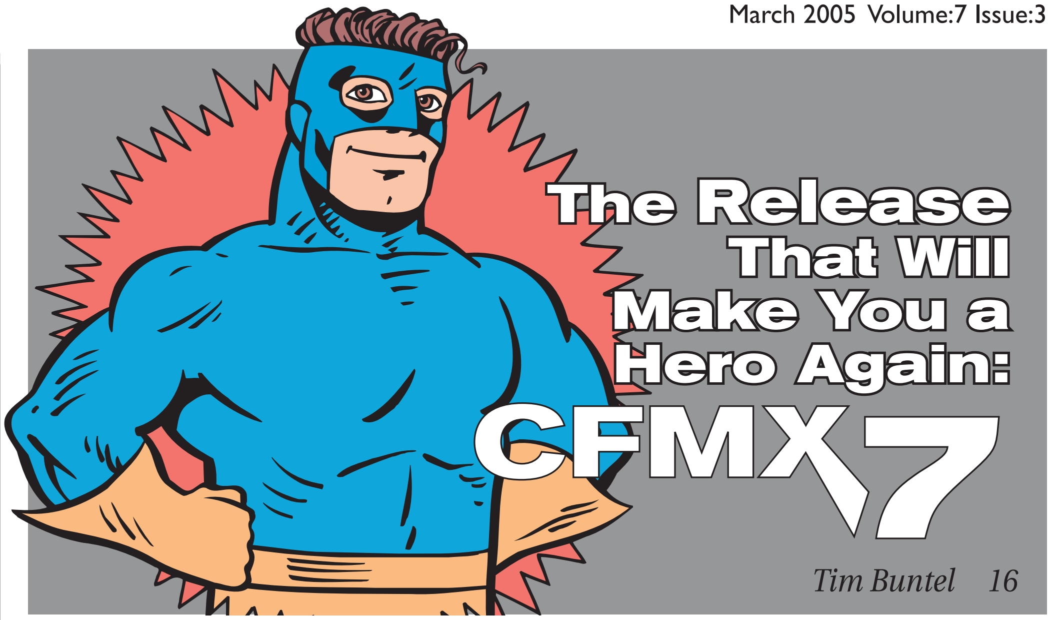
CREATE A BLOG, WIN COLDFUSION MX 7 PG 41

ColdFusion Developer's Journal

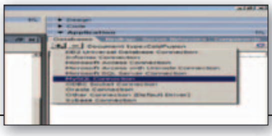
ColdFusionJournal.com

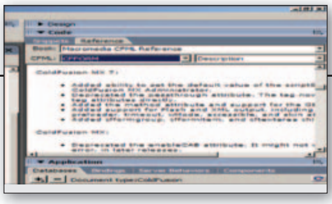
March 2005 Volume:7 Issue:3

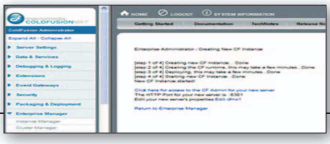


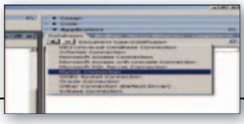
The Release That Will Make You a Hero Again: CFMX 7

Tim Buntel 16









RETAILERS PLEASE DISPLAY UNTIL MAY 31, 2005

$9.99US $9.99CAN

04

0 09281 02689 1

SYS-CON MEDIA

macromedia®

We figured it was about time that
web video stopped looking like web video.

This is a scene from one of the world's best websites, with video playing next to traditional web animation. The look of full-screen video, without ugly web players or pop-ups.

To tour the site—and see the future of video—visit:
macromedia.com/go/video5

Nothing beats our racks Absolutely nothing

CARRIER CLASS DATA CENTERS

Highly Secure Guarded Facility
24/7/365 Network Operations Center
24/7/365 Technical Support
Redundant Conditioned Air Systems
Redundant Fiber Entry Points
Multiple Uninterruptible UPS Systems
Multiple 1250 KW Generators Onsite
VESDA Early Warning Smoke Detection

Robert Marsh, Head Surfer

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from **$299***

Instant Activation!

Dedicated Server

Dual Xeon 2.4 GHz
2 GB RAM • 2 x 73 GB SCSI HD
Remote Console • Remote Reboot
2000 GB Monthly Transfer Included

Over 20,000 Servers!

1-800-504-SURF | ev1servers.net

PLESK7 RELOADED
Preferred Control Panel

IP Compliant. Price subject to change. Quantities Limited.
*Per month. Set-Up fees apply. See web site for complete details.

ARE YOU READY FOR

MX7

CFDynamics

1.866.233.9626

- 24/7 network monitoring
- Reseller program
- Comprehensive on line support
- State of the art data center

editorial offices

SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645
Telephone: 201 802-3000 Fax: 201 782-9638
COLDFUSION DEVELOPER'S JOURNAL (ISSN #1523-9101)
is published monthly (12 times a year)
by SYS-CON Publications, Inc.

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

It's Here

By Simon Horwith

It's finally here! ColdFusion MX 7 was released about an hour prior to this writing. This release is the most customer driven release of ColdFusion to date - new features for delivering printable content in the form of portable (FlashPaper or PDF) documents or reports, an event gateway that allows the server to communicate with any system or device via Java gateways (including the popular SMS gateway), an application event gateway, new enhancements to including X-Forms support, Flash Forms, and a ton of new validation support, enhancements to its XML support and Verity engine, much improved internationalisation support, and much, much more. You don't have to take my word for it - you can read all about these new features in this and upcoming issues of *CFDJ*.

I've spent the past 2 months or so planning a deep-focus issue for the release of ColdFusion MX 7 (formerly known as Blackstone). In this issue we've got several articles introducing the new features to our readers. But that's not all! This issue is also introducing several new columns and changes to our format that will regularly appear in *CFDJ* each month. Most notable is the new "Macromedia Speaks Out" column. Each month, a Macromedia employee(s) will write an article for CFDJ. This month is the first installment - Tim Buntel, product manager for ColdFusion at Macromedia, has written an excellent article introducing the new version of the server. Over the next several months, members of the ColdFusion Development Team and Q/A Team are going to write articles about the features that they developed first hand. In addition to articles from developers and Q/A engineers, future instalments will include articles from Macromedia Support, Training, Press, User Groups, Team Macromedia, Macromedia Executives, and more!

This month's issue is also going to begin the regular focus on a different case study each month. Learning about how other developers and companies are using ColdFusion helps us all to learn, to develop new creative ideas, and also gives us ammunition to help show prospective clients and employers how other companies are using it and that they should as well.

