neither is this book. It's rightly aimed at intermediate to advanced developers. I'd advise taking the Spectra class offered by Allaire (www.allaire.com/Handlers/index.cfm?ID=16136&method=full), and using the book as a reference to reinforce the information from the class.

Whether or not you take the class, if you're an intermediate developer you'll probably benefit from this book, even though it's not perfect. After reading the book from cover to cover, I came across the following problems:

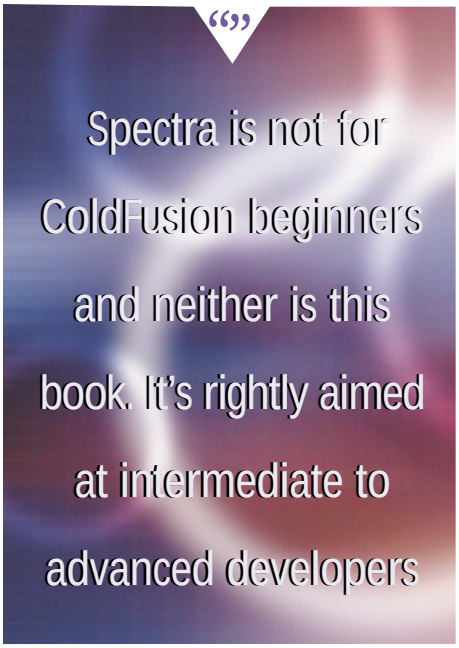
1. Although the screen shots were helpful, my installation didn't go smoothly. The knowledge base article listed in the book didn't address my errors. I ended up going to the new Spectra Support Forum (http://forums.allairecom/spectraconf) to find an answer. The Knowledge Base article 14378 fixed my 'Cfauthenticate tag error' and 'Internal security error' – installation problems usually stem from advanced security errors.
2. Unfortunately, the book was written before Spectra 1.01 was released, which means the MS Access databases provided won't work. (Spectra 1.01 uses the Upper() function that's not supported in Access.) If you're using Spectra 1.01, upsize the database to MS SQL using the Access Upsize Wizard or create a blank MS SQL database and import the Access tables to it. (The authors are aware of this problem and are working on a permanent solution.)
3. Be aware that the Webtop and the ColdFusion Studio Spectra toolbar are connected and work together. When you start in ColdFusion Studio, you sometimes end up on the Webtop. This is normal, but there's no warning about it so it can be confusing.
4. In Chapter 6 there's missing code (it's not in the book or on the CD), some of the templates aren't named correctly, some of the screen shots don't match the instructions, some of the instructions aren't clear, and two custom tags are used but never explained. You may have to reread this chapter to understand it. Most of the errors I found were in this section. You can get more information at www.forta.com/books/errata.cfm?p=9.
5. There was no discussion about where to put templates and why. Do you put them under the Web server root, the Spectra installation root, or an application root? Knowing this would have been helpful.
6. Make sure to check your paths and use the "Refresh button" (if presented) often. Each code listing has a path; if you don't use the correct one, Spectra won't find the templates it needs to create whatever you're creating. The error message will probably be too cryptic at this point for you to correct the mistake. Use the "Delete" button and start again.

Except for these few problems, here's what's best about the book:

- The background of Spectra is discussed, which creates the foundation needed for a successful application.
- The many screen shots help make sure you're where you should be.
- Each area of the Webtop is explained, giving you a chance to familiarize yourself with its navigation.
- It has well-written chapters (excluding Chapter 6, which I mentioned before).
- You create a simple application using different code-creation methods for various topic areas. This allows you to determine what coding method works best for you.

ABOUT THE AUTHOR

Annette Kunovic is cofounder of Trilemetry, Inc. (www.trilemetry.com), a Colorado-based Web applications development company. She is a certified Allaire instructor and was a technical editor on the book Mastering ColdFusion 4 *from Sybex.*



- The format of the code and how it's presented is great. You get to review the code first and then each line is discussed.
- A full application is reviewed in depth so you'll have some idea how to tackle a future project.
- The code itself is excellent. The self-submitting forms presented are very efficient.
- The general, tip, and caution boxes throughout the book explain known issues and how to get the most from Spectra. (Some of the issues are fixed in Spectra 1.01.)
- An appendix lists all the Spectra tags.

A few areas of the book to pay close attention to are:

- ***Object-based development:*** This is important to understand, so read Chapter 3 a couple of times if needed.
- ***The Attribute Prefix chart:*** You'll understand why when you start to review the code. This chart is also in Chapter 3.
- ***Planning an application:*** Most of your time should be spent in the planning stages of your Spectra application. Chapter 7 discusses all aspects.
- ***Deploying a Spectra application:*** The server configuration matters. Chapter 24 has information that helps you get the most from a Spectra application.

Overall, the *Allaire Spectra E-Business Construction Kit* is positively a worthwhile reference book to add to your development library.

ANNETTE@TRILEMETRY.COM

Ask the Training Staff

Here's where you find the answers to the questions that stump you

BY
BRUCE
VAN HORN

Welcome to the second installment of this column. If you're a new reader, the purpose of the column is to give you another source for your CF-related questions.

Many of you have taken the ColdFusion classes offered by Allaire…and many of you haven't. If you have, you probably know how to get in touch with your instructor for answers to questions that may come up after the fact. However, if you haven't taken any CF classes, you can still "ask the training staff" and we'll be happy to respond.

Here are just a few of the questions that have been asked recently:

Q: *I'm having trouble inserting a date into an MS Access database table. My SQL for the insert looks like this:*

```
<CFQUERY DATASOURCE="msp"
NAME="addReg">
INSERT INTO AttendanceInfo
(StudentID,Date,FirstName,Last-
Name)
VALUES (#form.StudentID#,#Cre-
ateODBCDate(form.Date)#,'#form.
FirstName#','#form.LastName#')
</CFQUERY>
```

Can you help me?

A: The good news is that there's nothing wrong with your SQL statement. The problem is actually with your database; specifically, your database field called "date". While it's tempting (and even logical) to name a database column "date" (since that's what it's going to store), you can't do it because DATE is a reserved word in SQL. Simply rename the database column to something like "CourseDate" and then it'll work.

Q: *I'm trying to set a cookie once my user has logged in to my site (see code snippet below), but the cookie is not getting set. I've tried everything I know. What do you suggest?*

```
<!--- this code is part of
loginAction.cfm --->
<CFIF GetLogin.RecordCount gt
0>
<CFCOOKIE NAME="UserID"
VALUE="GetLogin.UserID">
<CFLOCATION URL="../index.cfm">
</CFIF>
```

A: The problem has to do with using the CFCOOKIE and CFLOCATION tags in the same section of code. This code would work if you omitted the CFLOCATION and made the user click on a link to enter the site, or if you did a client-side (JavaScript or meta) relocation to index.cfm. It would work because the cookie travels back to the browser as part of the HTTP response that goes to the client. In this case loginAction.cfm would be returned to the browser and the cookie would get set. By using the CFLOCATION tag, which is a server-side relocation, the page that sets the cookie (loginAction.cfm) never goes to the client's browser, so the cookie never gets set. The only page returned to the browser after the login is index.cfm, which doesn't have a cookie attached to it. The bottom line is this: you can't use CFCOOKIE and CFLOCATION in the same page.

Q: *How can I display a random record from our products table every time someone comes to our home page?*

A: There are several ways to do this. The easiest way is to retrieve all the records (or some subset of records) and then let CF randomly pick one from the record set. Here's an overview, but look at the code in the snippet for exact syntax. First, run a query to retrieve all the products. Next, create a variable set to a random number between one and the total of all the records returned from your query. Last, randomly output the selected product.

```
<CFQUERY DATASOURCE="dsn"
NAME="qProducts">
 SELECT ProductID,
Product_Description
 FROM Products
</CFQUERY>
<CFSET RandProd =
RandRange(1,qProducts.Record-
Count)>
<CFOUTPUT QUERY="qProducts"
STARTROW="#Variables.RandProd#
"MAXROWS="1">
 #qProducts.ProductID# - #qProd-
ucts.Product_Description#
</CFOUTPUT>
```

• • •

Please send your questions about ColdFusion (CFML, CF Server, or CF Studio) to AskCFDJ@sys-con.com.

ABOUT THE AUTHOR
Bruce Van Horn, a regular contributor to CFDJ, is president of Netsite Dynamics, LLC (an Allaire Alliance Partner), and an Allaire certified instructor.

BRUCE@NETSITEDYNAMICS.COM

EvolutionB M39

Internet Relationship Management solution to attract customers and improve response times

BY
JERRY
BRADENBAUGH

There Goes the E-Neighborhood

Browse the Web, or nearly every business or financial publication, and you'll see that e-commerce is no longer just a great idea. It's everywhere. Sales over the Web are in the trillions of dollars. Books (amazon.com), cars (iMotors.com), groceries (HomeGrocer.com) – you name it. Almost anything that retails in a store is on sale now at somebody's Web site. And why not? Doing business over the Web solves many customer- and company-related problems associated with shopping at a store's physical location (inventory, product availability, hours of operation, to list a few). However, challenges remain.

Arguably, two of the biggest challenges still facing businesses are acquiring new customers and keeping existing customers. In Q4/2000 EvolutionB will provide an Allaire Spectra-powered solution to succeed in meeting those challenges. Codenamed M39 (see Figure 1), EvolutionB defines the product as an Internet Relationship Management (IRM) solution. M39 is designed to strengthen customer–company relationships by building customer loyalty and delivering dynamic personalized content.

M39 enables business and marketing managers to better profile their customer base and the reasons they buy the products and services they do. This increased understanding allows managers to optimize marketing strategies to attract new customers as well as customize existing customers' online experiences to generate further sales. M39 is also designed to allow customer service representatives (CSRs) to assist customers with requests for information quickly and more accurately. Let's take a closer look.

FIGURE 1: The M39 service interface

Some Background

M39's functionality is provided by way of Spectra. Spectra is Allaire Corporation's packaged system solution for content management, e-commerce, and personalization. Spectra leverages the capabilities of ColdFuson to enable companies to reduce time-to-market requirements in delivering enterprise-level Web sites, intranets, and extranets. The Spectra design is twofold: transactive content and spectrum of participants that Forrester Research Group defines as "active, intelligent multimedia information delivered and composed on demand." In other words, managed content, displayed in a personalized and dynamic way.

The spectrum-of-participants design concept provides a model and methodology by which an organization can scale its Web presence to its employees, business partners, and customers. The spectrum of participants encompasses all the different types of users involved with a system. From the lower-level programming of system administrators and developers to the higher end of business managers, customers, and site affiliates, Spectra features interfaces or services to accommodate the needs of each level of user.

How M39 Works

M39's architecture is fairly straightforward. As you can see in Figure 2, it extends from the content management and personalization capabilities of Spectra, which leverages the ColdFusion application serving environment, which is tied to a Web server and NT or Solaris operating system. On the back end, M39 taps Spectra's ContentObject API (COAPI). Spectra's Webtop (the front-end user interface) enables business users, customer service representatives, and customers alike to utilize the application.

The system's functionality is based on what EvolutionB calls the Customer Relationship Life Cycle. This life cycle encompasses the evolution (sorry, I just had to) of an online consumer from casual browser to the stage of making a purchase and thereby becoming a customer who may later request product support. The cycle continues as company marketing departments and customer support centers utilize detailed information that M39 gathers about individual customers. This information is stored in the Customer Knowledge Repository, which is made available to marketing and support personnel throughout the company.

E-Customer Marketing

Marketing departments can use customer profiles from a knowledge base to launch niche marketing campaigns. Whether shoppers are just browsing or actually making a purchase, M39 gathers detailed information about their activities. M39's transactive logging provides

data for analyzing customer behaviors and modifying marketing strategies accordingly.

M39's marketing module utilizes a campaign set, containers and articles, and an intelligence engine. The campaign set contains one or more campaigns, each of which has an associated audience that is most receptive to the campaign. Each campaign and its associated articles can be displayed in any number of containers using configurable publishing rules. A container is a rectangular portion of a Web page dedicated to specific placement of marketing material. A good example of a container is a rectangle, 468 pixels wide by 60 pixels high, located at the top of a page for banner ads. Articles are graphics, text, or other media inserted into the containers.

The intelligence engine controls which articles are placed in which containers. The engine weighs user profile, marketing campaign input, and personal preferences to generate a custom page. The page will be filled with articles that attempt to match the interests of the customer and the essence of the marketing campaign.

FIGURE 2 The M39 architecture

E-Customer Service

The customer service module gives CSRs the power to more quickly and efficiently help customers once they've made a purchase. For example, when an inbound support request arrives, say, by a customer submitting an online form or sending e-mail, M39 applies it to a number of service rules (which you can create and modify). One such rule determines if the request is related to an existing incident. If so, the originating support request and its threaded history are attached to it, along with the customer's profile, for the benefit of the CSR.

Other rules apply metadata keyword searches. These searches allow a number of intelligent operations to happen. The request can be classified according to a hierarchical structure such as Books>Internet>E-Commerce>Software. M39 can then forward the request to the CSR with the most experience in that area. The keywords can also be matched against a knowledge base that can provide the requesting user with related articles that may meet the needs of his or her request.

Wrap-up

EvolutionB's M39 is designed to enable a Web site's business users to better attract new customers, build loyalty among and offer further buying opportunities to existing customers, and improve customer support accuracy and response times. M39 provides marketing, and service modules to complete the core content management, commerce, and personalization solution offered by Allaire's ColdFusion-based Spectra platform. Pricing was unavailable when this article was written. You can learn more about M39 and even apply to be a participant in its beta program by visiting www.evolutionb.com/m39.

JERRY@CYNICPRODUCTIONS.COM

USING UML, XML, Data Islands & XSL with ColdFusion

Technologies you can use to create a CF Web application

BY
JAMES BRANNAN

Separating presentation logic from data logic is perhaps one of the biggest benefits of XML and XSL. I demonstrate this benefit here with a ColdFusion application. Using the design properties of the UML (Unified Modeling Language), we'll develop a ColdFusion XML application – from requirements to coding.

XML data islands are XML data embedded in HTML documents. Internet Explorer 5, the only browser currently supporting dynamic processing of such data, uses its XML parser, MSXML.DLL, to process XML into HTML using an XSL stylesheet. You can also use MSXML on your Web server by processing the XML and inserting the results prior to returning it to the client. This allows the use of XML data islands even when the user's browser isn't IE5. In this article we'll design and build a ColdFusion application that uses these data Islands.

XML and XSL

Extensible Markup Language is similar to HyperText Markup Language in that it's tag based. But XML describes data rather than a display layout. When using XML you define your own elements (tags) and element attributes. Using a DTD (Document Type Definition), you declare the elements your XML dataset contains along with their relationships. By structuring your data using XML, the data is self-describing in a text format that makes it obtainable by any application able to parse XML documents.

The Extensible Stylesheet Language is a powerful addition to XML. XSL transforms XML into a variety of text-based display formats such as HTML or WML (Wireless Markup Language). When using XSL, modifying your display is as easy as applying a different stylesheet to the XML.

Two-Tiered Web Design

Despite increases in size, complexity, and importance, many Web applications continue being built ad hoc and without a systematic development approach. One common problem is designing an application in which display logic and application logic are combined. Jason Masterman discusses this problem at length in "The Two-Tiered Web Problem," which he defines as "mixing data access code and presentation layer code when building Web applications." For example, consider the code in Listing A.

In this code the HTML presentation logic and business logic are mixed. In larger, more complex applications this mixing of logic can prove problematic. When the business logic underlying the statement <CFIF #deptname# EQ "Marketing"><CFSET str-Location = "New York"></CFIF> changes, the ColdFusion code must also change. If other applications use the same business logic, they too must change. A simple change in business logic could require changing numerous applications.

Mixing presentation with application logic makes the work breakdown harder. Increasingly, several different groups split development. Web artists are responsible for the layout and design, content specialists are responsible for the textual HTML, and developers are responsible for data access and making the pages dynamic. Having text, display logic, and application logic in one file makes coordinating development more difficult.

Mixing presentation with application logic also increases regression testing difficulty. A change to a table cell can introduce errors to the data access logic. Conversely, changes to data access logic can introduce errors to the display. In summary, two-tiered Web sites are harder and more expensive to maintain.

A commonly touted solution to the two-tiered problem is middle-tier components. This approach moves data access logic and business logic to components such as COM, CORBA, and Enterprise JavaBeans. Since the logic is isolated in the component, multiple applications can use the component – promoting reuse. The promise of reuse has been a driving force behind components.

Components are often a pain to develop and maintain, however. Moreover, developing a COM object for no good reason does little to improve an application. COM programmers routinely develop components that obtain results using ADO (Active Data Objects), format the results into HTML, then return the HTML to an ASP page. Now the situation is worse – every time the display logic changes, the component must be recompiled and retested. The display logic was simply moved from the scripting page to the COM object – not an improvement.

You could argue that the <CFQUERY>, <CFLOOP>, and other ColdFusion tags were designed specifically for data access, and there's no reason to mess with a good thing. However, these tags don't do you much good if you're developing a Visual Basic or Java application. You'll have to rewrite the logic used in the ColdFusion application in the other language. The same is true for ASP, JSP, and other scripting languages. Also, say that the display device changes from a Web browser to a cellular phone. The ColdFusion application must be rewritten to support the new format.

It would be nice to cleanly separate the display logic from the application logic. XML combined with XSL affords you that opportunity. You can develop a ColdFusion page with little to no processing responsibilities. ColdFusion pages simply become the "glue" holding applications together when using XML, XSL, and middle-tier components. The ColdFusion page's responsibilities are limited to determining the correct components to include and when to call them. I believe this clean separation of code provides a good argument for using XML and XSL in all Web applications.

The Project

Our client wants a list of all employees in his company. It should contain the employee's name, sex, and department. He provides us with a database (see Figure 1).

The list must be viewable in Web browsers and by field personnel using WAP-enabled cell phones. He also needs to share the raw data over the Internet with collaborating organizations.