Speaking of how other developers are using ColdFusion, I am extremely excited about seeing how developers put the new features in ColdFusion MX 7 to use. So, with this in mind I am introducing another new column, "Developer's Challenge." In this column, I will present a puzzle or a challenge with a set of requirements for the readers to code a solution to. I will judge the entries and each month the column will present the solution that "won" and pose the next challenge. Oh, did I mention that there would be prizes? Yes, there will be a prize to go with each challenge - and I'm not talking about lame T-shirts and Frisbees either. To find out more and to find out what this month's challenge and prize are, read the article.

Finally, in order to spend more time focussing on the new "Developer's Challenge" column and to spend more time focusing on my duties as editor-in-chief, Ray Camden has graciously agreed to take over the community focus column which I've been writing for nearly three years now. From time to time, I'm sure I'll still contribute articles about community activities, but I wish Ray luck!

So - if you haven't already downloaded ColdFusion MX 7, do so now! You're going to love the new features! As developers discover new uses and develop new techniques for the new features in ColdFusion MX 7, I expect plenty of new article submissions. .

About the Author

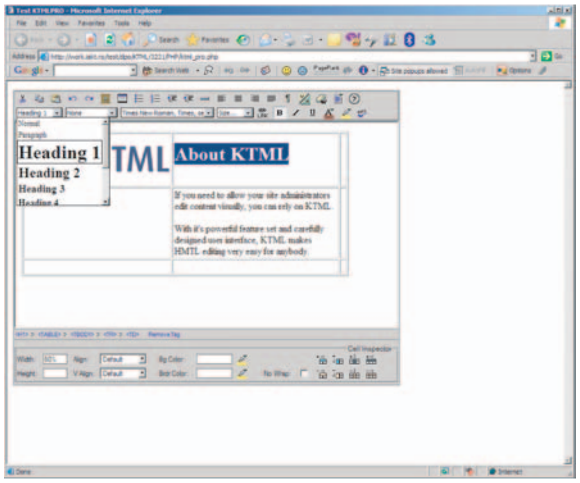
Simon Horwith is the editor-in-chief of ColdFusion Developer's Journal. *Simon is a Macromedia Certified Master Instructor and a member of Team Macromedia. He has also been a contributing author of several books and technical papers. You can read his blog at www.horwith.com*
simon@horwith.com

Web (r)evolution:

KTML

Let your clients edit their Web sites content through the browser

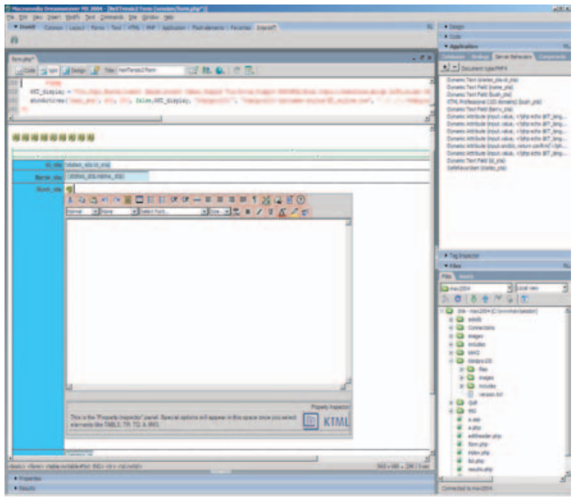
Visually use CSS styles (Word like)



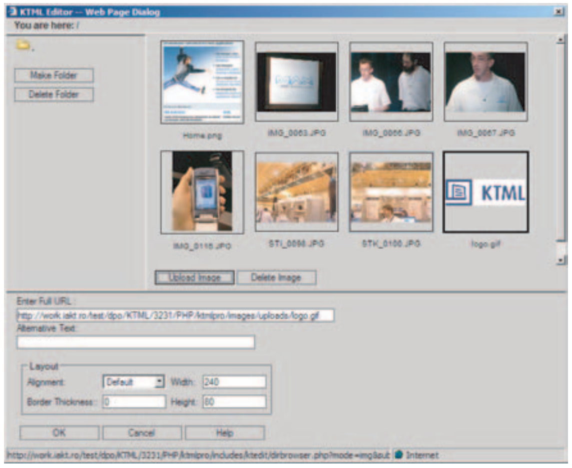
If you need to allow your site administrators edit content visually, you can rely on KTML.

With it's powerful feature set and carefully designed user interface, KTML makes HMTL editing very easy for anybody.

Native Dreamweaver MX integration

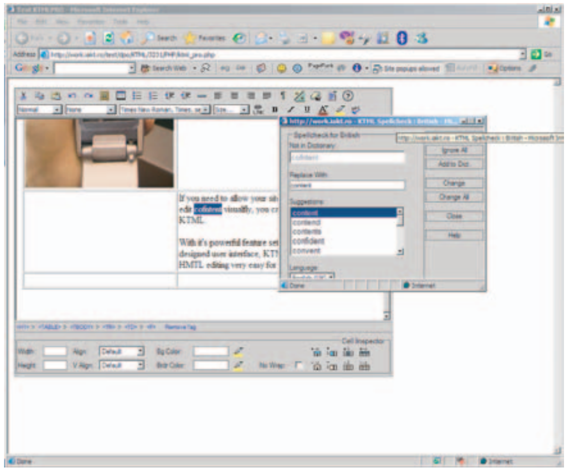


Easy-to-use image explorer



http://work.iakt.ro/test/dpo/KTML/3231/PHP/ktmlpro/images/uploads/logo.gif

Multi-language spell-checker



KTML 3.5 - HTML online editor

See all the features, benefits and a live demo at:

http://ktml.interaktonline.com/

Interakt

macromedia ALLIANCE PARTNER

Supported Technologies:

SYS-CON MEDIA

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

vp, business development
Grisha Davida grisha@sys-con.com

group publisher
Jeremy Geelan jeremy@sys-con.com

advertising

senior vp, sales & marketing
Carmen Gonzalez carmen@sys-con.com

vp, sales & marketing
Miles Silverman miles@sys-con.com

advertising director
Robyn Forma robyn@sys-con.com

advertising manager
Megan Mussa megan@sys-con.com

sales & marketing director
Dennis Leavey dennis@sys-con.com

associate sales managers
Kristin Kuhnle kristin@sys-con.com
Dorothy Gil dorothy@sys-con.com
Kim Hughes kim@sys-con.com

sys-con events

president, events
Grisha Davida grisha@sys-con.com

national sales manager
Jim Hanchrow jimh@sys-con.com

customer relations

circulation service coordinators
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com
Monique Floyd monique@sys-con.com

manager, jdj store
Brunilda Staropoli bruni@sys-con.com

sys-con.com

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

web designers
Stephen Kilmurray stephen@sys-con.com
Matthew Pollotta matthew@sys-con.com

online editor
Martin Wezdecki martin@sys-con.com

accounting

financial analyst
Joan LaRose joan@sys-con.com

accounts payable
Betty White betty@sys-con.com

credits & accounts receivable
Stephen Michelin smichelin@sys-con.com

subscriptions

Subscribe@sys-con.com
Call 1-888-303-5282

For subscriptions and requests for bulk orders, please send your letters to Subscription Department
Cover Price: $8.99/issue
Domestic: $89.99/yr (12 issues)
Canada/Mexico: $99.99/yr
All other countries $129.99/yr
(U.S. Banks or Money Orders)
Back issues: $12 U.S. $15 all others

Advanced Deployment with Ant (and DbUnit)

Put yourself in control of your development process

By Harry Klein

Apache Ant is a subproject being developed as part of the Jakarta project. Ant is a Java-based build tool which uses XML build files. Its main competitors are GNU Make, Jam, and NAnt. Many developers prefer Ant because it is environment-neutral, tightly integrated into Java, and faster than Make. XML build files are easier to create, read, and maintain than build files used with the Make utility (or other similar utilities).

The last ***CFDJ*** Ant article by John Ashenfelter (*CFDJ*, Vol. 6, issue 2) covered some basic Ant tasks that you would likely perform during the development process. Generalized, he used Ant to pull out the source code from VSS (Visual SourceSafe), make a build directory, and deploy it to the distribution directory.

In this article, I'll discuss some of Ant's more advanced features that are available in Ant's two task packages: the core task package and the optional task package. The core package contains tasks for building, packaging, and deploying projects, as well as tasks to execute SQL statements and tasks for integrating with Concurrent Versions System (CVS). The optional package extends Ant's functionality in a variety of ways, such as integrating with other source control systems and running unit tests using JUnit.

I will also provide an example of how to use the DbUnit Ant task. DbUnit is a JUnit extension targeted for database-driven projects that, among other things, puts your database into a known state between test runs.

The checksum sample application shows how to call the Ant framework through ColdFusion. Finally, I'll provide some Ant productivity tips.

Installing DbUnit

Read John Ashenfelter's Ant article or the Ant installation documentation located at http://ant.apache.org/manual/install.html in order to install Ant. Check that your environment variables are set correctly. Here is an example for Windows systems:

- Path: C:\apache-ant-1.6.2\bin
- ANT_HOME: C:\apache-ant-1.6.2
- JAVA_HOME: C:\j2sdk1.4.2\jre

The install process for DbUnit is very easy. First you have to download the DbUnit package from the DbUnit homepage (see resources). Then unpack the package and add the DbUnit jar to Ant's classpath. In a typical windows installation the path would be: C:\apache-ant-1.6.2\lib\ dbunit-2.1.jar

After this step you can use DbUnit by adding a <taskdef > element to your build script as follows:

```
<taskdef name="dbunit" classname="org.dbunit.ant.
DbUnitTask"/>
```

Use the task in the rest of the build file.

I will explain this in more detail in the DbUnit Integration section.

A Typical Ant Project

In the first example I will walk through a simplified variant of an Ant file my company uses to create hotfix packages. See sys-con.com/coldfusion/sourcec.cfm (sourcecode1/build_cvs.xml) for the complete code; I will focus only on some important parts.

The file is divided into three parts: in the first two parts properties and patternsets are defined. The final part contains the Ant targets.

Properties

The properties are found first under the <project> element. These property elements are like variables. You can use the value of a property anywhere within a project and most commonly by tasks defined in <target> elements by using the expression ${property-name}.

For instance I am defining properties for the work directories, CVS settings, and ColdFusion cfencode options.

It is possible to define or overwrite properties when you invoke Ant at the command line with the syntax -Dproperty=value. For example, if you wished to set the value of "cvs_password" to "mypassword," you would type ant -Dcvs_password=mypassword. For security reasons, you should not hard code your password in the build file.

Patternsets

FileSets are groups of files. These files can be found in a directory tree starting in a base directory and are matched by patterns. Patterns can be grouped to sets and later be referenced by their ID attribute. They are defined via a patternset element, which appears later in the targets section nested into a FileSet. Patterns can be specified by nested <include>, or <exclude> elements.

In our example we are excluding unnecessary files through the patternset "patternExcludeUnneededDirectoriesAndFiles." We are also using a patternset with the iID "notEncoded". As the name says, files using this patternset should not be cfencoded.

Targets and Tasks

Target elements define a set of instructions for carrying out some kind of work. They may be compared to program functions that are themselves constructed of tasks (comparable to programming instructions). For instance, in our example, the target named "initDirs" (the <target> element with attribute name= initDirs) contains multiple mkdir tasks to create the work folders. The target element name is chosen by the creator, but the task names are fixed and match those defined in the Ant tag reference. The target "deploy" depends on several other targets, which means that if the target "deploy" is invoked, it will call the other targets first. When calling Ant, you can pass a target to run on the command line. For instance, by invoking Ant with the command line Ant deploy, you would run the target "deploy". If you call Ant without specifying a target, the target specified in the project's default attribute will be launched.

An Ant build file is a collection of targets dealing with a given stage of the project build or deployment. Here are the targets used in our sample build file:

1. *deploy:* This is the default target.
2. *initDirs:* Creates the work directories using the task <mkdir>.
3. checkout: Exports the module from CVS. This target uses the cvspass task to write the cvsroot of the repository and the password to access it in the .cvspass file. The cvs task must authenticate on the CVS server, using data in the .cvspass file located in the home directory of the user.
   You can find details about the CVS task at http://ant.apache.org/manual/CoreTasks/cvs.html.

4. *dateFilter:* Filters files by date using the tasks fileset and date (property $filter_date).
5. *deleteUnneededFilesAndDirectories:* Deletes unneeded files and directories using the patternset defined in section two.
6. *encodeFiles:* Encodes files using the task exec and the Macromedia command-line tool cfencode.exe. At the end of this target we use the patternset "notEncoded" to copy the unencoded custom tags over the encoded ones.
7. *buildDistributions:* Copies the files to the distribution directory.
7. *zipFiles:* Zips the files using the task zip. You can find details about the zip task at http://ant.apache.org/manual/CoreTasks/zip.html
9. *makeFileList:* Builds file lists using the task exec and the DOS command line tool tree.exe.
10. *cleanup:* Removes work directories.