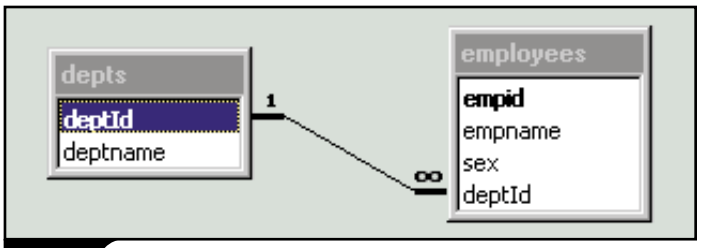


FIGURE 1: Client's employee database

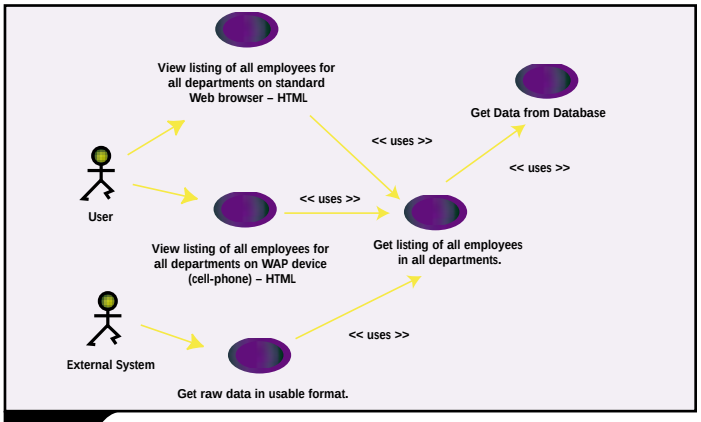


FIGURE 2: Use cases resulting from initial requirements analysis

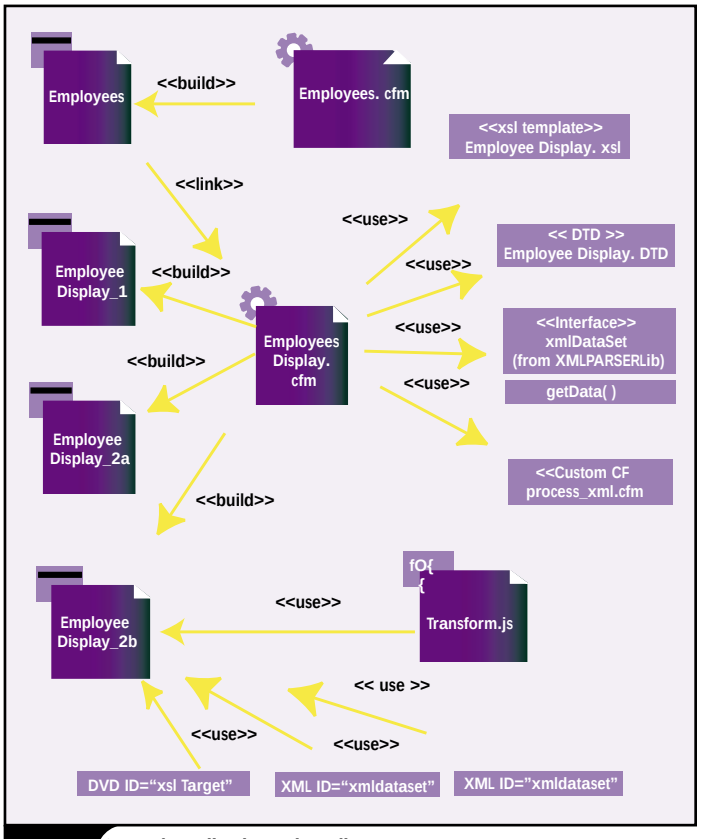


FIGURE 3: Our Web application's class diagram

The use cases resulting from requirements analysis is illustrated in Figure 2.

The UML use cases are helpful in determining your project's scope. If you've never been exposed to UML, use cases, and their applicability to Web development, a good place to start is *Building Web Applications with UML* by Jim Conallen. Another excellent introduction is *Applying Use Cases: A Practical Guide* by Geri Schneider and Jason Winters. For our purposes here, we simply

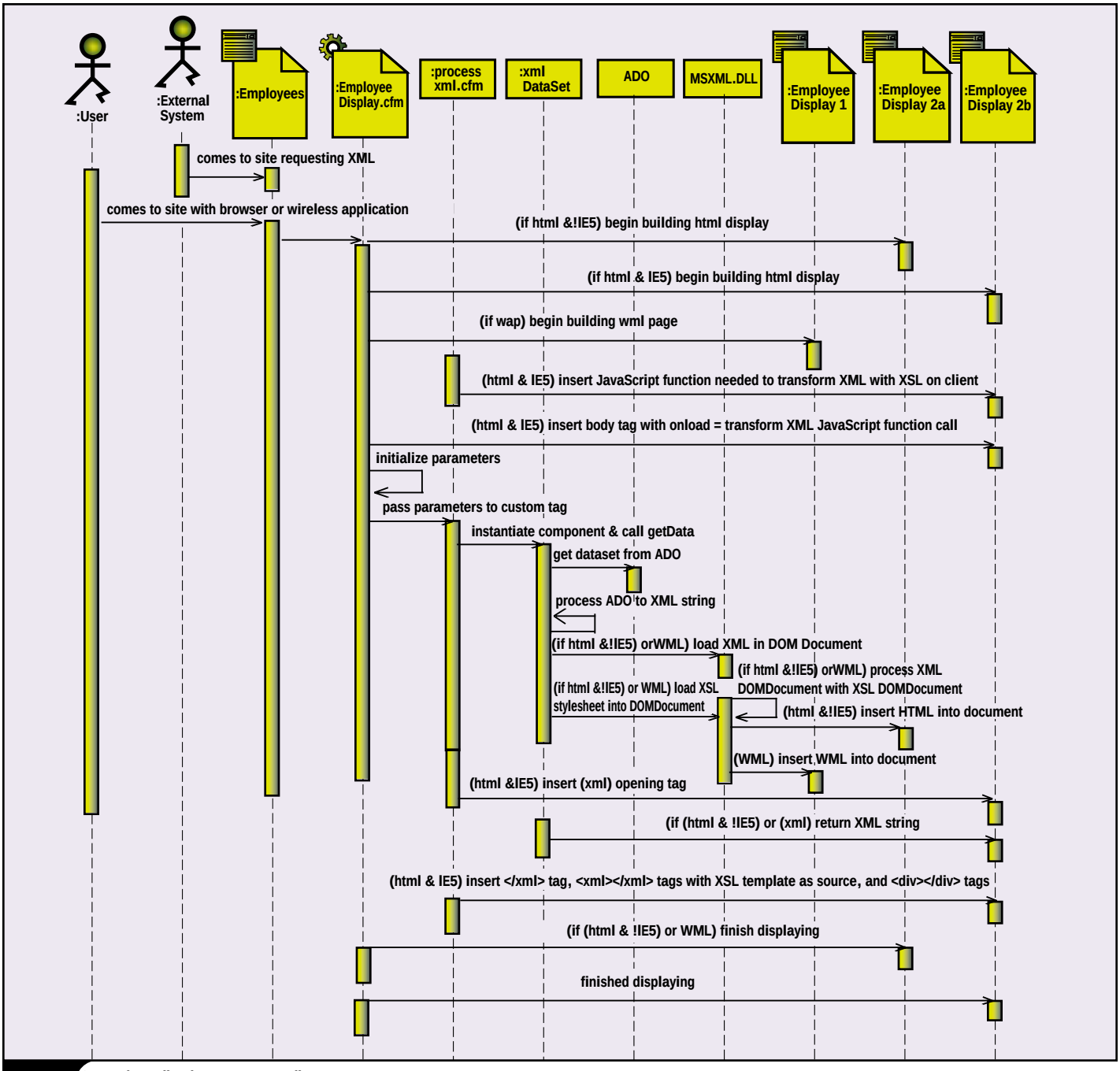


FIGURE 4: Our Web application's sequence diagram

need to know that use cases capture the interaction among Actors (a system's users) and the system. Use cases model a system's dynamic functional requirements.

What should be readily visible from Figure 2 is that displaying the information on a Web browser or handheld device and providing the raw data are logically distinct from obtaining the data from the database and creating a listing of all employees. Our use case diagram is telling us something.

Given this article's title, it's not surprising that the use case diagram is telling us to use XML and XSL. Seriously though, XML combined with XSL is well suited for this application. A middle-tier component that wraps data access logic and an XML parser can handle the use cases "Get data from database" and "Get listing of all employees in all departments." The "View listing of all employees for all departments on standard Web browser – HTML" and "View listing of all employees for all departments on WAP device – WML" are handled by XSL stylesheets. Ensuring that our wrapper component can also return XML meets the use case "Get raw data in usable format."

Figure 3 presents our application's class diagram. The notation is UML with Web Application Extensions (WAE). Note that the ColdFusion pages are represented by several pages. This is because of Server Sided Scripting Pages' dual life. On the server they have different functionality than on the client. For example, when a user comes to the Employees.cfm page, the page first must construct itself (Employees.cfm). This is done on the server. After constructing itself, the page is sent to the client (Employees). Distinguishing between the page on the server and the page on the client highlights these pages' dual nature.

The page displaying the results is more complex. Depending on what needs to be returned – WML, HTML, or XML – EmployeeDisplay.cfm builds itself differently. Employee_Display1 and