DbUnit Integration

DbUnit is a test framework for developing unit tests inside your projects. The DbUnit task allows you to execute these tests from your Ant build file. Thus, using the DbUnit task, you can easily run all of your unit tests every time you build or deploy your project. DbUnit has the ability to export and import your database data to and from XML datasets. DbUnit also helps you to verify that your database data match expected set of values.

The following targets are used in the sample DbUnit build file, which you can find at sys-con.com/coldfusion/sourcec.cfm (sourcecode2/build_dbunit.xml):

1. *exportxml:* Export database data to XML.
2. *exportdtd:* Export database structure to DTD.
3. *exportdata:* Partial database data export. Export two tables: table1, resulting from specified query and table2 entire content.
4. *importdata:* Import data defined in file data.xml.
5. *deletedata:* Delete data defined in file data.xml.
6. *comparedata:* Compare partial data from query and xml file against the database.

I specified the Microsoft JDBC driver for Microsoft SQL Server "com.microsoft.jdbc.sqlserver.SQLServerDriver" for all DbUnit tasks. You can download this driver from the Microsoft Web site.

Building a Checksum Application with Ant and ColdFusion

This example (sourcecode3) consists of four files. See sys-con.com/coldfusion/sourcec.cfm for the complete code.

1. *ant_checksum.xml:* Ant file containing only one target called "md5". This target uses the checksum task (s. http://ant.apache.org/manual/CoreTasks/checksum.html) to generate checksums for ColdFusion *.cfm and *.cfc files.

– continued on page 24

Seapine Software™
changing the world of software development

Complete source code and asset management in Dreamweaver MX—now possible with Surround SCM.

Dreamweaver users know a beautiful Web-based product is only skin deep. Underneath, it's a tangle of hundreds or thousands of ever changing source files. Without a good development process and strong tools, bad things happen. Surround SCM can help.

Surround SCM lets you...

Track multiple versions of your source files and easily compare and merge source code changes.

Check out files for exclusive use or work in private workspaces when collaborating on a team.

Automatically notify team members of changes to source files—push changes through your organization.

View complete audit trails of which files changed, why, and by whom.

Associate source code changes with feature requests, defects or change requests (requires additional purchase of TestTrack Pro).

Remotely access your source code repository from Dreamweaver MX.

Surround SCM adds flexible source code and digital asset control, powerful version control, and secure remote file access to Dreamweaver MX. Whether you are a team of one or one hundred, Surround SCM makes it easier to manage your source files, letting you focus on creating beautiful Web-based products.

Features:

Complete source code and digital asset control with private workspaces, automatic merging, role-based security and more.

IDE integration with Dreamweaver MX, JBuilder, Visual Studio, and other leading Web development tools.

Fast and secure remote access to your source files—work from anywhere.

Advanced branching and email notifications put you in complete control of your process.

External application triggers let you integrate Surround SCM into your Web site and product development processes.

Support for comprehensive issue management with TestTrack Pro—link changes to change requests, bug reports, feature requests and more.

Scalable and reliable cross-platform, client/server solution supports Windows, Linux, Solaris, and Mac OS X.

Want to learn more?

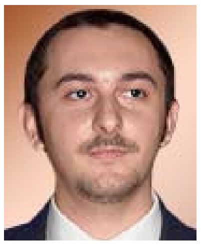
Download Seapine's just-released white paper, **Change Management and Dreamweaver**, to discover tips, tricks and best practices for achieving full software configuration functionality. Visit www.seapine.com/whitepaper.php and enter code WM0604 to receive your copy today.

www.seapine.com
1-888-683-6456

Surround SCM Source Control
TestTrack Pro Defect Management
QA Wizard Automated Testing
macromedia ALLIANCE PARTNER
software development 13th annual product excellence award

Dreamweaver Reloaded

How Macromedia improved Dreamweaver for CFMX 7 development



By Alexandru Costin

ColdFusion development is one of the few places with an undergoing IDE preference war (we'll ignore the war between VI and EMACS that has its roots in the last century). People like either CFStudio, or Homesite, of CFEclipse – but the best known player is Macromedia with its Dreamweaver platform.

Still, some highly vocal CF developers don't like Dreamweaver, for three reasons. First, they are hardcore programmers, and Dreamweaver is a tool mostly for designers; second, they see in Dreamweaver some glitches that prevent them from working as they would like to; and third, they've built their own code editors for their specific needs. If you're not one of those guys, this article is for you (let's not start another flame war now).

Introduction

In ColdFusion MX 7, Macromedia heard developer complaints and tried to catch up by adding features to Dreamweaver specifically for ColdFusion programmers. They have chosen to work with us (InterAKT) on providing a solid and innovative solution in extending Dreamweaver to cover their needs.

I will walk you through the improvements so you can see if you think Dreamweaver is the right tool for you, as they target both the programmer and the newbie web developer.

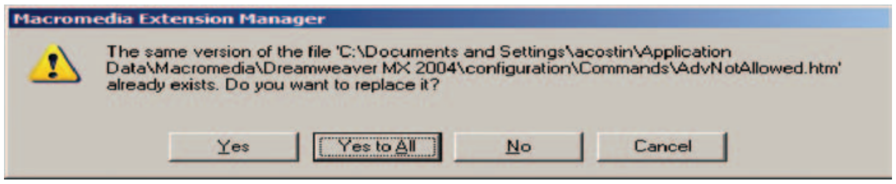
You will find better code completion for the new ColdFusion MX 7 tags, and also an inline reference to help you code faster. Debugging and creating datasources is also integrated natively in Dreamweaver, so you won't have to keep a window open in the CFIDE Administration section. For the newbie programmer, there will be wizards to help create CFCs and reuse CFQUERIES between pages, together with visual property inspectors for the CFFORM tags.

Prerequisites

Let's see what's needed to install and play with the ColdFusion extensions. You need:

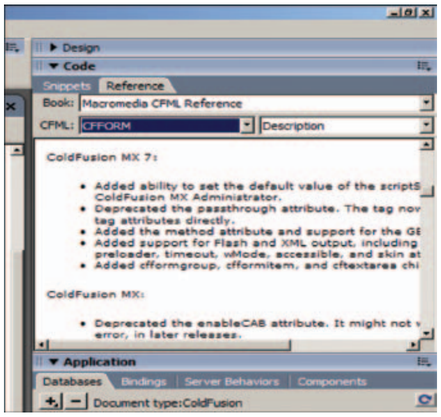
1. ColdFusion MX 7, which can be downloaded at: http://www.macromedia.com/software/coldfusion/trial/
2. Dreamweaver MX 2004 (Dreamweaver MX is not supported). You can download a trial version from http://www.macromedia.com/go/trydreamweaver.
3. CFMX 7 Extensions, which you can download from: http://download.macromedia.com/pub/developer/cfmx_extensions.zip (you will be also asked if you want to install those extensions during the ColdFusion MX 7 installation).

After installing ColdFusion MX 7 and Dreamweaver MX 2004, if


Macromedia Extension Manager

The same version of the file 'C:\Documents and Settings\acostin\Application Data\Macromedia\Dreamweaver MX 2004\configuration\Commands\AdvNotAllowed.htm' already exists. Do you want to replace it?

Figure 1: You can safely choose "Yes to all"



Book: Macromedia CFML Reference
CFML: CFFORM   Description

ColdFusion MX 7:

- Added ability to set the default value of the scriptS ColdFusion MX Administrator.
- Deprecated the passthrough attribute. The tag nov tag attributes directly.
- Added the method attribute and support for the GE
- Added support for Flash and XML output, including preloader, timeout, wMode, accessible, and skin at
- Added cfformgroup, cfformitem, and cftextarea chi

ColdFusion MX:

- Deprecated the enableCAB attribute. It might not v error, in later releases.

Document type:ColdFusion

Figure 2: The Code panel, Reference tab

you haven't installed the CFMX 7 Extensions yet, you can do so by double clicking on the CFMX7.mxp file, that will open the Extension Manager and auto-install in Dreamweaver.

During the install process, you might be asked to overwrite some files. You can safely choose "Yes to all" (see Figure 1) – as we've previously mentioned, Dreamweaver received a facelift, and this was only possible with some surgery to improve its core. And don't worry; you'll be able to uninstall the extension later on, restoring Dreamweaver to its original state.

In order to use some of the new features, you'll also need to have RDS access to your server. I know that this might be an issue, but when you see the things Dreamweaver can do to save you time once it has access to the ColdFusion server, you'll surely appreciate it.

You can start Dreamweaver now and see what was improved to help you code better.

Features for Newbies

From its inception back in '95, ColdFusion was a language that made development easy for HTML designers. While several major versions along the way focused alternatively on designers or developers, Macromedia never forgot the newbie programmers, and included in the 7.0 release improvements to help them create powerful stuff, without super-programming skills.

Integrated Reference library

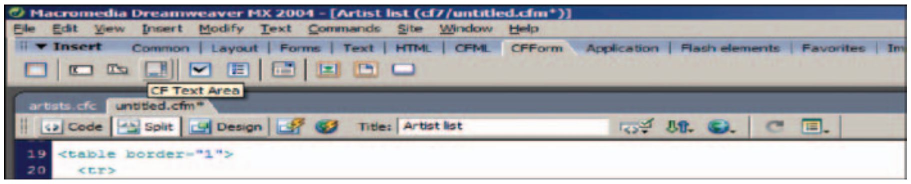
As ColdFusion MX 7 brings a lot of improvements, a new reference was required in order to help you learn the new platform features. The ColdFusion MX Extensions add this new reference to Dreamweaver, and you can use them in a similar way with the other references (see Figure 2).

CFFORM Native Support

ColdFusion MX 7 also brings a lot of improvements in CFForms – things like better validation, Flash or XForms output – so it became necessary to support CFForms in Dreamweaver more fully. This means that you now have access to an Insert Panel that will add the CFForm tags in the page, that the tags will be rendered visually in the Dreamweaver Design View and also that you'll be able to change all the tag attributes using intuitive Property Inspectors and Tag Editors.

First, let's take a look at the new CFForm Insert Panel (see Figure 3). It's located in the Dreamweaver Insert Panel, near the regular FORM counterparts.

Improvements have been also brought to the way CFFORM elements are rendered in Design View (see Figure 4) – as



```
<table border="1">
  <tr>
```

Figure 3: The new CFForm Insert Panel

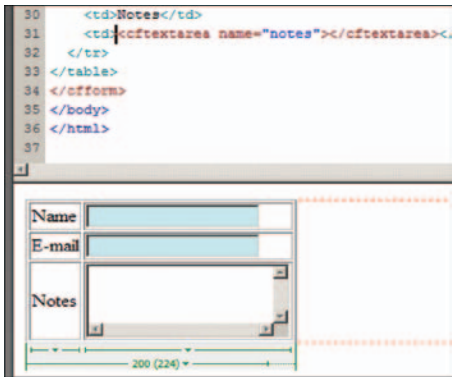
some of them were completely ignored previously.

Another powerful improvement consists in the new and up-to-date property inspectors that will allow you to change the most common properties of a CFFORM tag from Dreamweaver. The new validation attributes added in ColdFusion MX 7 are editable here, along with more generic ones.

And if you want to have access to the full list of attributes for a tag, you will surely appreciate the new CFForm Tag Editors (see Figure 5). They let you change all the parameters that can be sent to a CFFORM element, working both in code view and design view. It's now up to you to use Dreamweaver to create powerful ColdFusion MX 7 forms.

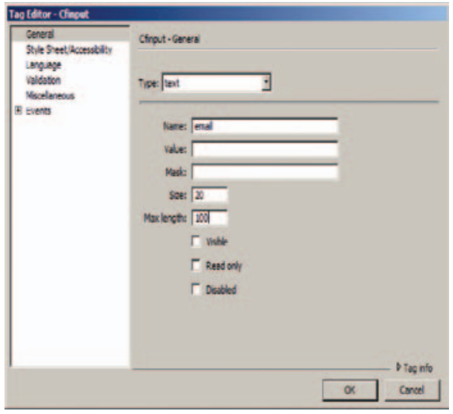
Datasource Management

ColdFusion really shows its muscles when used to create dynamic sites – that is, querying the database and creating HTML pages on the fly. Until now, datasource management was a slightly complex pro-



```
    <td>Notes</td>
    <td><cftextarea name="notes"></cftextarea><
  </tr>
</table>
</cfform>
</body>
</html>
```

Figure 4: CFFORM in Design view


Tag Editor - Cfinput

Cfinput - General

Type: text

Name: email
Value:
Mask:
Size: 20
Max length: 100
☐ Visible
☐ Read only
☐ Disabled