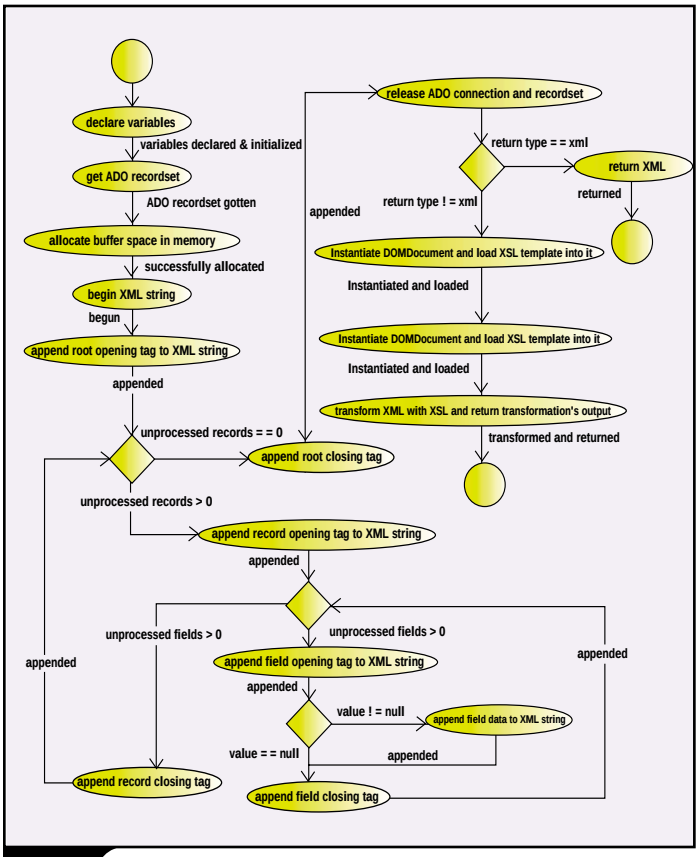


FIGURE 5: xmlDataSet's activity diagram

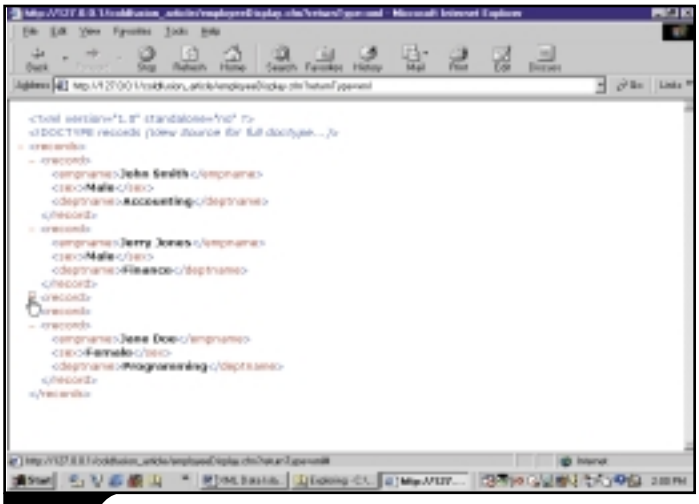


FIGURE 6: Resulting XML data

Employee_Display_2a represent the pages built when requesting WML or HTML, respectively. Employee_Display_2b represents the page built when requesting HTML and the requesting browser is IE5. In this situation the XSL processes the XML on the client. Employee_Display_3 is composed of a JavaScript file to process the XML data island (Transform.js), two XML tags, and a Div tag.

EmployeeDisplay.cfm builds itself using the specified XSL stylesheet, a DTD, and XML data from a COM object we create that wraps ADO and MSXML. Because we anticipate that getting a stylesheet and DTD and working with the COM object is something we want to repeat, we add a custom tag, process_xml.cfm, to our diagram (see Figure 4). For more information on WAE go to Rational's Web site at www.rational.com or see Jim Conallen's book mentioned above.

We should have a good idea of how our application's pieces will work together. Both Activity Diagrams and Sequence Diagrams work well in this respect. Figure 4 is a sequence diagram illustrating our site's functionality. It shows the communication among different pieces of our application as well as the message's general sequence. Note that each piece's internal behavior is a "black box" in this diagram. For example, we have no idea how xmlDataSet will process the ADO recordset to an XML string – we just know that it needs to do it. Our sequence diagram gives us a good logical overview of what all our application's pieces must do.

We now model each piece's internal behavior. A valuable technique for this behavior is UML Activity Diagrams, powerful yet simplistic. Figure 5 is an activity diagram of our xmlDataSet component that shows a start state, an end state, and a series of processes that must occur to progress from one to the other. Between each process is a guard condition. To move between processing states, a guard condition must be met. Diamonds represent logical branches. From Figure 5 we see that we're planning to loop through each record and build an XML string. Then, depending on the parameter, either return XML or process the XML with the indicated XSL stylesheet. In the interest of space I present only the Activity diagram for xmlDataSet.

Our ColdFusion Page

Listing 1 (all listings from this point appear on the ***CFDJ*** Web site) contains the code for our EmployeeDisplay.cfm page. First we set the datasource name and the query. Next we determine if a URL paramater indicates XML, WML, or HTML. If HTML, we begin an HTML page. If HTML, and the user's browser is IE5, we write a body tag containing a transformXML() method call in its onload statement. Next we choose the appropriate XSL stylesheet using the URL parameter. We pass the relevant information to the custom tag CF_process_xml. After the tag completes, if HTML, we complete the document.

Our Custom Tag

At first glance the code for process_xml.cfm in Listing 2 appears complex. However, most logic is related to <cfif> statements checking the passed parameters. First the tag ensures that the necessary parameters were passed to it; an error is thrown if any are omitted. If the user's browser is IE5, the function transformXml() was already written to the browser; in this case we write a "dummy" version of this function to the browser to avoid a JavaScript runtime error. Next we check if the user's browser is IE5 and that the URL parameter is not XML. If true, we include a script tag pointing to the JavaScript file containing the script that processes the XML on the user's browser. We then initialize several more parameters. If processing occurs on the server, we need a full path to the DTD and the XSL stylesheet. If processing occurs on the client, a relative URL is needed. After setting these parameters we instantiate xmlParser and call its getdata method, passing the connection source name, the query, the path to the DTD, and the path to the XSL template. If the component is to return XML, we simply put "rawxml" in the XSL template parameter. This method returns either HTML or an XML dataset, depending on the passed parameters.

Our C++ Component

One common way of getting XML data from a recordset is building the XML DOMDocument node by node. Another is building the XML recordset as a string, then loading it into an XML DOMDocument. In this article we opt for the second solution. If the user's browser is IE5, we don't instantiate an XML DOMDocument since we return XML. Building a string uses

fewer computer resources than instantiating an XML DOMDocument object. With the current MSXML parser (not the beta version) it has been demonstrated that loading an XML string directly into a DOMDocument is more efficient than building it node by node (see "Inside MSXML Performance" by Chris Lovett on MSDN).

Neither ColdFusion nor Visual Basic concatenate strings efficiently. In Visual Basic each time we concatenate to the string a new string is created and the old one is destroyed – using ASP it's worse – now the data type is simply a variant. I imagine ColdFusion has the same problems. We must somehow preallocate the space needed in building the XML string

A C++ ATL component, using a preallocated string class from the Standard Template Library (STL), would be both efficient and (even more important) easy to use. Listing 3 shows the relevant C++ code. By importing the MSXML and ADO header libraries, we're able to use the DLL's methods in our component via something called smart pointers. Smart pointers make it easier and safer to work with the components via pointers. As Figure 5 illustrates, we simply loop through each record and build the XML string. If rawxml must be returned, we return the XML string and are finished processing; otherwise we instantiate two MSXML DOMDocuments, load the XML into one and the XSL into the other, process the XML DOMDocument with the XSL DOMDocument, and return the resulting output.

The resulting XML data is illustrated in Figure 6. The root tag is <records>, each ado record is a <record>, and each ado field is a child element of <record>.

Space doesn't permit a full description of the C++ ATL component, but there are numerous online tutorials on the subject. My tutorial on the C++ ATL component used in this article is at www.asptoday.com. In it I explain each step in generating the C++ ATL component. The C++ project is also provided as a download at ASPToday.

Our JavaScript

Listing 4 contains the code for transform.js. This JavaScript applies the XSL stylesheet to the XML when the user's browser is IE5. Since MSXML.dll – and the DOMDocument that it loads the XML into – exists on the client, client-sided script can be written directly against the MSXML DOMDocument. The parseError is a subclass of the DOMDocument. Transforming the XML with the XSL is performed by the statement:

```
result = xmldata.transformNode(xsldata.XMLDocument);
```

The result is then inserted into the Div tag xlsTarget using the statement:

```
xslTarget.innerHTML = result;
```

Most of Listing 5's code is error handling. A word of warning: it's been my experience that uncaught errors simply end up as large blank spots where the processed XML should be. Catching the errors avoids that. Note that the try/catch block is IE5-specific JavaScript.

Our XSL Stylesheets

The XSL stylesheets (see Listings 5 and 6) simply apply formatting rules to XML data so it can be displayed. The display tags (i.e., html, wml, etc.) are mixed with the XSL tags. Norman Walsh, in *Web Techniques* magazine, provides a good casual definition for XSL. He writes, "In simplest terms, a style sheet contains instructions that tell a processor…how to translate the logical structure of a source document into a presentational structure." This is a good definition for our purposes.

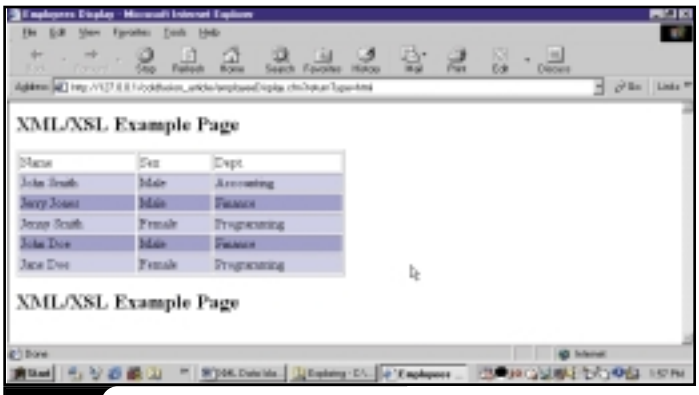


FIGURE 7A Resulting HTML display

We have two stylesheets. Listing 5 contains the code for employeeDisplay.xsl. This stylesheet is applied when HTML is to be returned. Listing 6 contains the code for employeeDisplayWML.xsl. This stylesheet is applied when WML is to be returned

Both stylesheets contain multiple templates. Templates are indicated by the tag <xsl:template match="matchingnode">. The main template, <xsl:template match="/">, matches our document's root element. Inside that template we must call another template, where the match is records. Since the XML dataset has only one records tag, this template is applied once. Inside the records template we then loop through each record using the statement <xsl:for-each select="record">.

In Listing 5 we build a table row containing three table cells for each record. The cells contain data obtained from the XML dataset using the xsl:value-of tag. Each row is colored a different shade of blue depending on whether the row is odd or even. This is done using the code:

```
<xsl:script>

 <![CDATA[function even(e) {return absoluteChildNumber(e)%2==0;}]]>
</xsl:script>
```

and

```
<tr>
<xsl:choose>
<xsl:when expr="even(this)">
<xsl:attribute name="bgcolor">#9999cc</xsl:attribute>
</xsl:when>
<xsl:otherwise>
<xsl:attribute name="bgcolor">#ccccff</xsl:attribute>
</xsl:otherwise>
</xsl:choose>
```

The function even returns a true or false, depending on whether the number is even. This function is called by the conditional statement <xsl:when expr="even(this)"> inside the xsl:choose block. The parameter this refers to is the current node, which is record. The JavaScript function then determines if the row is odd or even using the XML Document's absoluteChildNumber property. The attribute bgcolor is added to the <tr> tag and the transformed tag becomes either <tr bgcolor="#9999cc"> or <tr bgcolor="#ccccff"> (see Figure 7).