Figure 5: The new CFForm Tag Editors

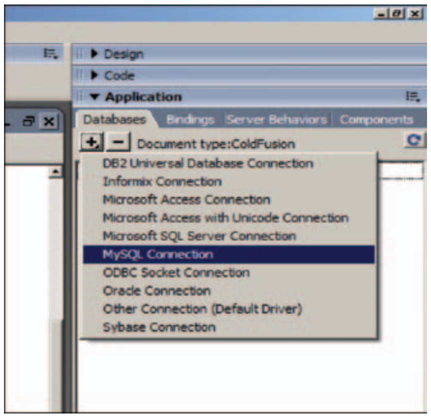
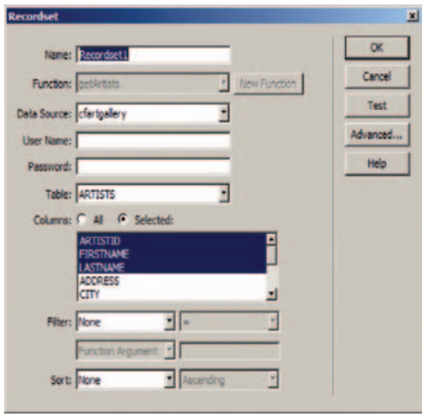
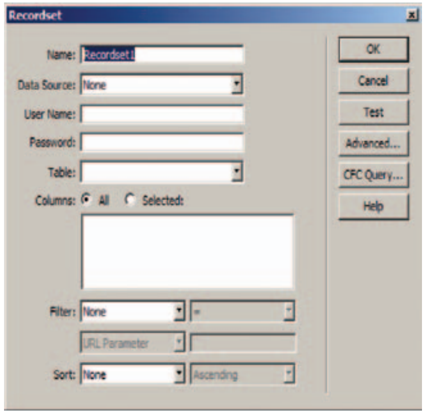


Figure 6: Creating a new datasource



Name: Recordset1
Function: getArtists
Data Source: cfartgallery
User Name:
Password:
Table: ARTISTS
Columns: ( ) All (•) Selected:
ARTISTID
FIRSTNAME
LASTNAME
ADDRESS
CITY
Filter: None =
Function Argument
Sort: None Ascending

Figure 7: The CFC Recordset user interface



Name: Recordset1
Data Source: None
User Name:
Password:
Table:
Columns: (•) All ( ) Selected:
Filter: None =
URL Parameter
Sort: None Ascending

Figure 8: The new CFC Query button

cess, as it requires a browser opened in the CFIDE Administration section. Now, it is very simple from Dreamweaver to create new datasources, in a way similar to the rest of the server models. You just have to click the "+" button in the Application panel, the Database tab, and you can create a new datasource without leaving Dreamweaver (see Figure 6).

The user interface for adding a new datasource is very intuitive as it resembles with the original version from the CF Administrator. You can even configure advanced settings for creating a new datasource.

CFC Queries and Recordsets

The most significant feature included in the ColdFusion MX 7 extensions is the possibility to create reusable CFC Queries in the Dreamweaver visual way. We'll not document one more time why components are good, as there are many articles and books about this matter. The basics are simple: components help you reuse code between pages, and implement complex things with simple usage interfaces.

With the CFMX 7 extensions Macromedia wanted to make components easy to be used by anybody – so the product development team focused on a simple workflow, creating recordsets in CFCs and reusing them in multiple pages. For example, when building a web application to list art items, you will need the Artists query in multiple pages – both in the front end, and in the back end. If you choose to add the query to the Artists table in all your pages that need it, then when your client will ask you to change the Artists' order (for example, ascending after name), you will have to modify all your pages to reflect the change. If you put the query to Artists in a Component (CFC file), you'll be able to change the Artists' order only once, and it will be reflected in all your pages that use that component.

In order to demonstrate this new Dreamweaver functionality, let's create a component that will return in one of its method a query with all the artists. After creating an empty component, you will be able to easily add a CFC Recordset to it, by using the Dreamweaver "New Recordset" functionality. It will automatically detect that you are in a CFC file, and it will start the CFC Recordset user interface (see Figure 7).

This interface is pretty similar to a regular Recordset one, with one major addition: it allows you to specify a function where this recordset will be added. It also allows you to add a new empty function in your component if you need to. When applying the CFC Recordset, a CFQUERY will be added at the end of the selected function, insuring that the CFQUERY will be returned.

After the CFC is created, Dreamweaver lets you call it from any CFM page, also in a very simple way. When calling the "New recordset" functionality in a CFM page, you will notice a new button in the right side of the interface – CFC Query (see Figure 8). This button will allow you to call the CFC Query UI, importing a recordset from a CFC function in your current page.

The CFC Query will ask you to point to the Component that includes a function that returns a recordset, and it will specifically filter all the current site available packages to ease your selection (only components that include eligible functions will be listed). This interface is pretty straight forward, as it points you to a specific function from a CFC, also previewing the SQL query to help you choose. Parameters can be also sent to the component, to filter a query and return a specific record, but we'll not cover them in the current article as using them is pretty straightforward.

After adding the CFC Query on a CFM page, the generated code is a call to the getArtists function in the selected Component (see Figure 9).

The beauty of the native Dreamweaver integration (that also forced us to improve its internals) is that, after calling one CFC Query, you can use it as a normal recordset. You can see its columns in the Recordset bindings. You can also use this reusable CFQUERY in generic Server Behaviors, for example in a dynamic list, and the generated page will look like any other ColdFusion generated page in Dreamweaver – with the exception of the SQL query that will not be included – but that will be reused from the Component.

With this significant improvement, Dreamweaver is really a solid step forward for ColdFusion developers. Not only can newcomers understand ColdFusion components better, but even advanced users are able to mix Components with Dreamweaver HTML and CSS editing in a usable way.

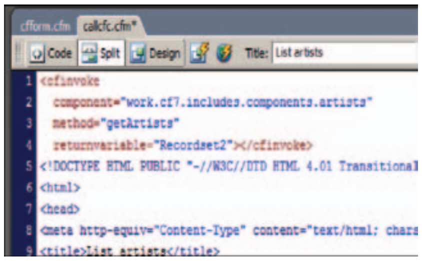
Features for Advanced Coders

Macromedia hasn't forgotten programmers when building the Dreammacromedia

There are workarounds for a mediocre intranet, but they still waste time and money.

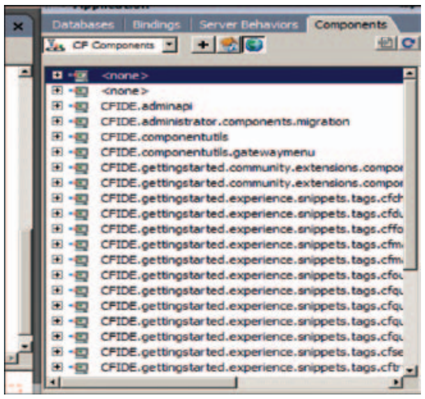
Meet the Macromedia Web Publishing System. Unlike the average CMS, this works. By allowing business users to publish content, IT bottlenecks are eliminated. Users can simply point and click to update specific pages or sections–while other areas remain protected. So everyone in your organization can share critical information quickly and easily, and you have an intranet that actually lives up to its potential.

Learn how to make your intranet work for you, not the other way around:
macromedia.com/go/webupdate



```
<cfinvoke
  component="work.cf7.includes.components.artists"
  method="getArtists"
  returnvariable="Recordset2"></cfinvoke>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; chars
<title>List artists</title>
```

Figure 9: A call to the getArtists function



```
<none>
<none>
CFIDE.adminapi
CFIDE.administrator.components.migration
CFIDE.componentutils
CFIDE.componentutils.gatewaymenu
CFIDE.gettingstarted.community.extensions.compor
CFIDE.gettingstarted.community.extensions.compor
CFIDE.gettingstarted.experience.snippets.tags.cfch
CFIDE.gettingstarted.experience.snippets.tags.cfdu
CFIDE.gettingstarted.experience.snippets.tags.cffo
CFIDE.gettingstarted.experience.snippets.tags.cfm
CFIDE.gettingstarted.experience.snippets.tags.cfm
CFIDE.gettingstarted.experience.snippets.tags.cfou
CFIDE.gettingstarted.experience.snippets.tags.cfqu
CFIDE.gettingstarted.experience.snippets.tags.cfqu
CFIDE.gettingstarted.experience.snippets.tags.cfqu
CFIDE.gettingstarted.experience.snippets.tags.cfqu
CFIDE.gettingstarted.experience.snippets.tags.cfse
CFIDE.gettingstarted.experience.snippets.tags.cftr
```

Figure 10: Components list

weaver improvements. Together with the features presented above – I'm sure some of you will have found the CFC Query interesting – a nice suite of coder-oriented features are here.

Code Completion and Code Coloring

The first and most visible improvement is the updated tag library that makes coding much more pleasurable. If you've ever tried to use Dreamweaver to code, and if you use Components, you were probably frustrated by the way components are listed. You see all the components on the server, and this could be VERY frustrating on a server with 20 sites and 100 components.

In the ColdFusion MX 7 extensions, you have the possibility to filter the components to your local site, getting a much cleaner and easier to use components list (see Figure 10).

Faster Debugging

At last, there's one more feature to ease your coding work – the improved debugging. As you might know, it's pretty simple in Dreamweaver to do a request on the server and retrieve debugging information. The ColdFusion extensions improve this feature, simplifying debugging, even on a live server where your IP is not included in the IP list for debugging.

In order to do this, the new command will:

- Add your IP to debug list (using a RDS call)
- Request the current page to retrieve the debug information
- Remove your IP from the list

Missing Features

Naturally, some feature requests didn't make it in this version. Let's walk through the most-requested features:

CVS/SVN integration: There's no support yet for source control ***in Dreamweaver*** – implementing it was not yet a priority for Macromedia.

Step debugger: ColdFusion programmers can't debug their pages line by line and, considering that PHP programmers can do this, it's pretty frustrating. However, the complexity of implementing this is very great and it requires lots of changes in the ColdFusion server and in Dreamweaver to make it work.

Framework support: There are two major frameworks for ColdFusion development – Mach II and Fusebox – and Dreamweaver doesn't support neither of them (yet, but you can check Steve Nelson's blog [see Resources, below, for URL] – he had one Dreamweaver extension for Fusebox editing published).

Better code editing: Of course, Dreamweaver focused much more on visual HTML/CSS editing than on code writing. But the hope is that this area will be improved.

Conclusion

We believe that you will like the new improvements in Dreamweaver MX 2004 for ColdFusion MX 7 development. One thing is clear – your needs are important to Macromedia, and they did everything in their power to make Dreamweaver a better ColdFusion editor.

If you think that you might have some ideas about what's missing in Dreamweaver, don't hesitate to ask them on the Dreamweaver user groups: http://www.macromedia.com/cfusion/webforums/forum/index.cfm?forumid=12 .

I know for sure that the Dreamweaver people are monitoring those groups, and maybe you will see your pet feature present in a future version of Dreamweaver.

Resources

If this article made you reconsider Dreamweaver as your editor for ColdFusion – you might also consider various other sites with free and commercial Dreamweaver extensions for productive web development.

- InterAKT: http://www.interaktonline.com/
- *Cartweaver:* http://www.cartweaver.com/
- *CommunityMX :* http://www.communitymx.com/
- *Webassist:* http://www.webassist.com/
- *Steve Nelson's blog:* http://steve.secretagents.com/

“In CFMX 7, Macromedia heard developer complaints and tried to catch up by adding features to Dreamweaver specifically for ColdFusion programmers”