In Listing 6 we build a card for each record. Because of the limited display on a WAP device, very little can be displayed on the screen. To prevent multiple round-trips to the server, multi-

FIGURE 8: Resulting WML display

ple cards are sent to the browser. In our code we provide each card with a numeric ID calculated by the current record number. This is done using the <xsl:eval> tag combined with JavaScript. When there are no more cards, a link pointing to the beginning of the cards is presented. Note that the two templates are somewhat tricky. The code for the card at the top of Listing 6 is the last card. Notice the

```
<xsl:apply-templates
select="records/record"/>
```

just before the code for the card. This is applying the bottom template (the looping through the records) before building the last card (see Figure 8).

As with the C++ ATL component, providing a complete description of XSL or WML is outside this article's scope. For more on WML download the UPLink.com Developer's Kit from www.phone.com. For more on XSL go to MSDN – note that XSL isn't a standard yet and what works using MSXML may not work using other XSLT processors.

Our DTD

When a document is well formed, it follows the rules of XML. When a document is valid, it follows the rules of the DTD as well. Checking both sets of rules is expensive. Although a DTD slows down a parser because it also has to check the validity as well as how well formed a document is, you should weigh the relative importance of speed versus the probability of your database changing when deciding whether to use a DTD. Note that xmlDataSet doesn't support generating XML without a DTD but could be easily modified.

Our application uses a DTD for a very practical reason. By specifying the DTD in Listing 7, we assure that changes to the database or query without subsequent changes to our DTD cause our application to stop working. The line <!ELEMENT record (empname, sex, deptname)> ensures that each record must contain a name, sex, and deptname element. The DTD helps keep the configuration of the database/queries and the XSL templates the same.

Conclusion

In this article we covered a lot of ground. While creating a ColdFusion Web application, we used UML, XML, XSL, WML, and a C++ ATL component. Because of space constraints, I didn't pause to fully explain any of them. They're currently all very hot topics, and you should have no trouble locating tutorials on any of them. This article's goal was to present these technologies in a straightforward ColdFusion application.

The presented solution was biased toward a COM solution on Windows NT. Since ColdFusion now supports Java servlets, before deciding on using COM and MSXML you should also review Java-based parsers. Oracle's Java XML/XSL parser and XSQL servlet are both excellent free products. The Apache Software Foundation's Xerces/Xalan Open-Source XML/XSL parsers are also excellent choices. And Sun recently released an XSL parser. Using ColdFusion 4.5 you're no longer restricted to Microsoft's COM.

Web applications are becoming increasingly complex and intertwined with the rest of business information systems. Moreover, Internet technologies are changing at an increasingly rapid rate. Web-enabled phones will become a standard display device soon…a multitude of other devices will follow. Developing your Web applications to facilitate both reuse and easy modification of the display is becoming increasingly important. XML and XSL are two technologies that help facilitate this flexibility. Initially, scripting languages such as ASP and ColdFusion were seen as providing a thin layer of processing and display in a Web application. Now the emphasis is moving to XML and XSL/XSLT as providing that functionality. ASP and ColdFusion will still be necessary, but increasingly they'll become primarily an application's "glue," linking XML, XSL, and business components.

Listing A

```
<CFQUERY NAME="ExampleRecords"
    DATASOURCE="employees">
    SELECT * FROM employees
</CFQUERY>

<TABLE BORDER="1">
<CFLOOP QUERY="ExampleRecords">

   <CFIF #deptname# EQ "Marketing"><CFSET strLocation = "New York"></CFIF>

   <TR WIDTH="99" HEIGHT="33" ROWSPAN="2">
      <TD COLSPAN="2" WIDTH="4"><CFOUTPUT>#empname#</CFOUTPUT></TD>
      <TD COLSPAN="3" WIDTH="2" ALIGN="left"><CFOUTPUT>#sex#</CFOUTPUT></TD>
      <TD>
<CFIF deptname EQ "Accounting>
<FONT SIZE="2" COLOR="red">
</CFIF>
<CFOUTPUT>#deptname#</CFOUTPUT></FONT></TD>
      <TD><CFOUTPUT>#strLocation#</CFOUTPUT></TD>
   </TR>
</CFLOOP>
</TABLE>
```

CODE LISTING

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

About the Author

James A. Brannan is a consultant specializing in Internet programming in the Washington, DC, metropolitan area. He is also pursuing a master's degree at the University of Maryland.

brannanj@ieee.org

2000 READERS' CHOICE AWARDS

The First ColdFusion Readers' Choice Awards Acknowledge the Pursuit of Excellence in the CF Software Industry

The Best of the Year Are Here!

ColdFusion Developer's Journal proudly announces the first ColdFusion Readers' Choice Awards. It's the first CF program of its kind and we're proud to be the only publication to present these awards to top corporations. CF developers and their companies are finally being acknowledged for their product design and excellence. It is recognition that we at ***CFDJ*** feel is long overdue and well deserved. Needless to say, the competition was heated and the decisions difficult…and impossible to make without our true barometer of excellence, our readers. But after months of hard deliberation we finally came up with our winners.

How We Did It…

We offered 13 separate product categories and criteria and put them up for vote on the ***ColdFusion Developer's Journal*** Web site. The response was immediate and vocal. No one ever said that the CF professional was shy or without opinion! On the contrary, our savvy readership displayed once again that they keep their ear to the ground when it comes to trends and leading-edge product developments. They keep us on our toes, too, informing us about what they feel truly merits the CF seal of approval. These awards are merely an extension of that ongoing dialog. The stuff either works or it doesn't. Period.

Free of Outside Influences

It's out there in the field that a product earns its stripes. That's why we created the ColdFusion Readers' Choice Awards. You know that when a product and/or company displays the award, they have passed a rigorous and painstaking inspection of their product or service. Free of outside influences, the awards are a true reflection of what ColdFusion developers are using, what they like…and in many cases what they don't! What we've come up with at the conclusion is an extremely clear picture, one that has several things we expected – as well as a few surprises....

ColdFusion Readers' Choice Awards Recognizes 13 Winners and 26 Finalists During an Awards Ceremony in Washington, DC, at the Worldwide Allaire Developer Conference, November 5–8.

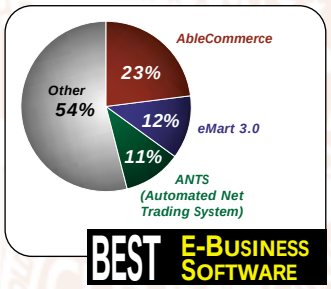
As the world's leading publication targeting CF professionals, ***ColdFusion Developer's Journal*** announced at the Worldwide Allaire Developer Conference the results of its annual awards program. These awards recognize the best software products providing business solutions with ColdFusion.

The awards were presented on November 6, 2000, to the winners while **SYS-CON Radio** brought the event live to millions in the software industry and the readers of ***CFDJ***.

Congratulations!

Congratulations to all this year's category winners and finalists! Many thanks to the readers of ***ColdFusion Developer's Journal*** who were instrumental in making these awards possible: they were the elite of CF professionals who cast their votes on our CF Web site during this past year and helped with the difficult decisions. It was an impressive field, but the best of the best eventually rose to the top. These awards are the only ColdFusion awards given in the industry, and what sets them apart from other software awards is that they are nominated by peers: the awards are given only to superior products nominated and selected by the largest number of professional ColdFusion developers ever assembled for such a purpose. In fact, an astounding 13,827 of these professionals cast their votes to help us select the winning products.

Once again, congratulations! **SYS-CON Media** wishes each of these companies and its employees continued excellence in the software industry. Following are the products that you, our readers, selected as this year's first-ever ColdFusion winners.



— Winner —
ABLECOMMERCE

from AbleCommerce

AbleCommerce enables users to quickly generate secure online storefronts with a robust feature-set including international currencies, bundled product "Kits," automated shipping, and multi-tier taxation. Additional capabilities include cross-selling, membership pricing, and the ability to create targeted discounts. The Source Code version offers developers exceptional scalability, adaptability, and the flexibility to meet future needs. AbleCommerce is the flexible solution for building electronic commerce.

www.ablecommerce.com

— 1st Runner-up —
eMART 3.0

from ecom enterprises, inc.

eMart is a high-end e-commerce solution for companies that are serious about e-business. Unlike an "out-of-the-box" solution, eMart is based on the highly customizable eTechnology code library. This code library allows eMart to be tailored to a company's exact needs and specifications. eMart provides rapid deployment with the scalability, reliability, and high-end performance necessary for a successful e-commerce application. eMart belongs to the eTechnology family of products created by ecom enterprises, inc.

www.ecomenterprises.com

— 2nd Runner-up —
ANTS

from WebRaven

The **Automated Net Trading System** allows merchants to establish and easily maintain an online trading presence. Enhanced features within ANTS enable B2B and B2C e-commerce models to be effectively implemented by merchants in order to meet individual customer needs. The ANTS supplier, product, customer, and order management facilities effectively automate the supply chain process and provide merchants with standard and statistical reports on customer and trading activity.

www.webraven.com

— Winner —
FIG LEAF SOFTWARE

Comprised of a team of award-winning developers, analysts, and authors, Fig Leaf Software's Consulting Services Group has been repeatedly recognized as the technology leader for both Allaire and Macromedia platform development.

Working in a team-based environment led by a senior mentor, our staff members are able to create and grow in a way that allows them to become true "masters" at their trade.

This dynamic and creative environment allows us to develop applications that consistently exceed our clients' expectations.

www.figleaf.com

— 1st Runner-up —
ECOM ENTERPRISES

ecom enterprises is an innovative, results-oriented Internet solutions company that offers proven Internet solutions. Our expert team of designers, engineers, and copywriters have been sought after by companies interested in expanding and/or improving their Internet presence without incurring the costs of an in-house development team. With ecom enterprises, your company can realize its Internet marketing goals with confi-

dence. ecom enterprises also offers a library of highly customizable e-business solutions including eMartT, eContentT, eCalendarT, and eRegistryT.

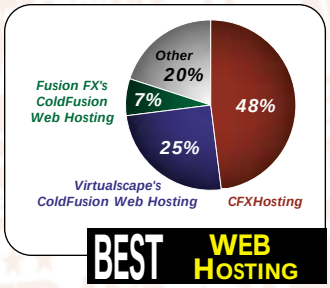
www.ecomenterprises.com

— 2nd Runner-up —
SYNTEGRA

Syntegra is the global e-business consulting and integration arm of British Telecommunications, one of the largest telecommunications companies in the world. We create winners in the digital economy by helping organizations seize business opportunities through the creative application of new technology.

Syntegra specializes in the design, development, and support of enterprise, Internet, and systems-integration solutions, Web applications, and e-commerce outsourcing services. With custom-designed solutions and services, Syntegra is the e-business engine for some of the world's largest corporations.

www.syntegra.com



BEST WEB HOSTING

— Winner —
CFXHOSTING

from cfxhosting

Welcome to CFXHosting.com – the Internet's leading hosting provider of ColdFusion Web sites. With over 35,000 ColdFusion enabled-Web sites hosted on our servers, you can have peace of mind - knowing your Web hosting company was founded by a team of ColdFusion professionals! CFXHosting.com offers the only unrestricted ColdFusion development environment that does not require your own dedicated server. For just $99.95 per month, you'll have completely unrestricted access to develop your applications using Allaire's ColdFusion!

www.cfxhosting.com

— 1st Runner-up —
COLDFUSION WEB HOSTING

from Virtualscape, Inc.

As one of the first Premier Allaire Hosting Partners, Virtualscape has provided ColdFusion hosting since early 1996. The experience we have gained has helped us create one of the most stable and affordable ColdFusion environments anywhere. We understand that hosting a database-centric Web site is far more difficult than hosting static Web pages. For ColdFusion to operate at peak efficiency, it needs powerful servers and our custom-built machines are up to the task.

Even with all this power, we realize ColdFusion may still experience problems even though the Web server may be operating correctly. This is why we have set up a three-tiered monitoring system. The first layer is an expert system that not only monitors the servers, but can automatically repair problems without a technician's assistance. In fact, 90% of server problems can be solved directly by our monitoring system. However, if the problem cannot be solved directly by the expert system, a technician can be alerted 24 hours a day, 7 days a week, to manually repair any problems. Just to be safe, we have an outside monitoring system (Red Alert) that conducts its own independent tests on every one of our servers and alerts our technical staff to any undetected problems. This last line of defense can detect problems that our internal system may miss.

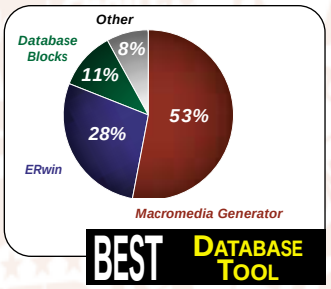
Finally, not only does Virtualscape host ColdFusion Web sites, we develop them as well. This means we can answer most questions developers may have. As an alpha tester for Allaire, we are able to test the latest features of ColdFusion and know how they may affect our customers' Web sites before other Web hosts. Most important, our intimate knowledge of ColdFusion allows us to configure our systems to best take advantage of the power contained within ColdFusion.

www.virtualscape.com

— 2nd Runner-up —
COLDFUSION WEB HOSTING
from Fusion FX, Inc.

FusionFX, Inc., and **CF Hosting, Inc.,** wish to thank ***CFDJ*** for sponsoring a Readers' Choice Award to recognize the best providers of ColdFusion hosting. The people responsible for delivering this award-winning hosting for FusionFX and are now known as CF Hosting, Inc., The UN.COMmon Hosting Company, are dedicated to providing award-winning ColdFusion hosting solutions. FusionFX would like to honor CF Hosting, Inc., for making this award possible and encourage everyone to consider CF Hosting, Inc. (www.cfhosting.com), for all ColdFusion hosting or consulting needs. CF Hosting, Inc., can be reached at 770 982-8382.

www.fusionfx.com



BEST DATABASE TOOL

— Winner —
MACROMEDIA GENERATOR
from Macromedia

Macromedia Generator 2 Enterprise Edition is the only scalable, high-performance solution for companies delivering dynamic visual content to audiences who require fast, time-sensitive, and personalized Web sites. Generator automates the delivery of targeted and customized graphical content creating a personal, one-to-one experience for each end user.

www.macromedia.com

— 1st Runner-up —
ERWIN-DATABASE DESIGN MADE EASY
from Computer Associates

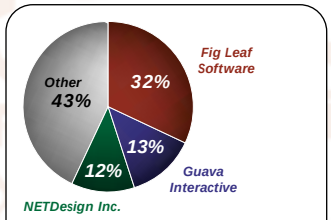
ERwin is an industry-leading data modeling solution that creates and maintains databases, data warehouses, and enterprise data resource models. ER*win* models visualize data structures in a manner that assists in organizing, managing, and even mitigating the complexities of data, database technologies, and the deployment environment. Databases are not only developed faster, their quality and maintainability are dramatically improved.

www.ca.com

— 2nd Runner-up —
DATABASE BLOCKS
from Productivity Enhancement, Inc.

DatabaseBlocks quickly generates ColdFusion source code for a complete relational database interface with search, sort, list, report, add, edit, view, delete, and drill up and drill down capabilities. If you want to bring your projects in profitably, on time, and within budget, then having DatabaseBlocks in your toolbox is a must. Version 2 adds many new features and is over 100 times faster!

www.commerceblocks.com



BEST DESIGN SERVICE

— Winner —
FIG LEAF

Fig Leaf's Creative Media Department has designed and developed engaging, Web-based media for some of the most recognized companies and organizations in the country. Comprised of a talented team of award-winning artists,

programmers, writers, and instructional technologists, our staff strives to create ingenious and innovative solutions.

Our Graphic Website Design, Interactive Media Design, Corporate Branding, and Usability Consulting services have proven critical in building successful online solutions.

www.figleaf.com

— 1st Runner-up —
GUAVA INTERACTIVE

Established in 1995 with offices in Sydney and Melbourne, **Guava Interactive** has become a world leader in the Web development industry, as evidenced by the clients we attract.

We believe that for every problem there is a simple solution, and Guava Interactive enjoys the challenge when people say: "IT can't be done on the Web." We believe it can. And we show how. Guava Interactive has created dynamic, innovative, user-friendly, and customized Web solutions to a range of industry sectors including government, small to medium businesses, clubs, and associations. In addition, Guava Interactive devotes many resources to developing cutting-edge e-commerce solutions, in order to help our clients maximize their online business solutions.

Guava Interactive takes into consideration the existing business models of corporations and offers them the opportunity to upgrade onto the Internet with a fully dynamic Web site. Through the Internet, companies can establish direct links with their customers and stakeholders as well as explore business opportunities to develop and deliver new products and services for new customers. Our assistance in your Web site creation will ensure that it is:

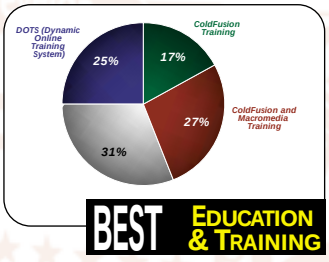
- Commercial in nature and deployment
- Produces the desired image for your business
- Cost effective to operate and maintain
- Easily accessible to Web visitors
- Clear and logically set out
- Relevant, concise, and easily updated

www.guavainteractive.com.au

— 2nd Runner-up —
NETDESIGN INC.

NETDesign Inc. is a nationally recognized Web developer. Our brand name, CFDev.com, is recognized across the country as a great source for developer-related information and tools. Our newest brand, Linkway.com, is a small business solution built using ColdFusion. From very large custom solutions to very small dynamic solutions, NETDesign exists in nearly every market in nearly every corner of the U.S. and in Israel and Britain and growing.

www.netdesigninc.com



— Winner —
COLDFUSION & MACROMEDIA TRAINING
from Fig Leaf Software

Grow your skills with certified Macromedia and Allaire training courses from a recognized and respected industry leader. Our developers/instructors receive stellar ratings by providing "real-world" insights and creative solutions to your development challenges. Fig Leaf has coauthored official courseware for both Macromedia (Generator, Flash 5) and Allaire (Spectra). With unparalleled qualifications, experience, and innovative leadership we've got you covered.

www.figleaf.com

— 1st Runner-up —
DOTS
from WebRaven

The Dynamic Online Training System is complete online training environment that is available as a packaged product to any corporation that needs to train staff, members, new clients, etc. It offers complete flexibility to

build your own online courseware using multimedia components such as videos, sound clips, images, PDF files, and animated GIFS.

DOTS contains a complete assessment engine, including the ability to build tests made up of different types of questions and to offer randomized tests. Students can immediately see their results after taking a test, and trainers can quickly and easily grade subjective questions and recalculate course grades. DOTS offers complete grade weighting and calculates grades automatically based on the ones that are entered into the system. DOTS even allows trainers to conduct traditional classroom courses supplemented with online materials, grading, and attendance tracking. In addition, DOTS manages certifications and automatically suggests appropriate courses to students that need to renew particular certifications.

DOTS is built into WAMS, WebRaven's intranet software, which allows even greater flexibility as all users can interact and share information in an intranet environment. It includes a comprehensive user management system and the ability to import user details from other programs.

Features

- Can be used through a Web browser with no external plug-ins or downloads
- Students view their grades on their desktop and monitor their own progress
- Online grading sheets are automatically filled in and totals calculated
- Online attendance tracking
- Automatic certification tracking
- Search for users with/without specific certifications
- Question and answer facility to aid student learning
- Discussion facility where students can discuss courseware
- Automatic generation of Online Course catalog; students can enroll online
- Mail merge facilities
- Easily create tests and assignments
- Creation of courseware can be outsourced
- Password expiry after use inactivity at workstation
- Complete logging of activities of users
- Multilanguage support
- Updatable news bulletins

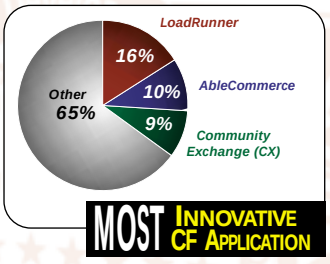
www.webraven.com.au

— 2nd Runner-up —
COLDFUSION TRAINING

from Highlander Software, Ltd.

Highlander Software is an Allaire Premier Partner and Authorized Training Center based in the UK. Highlander runs a range of Allaire Authorized courses including FastTrack to ColdFusion, Advanced ColdFusion Development, and Spectra for Developers, as well as offers consulting/mentoring services and reselling the whole Allaire product range.

www.highlander.co.uk



— Winner —
LOADRUNNER

from Mercury Interactive

LoadRunner is the industry-standard load-testing tool used to predict system behavior and performance. It exercises an entire Web infrastructure by emulating thousands of users to identify and isolate problems. LoadRunner's integrated real-time monitors enable organizations to minimize test cycles, optimize performance, and accelerate deployment.

www.heva.mercuryinteractive.com

— 1st Runner-up —
ABLECOMMERCE

from AbleCommerce

AbleCommerce enables users to quickly generate secure, online storefronts with a robust feature set including international currencies, bundled product "Kits," automated shipping, and multitier taxation. Additional capabilities include cross-selling, membership pricing, and the ability to create targeted discounts. The Source Code version offers developers exceptional scalability, adaptabili-

ty, and the flexibility to meet future needs. AbleCommerce is

the flexible solution for building electronic commerce.

www.ablecommerce.com

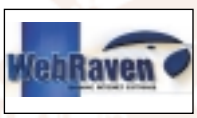
— 2nd Runner-up —
COMMUNITY EXCHANGE (CX)

from WebRaven

The Virtual Community – Community Exchange application is a revolutionary off-the-shelf Web application that integrates users across a physical or interest-related community.

Members of the community can join any number of community interest groups.

By administering news bulletins, discussion groups, surveys, event calendars, document libraries, and individual Web pages, members maintain the content that's available for other community members to view.



The Virtual Community includes security and management functionality to ensure that usage is logged and reported on, and only appropriate content is published.

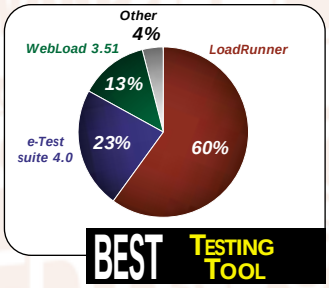
The structure of the community is hierarchical, but it is designed so that any one person can fit into many places in the hierarchy. Users would belong to groups that reflect the neighborhoods they live in and the activities they enjoy. They may be assigned to administer one or more groups.

The Virtual Community software can be used successfully not only as a public community, but also in corporate communities.

Features

- Provides groups where users can interact with others with similar interests.
- Users can join as many groups as they like and leave them at any time.
- Group members can access news, documents, live discussions, event calendars, and surveys that relate to that group.
- Group members can subscribe to information from a group, requesting that all information be sent to them directly via e-mail.
- Groups become "mini-communities" that are self-managed by group members. Designated managers have veto power over information posted to the group.
- Each group can create a Web site using the built-in Web site builder, or a manager can simply upload a Web site created in another program. Each group has a unique URL for its Web site.
- Each user has a calendar on which events are posted by the groups to which they belong. Users can also add personal events to their calendar that only they see.
- Contains a built-in e-mail program. Simply enter your e-mail account settings and use the program to send and receive e-mail. Users can keep their own address book.
- Users can keep a task list or "to-do" list.
- Users can create their own Web site, either by using the built-in Web site builder or by uploading one created in another program. Users have a unique URL for their Web site.
- Manage banner advertising effectively by setting the size of the banner, and choosing where to display it and whether to display associated text below the banner. Reports will show how often the banner is being displayed and give its click-through rate.
- A comprehensive log file search facility is available with reports displaying information about the user, type of action taken, and the date. Log files can also be archived to save space.
- Master administrators can delegate administrative functions to other users throughout the community.
- Create multiple revenue streams through banner advertising and corporate sponsorship by actively promoting your community to other related sites or interest groups.

www.webraven.com.au



— Winner —
LOADRUNNER

from Mercury Interactive

LoadRunner is the industry standard load-testing tool used to predict system behavior and performance. It exercises an entire Web infrastructure by emulating thousands of users to identify and isolate problems. LoadRunner's integrated real-time monitors enable organizations to minimize test cycles, optimize performance, and accelerate deployment.

www-heva.mercuryinteractive.com

— 1st Runner-up —
E-TEST SUITE 4.0

from RSW Software

The **RSW e-TEST suite** is the only easy-to-use, seamlessly integrated test solution on the market optimized for Web applications. e-TEST suite includes e-TESTER for functional/regression testing, e-LOAD for load/scalability testing, and e-MONITOR for 7x24 quality monitoring of deployed applications. All three tools are powered by a common set of Visual Scripts and require no programming. There are no proprietary languages to learn, no special proxies to set up, and no training classes required. As your application changes, any differences in your tests are highlighted in the Visual Scripts and can be automatically updated in place. That means that your regression, load, and monitoring tests will always stay synchronized with your application, and you can make automated testing a routine part of your Web development process.

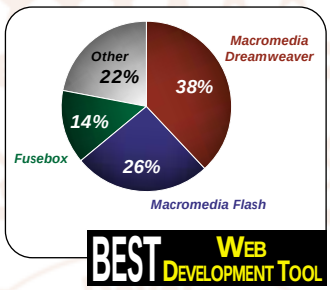
www.rswsoftware.com

— 2nd Runner-up —
WEBLOAD 3.51

from RadView Software

WebLoad provides users with a precise, easy-to-use, and cost-effective solution for verifying Web applications. WebLoad integrates performance, scalability, and integrity verification into a single process for shortened development cycles and unmatched verification of Web application prior to deployment.

www.radview.com



— Winner —
DREAMWEAVER 3

from Macromedia

Professionals choose **Dreamweaver** to develop Web sites, automate production, and enhance team efficiency. Control your code with Roundtrip HTML and the revolutionary Quick Tag Editor, and accelerate workflow with integration with Web applications, Microsoft Office, and leading e-commerce and application servers. Only Dreamweaver can be customized using HTML, JavaScript, and XML, giving you the power you need for rapid Web development.

www.macromedia.com

— 1st Runner-up —
MACROMEDIA FLASH 5

from Macromedia

Design and deliver distinctive, low-bandwidth Web sites with **Macromedia Flash 5**, the professional standard for producing high-impact Web experiences, used by over half a million Web authors worldwide. Designers easily create engaging graphics with the familiar Macromedia user interface, and developers build advanced Web applications

using scripting, forms, and server-side connectivity.

www.macromedia.com

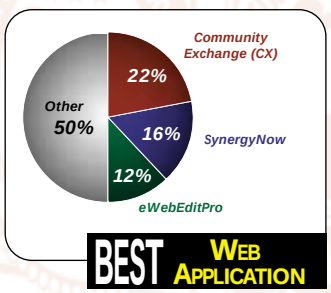
— 2nd Runner-up —
FUSEBOX

from ANTHC

Would you try and build a house without sitting down and deciding how many rooms you need and where they should be? Would you try to start construction without blueprints? No. So, why would you try to build an application without a framework?

Fusebox is a free framework created by the collaborative research of over 200 professional Web developers. The result is a simple yet elegant structure that will work in any Web application.

www.anthc.org



— Winner —
COMMUNITY EXCHANGE (CX)

from WebRaven

The Virtual Community – Community Exchange application is a revolutionary off-the-shelf Web application that integrates users across a physical or interest-related community.

Members of the community can join any number of community interest groups.

By administering news bulletins, discussion groups, surveys, event calendars, document libraries, and individual Web pages, members maintain the content that's available for other community members to view.

The Virtual Community includes security and management functionality to ensure that usage is logged and reported on, and only appropriate content is published.

The structure of the community is hierarchical, but it is designed so that any one person can fit into many places in the hierarchy. Users would belong to groups that reflect the neighborhoods they live in and the activities they enjoy. They may be assigned to administer one or more groups.

The Virtual Community software can be used successfully not only as a public community, but also in corporate communities.

Features

- Provides groups where users can interact with others with similar interests.
- Users can join as many groups as they like and leave them at any time.
- Group members can access news, documents, live discussions, event calendars, and surveys that relate to that group.
- Group members can subscribe to information from a group, requesting that all information be sent to them directly via e-mail.
- Groups become "mini-communities" that are self-managed by group members. Designated managers have veto power over information posted to the group.
- Each group can create a Web site using the built-in Web site builder, or a manager can simply upload a Web site created in another program. Each group has a unique URL for their Web site.
- Each user has a calendar on which events are posted by the groups to which they belong. Users can also add personal events to their calendar that only they see.
- Contains a built-in e-mail program. Simply enter your e-mail account settings and use the program to send and receive e-mail. Users can keep their own address book.
- Users can keep a task list or "to-do" list.
- Users can create their own Web site, either by using the built-in Web site builder or by uploading one created in another program. Users have a unique URL for their Web site.
- Manage banner advertising effectively by setting the size of the banner and choosing where to display it and whether to display associated text below the banner. Reports will show how often the banner is being displayed and give its click-through rate.
- A comprehensive log file search facility is available with reports displaying information about the user, type of action taken, and the date. Log files can also be archived to save space.
- Master administrators can delegate administrative functions to other users throughout the community.
- Create multiple revenue streams through banner advertising and corporate sponsorship by actively promoting your community to other related sites or interest groups.

www.webraven.com.au

— 1st Runner-up —
SYNERGYNOW

from evolutionB (formerly Catouzer)

SynergyNow is a complete solution for developing and hosting run-anywhere applications. It delivers breakthrough value to ASPs by dramatically reducing their deployment cost and time-to-market while giving them a large and growing selection of ready-to-rent applications for the wired and wireless Web.

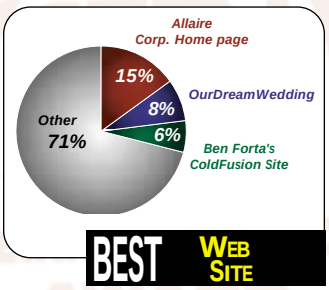
www.evolutionb.com

— 2nd Runner-up —
EWEBEDITPRO

from Ektron, Inc.

This first-of-a-kind, browser-based (Netscape and IE) WYSIWYG Web editing tool, **eWebEditPro,** allows organizations to leverage their dynamic Web servers by enabling business end users to create, edit, and publish content directly to the Web via the browser. Includes support for fonts, tables, spell checking, images, hyperlinks, and more. IT professionals maintain control by configuring the solution to ensure that the site's look and feel remain intact. eWebEditPro easily replaces a text area field and enables Webmasters/developers to be removed from time-consuming HTML and FTP tasks. Now XHTML compliant – MS Office 2000 filters remove extraneous tags to produce well-formed HTML and Netscape-compatible content.

www.ektron.com



— Winner —
ALLAIRE CORP. HOME PAGE

from Allaire

Allaire Corporation is a leading provider of Internet software products and services for companies building their businesses on the Web. Supported by a community of 450,000 developers and global network partners, the Allaire Business Platform enables tens of thousands of companies worldwide to seize new business opportunities by creating e-commerce, content management, customer service, and business automation systems. Headquartered in Newton, Massachusetts, Allaire has offices in Europe and Asia Pacific and can be found on the World Wide Web at www.allaire.com.

www.allaire.com

— 1st Runner-up —
OURDREAMWEDDING–THE WEDDING RESOURCE

from ecom enterprises, inc.

Welcome to **OurDreamWedding** and congratulations on your upcoming wedding! Let OurDreamWedding be there every step of the way during this important and exciting time to assist you with your wedding planning needs. OurDreamWedding offers helpful and informative resources for all brides and grooms. We hope you find everything you need

here to plan your dream wedding.

Throughout these pages you will be able to skim through dozens of wedding gowns, bridesmaid gowns, flower girl dresses, fashions for the mothers of the bride and groom as well as find helpful fashion articles to assist you in choosing your wedding gown.

Whether you're getting married or looking for a wedding gift, you've come to the right place. OurDreamWedding.com offers a vast selection of top quality items. Forming a registry and shopping for gifts is fast, simple, convenient and it's all here at OurDreamWedding.

www.ourdreamwedding.com

— 2nd Runner-up —
BEN FORTA'S COLDFUSION SITE

www.forta.com is one of the most popular online destinations for ColdFusion enthusiasts (and future enthusiasts). Hosted by Ben Forta, Allaire's Product Evangelist and author of the best-selling *ColdFusion Web Application Construction Kit*, this site features a tip-of-the-day, ISP listings, lists of CF-powered sites, presentations, columns, and other valuable resources. Support is also provided for Ben's books, and there are links to other books that will be of interest to CF developers.

www.forta.com

— Winner —
ADVANCED COLDFUSION 4.0 APPLICATION DEVELOPMENT

By Ben Forta
Macmillan

Written as an advanced companion to Ben Forta's *ColdFusion 4.0 Web Application Construction Kit*, ***Advanced ColdFusion 4.0 Application Development*** is a topical guide to the more sophisticated aspects of ColdFusion 4.0.

The book is organized into four independent parts – scalability, security, extending ColdFusion, and advanced application development. Scalability covers the various ways to keep track of your server performance, scale up configurations, and keep track of clients. This first part also covers Cluster Cats – a load-balancing solution bundled with the Enterprise version of ColdFusion 4.

In the security section, the author discusses the User Authentication Framework and security sandboxes and explains how to integrate ColdFusion security with your network operating system. The third part focuses on how to modify and extend the ColdFusion environment to truly make the most of it. It begins with an extensive discussion on custom and CFX tags and then covers integration with Delphi, CORBA, Visual Tool Markup Language (VTML), and customization of the development environment.

The final part is a rapid-fire discussion of many key areas: Web Dynamic Data Exchange, code sharing with Visual Basic and Perl, scripting, intelligent agents, and interactions with the System Registry. The book wraps up with useful appendixes that provide references for tags, functions, VTML, the Wizard Markup Language (WIZML), and the DTD file format. A companion CD-ROM provides a trial version of ColdFusion to make this definitive work complete.

www.mcp.com

— 1st Runner-up —
THE COLDFUSION 4 WEB APPLICATION CONSTRUCTION KIT

By Ben Forta
Macmillan

Using Allaire's ColdFusion 4, the newest version of this popular Web server and development environment, requires careful

guidance. Ben Forta has been writing the best ColdFusion instruction and reference books for years, and this latest edition of ***The ColdFusion 4.0 Web Application Construction Kit*** is the finest guide to date.

Using ColdFusion to its full potential requires a fairly broad knowledge base, which Forta helps you develop. He begins with simple database-query applications and moves on to full-featured electronic-commerce systems. He includes instructions for using the ColdFusion tools, of course, but you'll also discover tutorials and references on the ColdFusion Markup Language (CFML), SQL, and the Verity search language. Even if you've never used any language other than HTML, you'll feel comfortable learning from this book.

The illustrative examples in *The ColdFusion 4.0 Web Application Construction Kit* really shine. Whether you want to know how to perform a three-table join, cache queries, or do any of hundreds of other tasks, there's a code listing for you in these pages. You'll also find some plug-and-play applications, including an online store that's ready for customization and deployment. All examples appear on the companion CD-ROM with evaluation versions of ColdFusion Studio and ColdFusion Server.

www.mcp.com

— 2nd Runner-up —

MASTERING COLDFUSION 4

By Arman Danesh and Kristin Motlagh

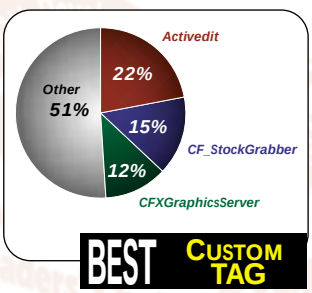
Sybex

This site is designed to supplement the recently released book ***Mastering ColdFusion 4*** from Sybex. Written by Arman Danesh, Kristin Motlagh, and several expert contributors, ***Mastering ColdFusion 4*** will teach you to develop exciting interactive Web sites using the ColdFusion Application Server from Allaire.

The book begins with clear, concise coverage of the basics to help turn newcomers into productive users as quickly as possible. From there, it proceeds to intermediate topics certain to expand your knowledge and skills, then concludes with some of the most advanced ColdFusion instruction you'll find anywhere. Regardless of your experience or needs, you'll find what you're looking for including information on:

- Getting data from a database
- Writing data to a database
- Including outside code
- Creating and manipulating variables
- Using SQL for advanced, dynamic database interaction
- Creating and validating ColdFusion forms
- Using the ColdFusion Web Application Framework
- Interacting with email, Web, FTP, and LDAP servers
- Implementing error control
- Scheduling automatic execution of ColdFusion programs
- Implementing fine-grained, advanced security
- Creating custom tags
- Using WDDX
- Administering ColdFusion servers

www.juxta.com



BEST CUSTOM TAG

— Winner —

ACTIVEDIT

from CFDevcom

Activedit is a tool for ColdFusion developers that enables them to embed a word processor similar to MS Word into a Web page. The developer using Activedit can then empower users of the site to publish dynamic Web pages by simply clicking the save button. Allaire recognized Activedit as a cool tag and, more recently, CFDev, creator of Activedit, as a useful e-Zine. Activedit will be upgraded to 2.0 before Christmas 2000.

www.cfdvc.com

— 1st Runner-up —

CF_STOCKGRABBER

from Amkor Technology

CF_StockGrabber is a custom CFML tag that allows a user to pass a single stock, mutual fund, or index ticker symbol or a comma-delimited list of stock, mutual fund, or index ticker symbols for processing. The tag takes the symbols and queries Yahoo's www.quote.yahoo.com site using CFHTTP. The results are returned to the user as variables for further processing. Because the tag makes use of CFHTTP, the user never has to leave the site calling the tag. In addition, the tag is extremely useful for gathering stock quote information on an automated basis using ColdFusion's scheduling engine. The uses of this tag are virtually unlimited and demonstrate the power of intelligent agents within the ColdFusion framework.

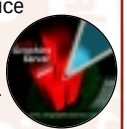
www.amkor.com

— 2nd Runner-up —

CFXGRAPHICSSERVER

from Internet Vision Software Ltd.

CFXGraphicsServer is a unique, 100% server-side, high-performance graphing and charting engine for Allaire's ColdFusion that enables Web developers to produce professional-quality graphs and charts from real-time database sources.



CFXGraphicsServer requires no additional client-side controls such as ActiveX or Java and incorporates over 30 graph types and nearly 200 configurable graph attributes, including full "Point & Click" drill-down capability to graphically present complex data at various levels of detail.

www.cfxgraphicsserver.com