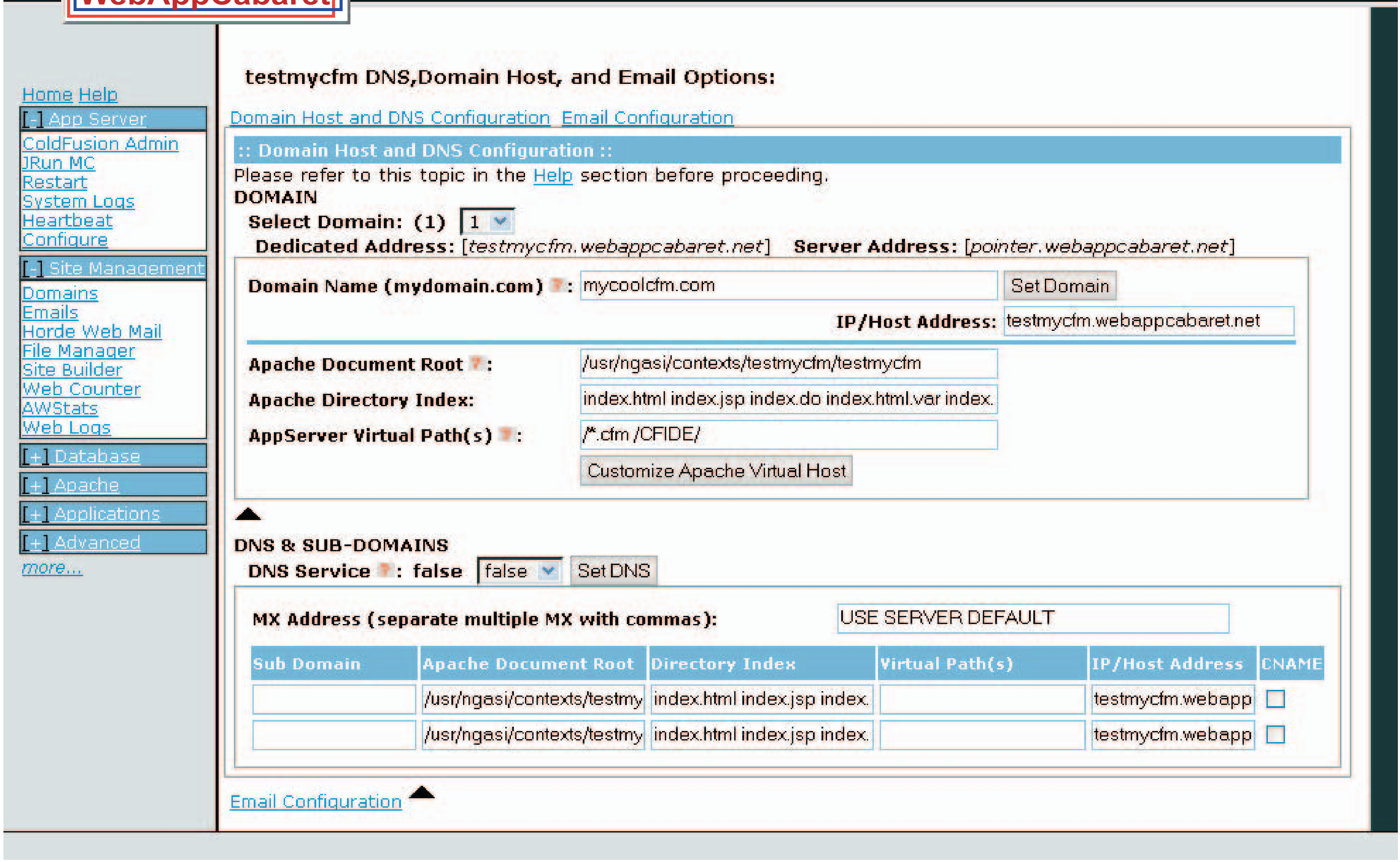
About the Author

Alexandru Costin is president of products division at InterAKT Online. As one of the InterAKT product architects, he has led the ColdFusion MX7 Extensions project and he is also involved in maintaining the InterAKT suite of Dreamweaver productivity tools. He co-authored several books and he posts regularly on Dreamweaver related newsgroups and boards.

acostin@interaktonline.com

WebAppCabaret sm

ColdFusion MX 6.1 Enterprise Hosting made easy.



testmycfm DNS,Domain Host, and Email Options:

Domain Host and DNS Configuration Email Configuration

:: Domain Host and DNS Configuration ::

Please refer to this topic in the Help section before proceeding.

DOMAIN

Select Domain: (1) 1

Dedicated Address: [*testmycfm.webappcabaret.net*] Server Address: [*pointer.webappcabaret.net*]

Domain Name (mydomain.com): mycoolcfm.com Set Domain

IP/Host Address: testmycfm.webappcabaret.net

Apache Document Root: /usr/ngasi/contexts/testmycfm/testmycfm

Apache Directory Index: index.html index.jsp index.do index.html.var index.

AppServer Virtual Path(s): /*.cfm /CFIDE/

Customize Apache Virtual Host

DNS & SUB-DOMAINS

DNS Service: false false Set DNS

MX Address (separate multiple MX with commas): USE SERVER DEFAULT

Sub Domain	Apache Document Root	Directory Index	Virtual Path(s)	IP/Host Address	CNAME
	/usr/ngasi/contexts/testmy	index.html index.jsp index.		testmycfm.webapp	☐
	/usr/ngasi/contexts/testmy	index.html index.jsp index.		testmycfm.webapp	☐

Email Configuration

Imagine a hosting company dedicated to meet the requirements for complex web sites and applications such as those developed with ColdFusion MX. At WebAppCabaret our standards based process and tools make deploying ColdFusion MX applications as easy as a point-and-click.
We call it **Point-and-Deploy Hosting**.
Our advanced NGASI Web Hosting management Control was designed for the hosting and management of ColdFusion web sites and applications thus cutting down on maintenance time.

Backed by an experienced staff as well as a **Tier 1 Data center** and network. Our network is certified with frequent security audits by reputable security firms.

All ColdFusion hosting plans have separate and individual installation AND instances of ColdFusion MX 6.1 Enterprise with **full access to ColdFusion Administrator** and JRun Management Console; so there is virtually no restriction or customization required for your application.

Complete power and control at the tip of your fingers. We take care of the system and hosting infrastructure so you can concentrate on development and deployment of your application. That is **real ROI**.

Log on now at http://www.webappcabaret.com/cdj.jsp or call today at 1.866.256.7973

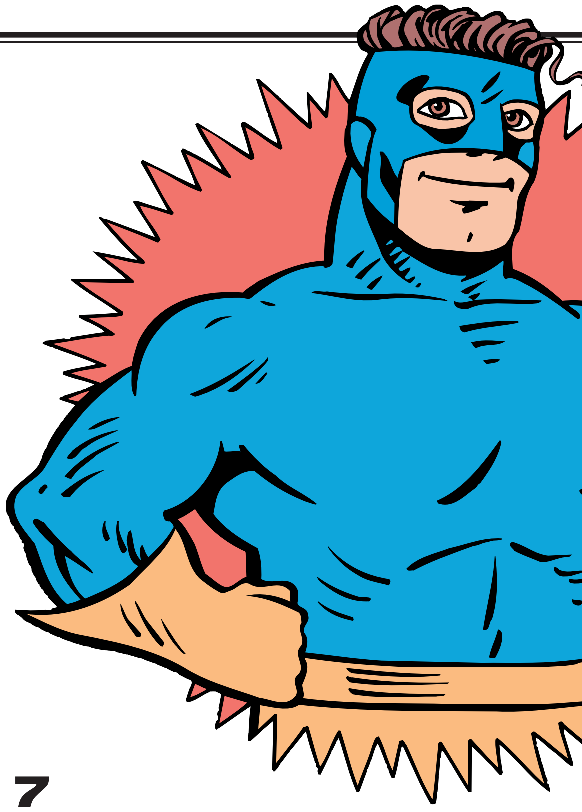
NGASI POWERED

WebAppCabaret sm

http://www.webappcabaret.com/cdj.jsp **1.866.256.7973**

Point-and-Deploy J2EE Hosting

The Release That Will Make You a Hero Again: CFMX 7



Introducing ColdFusion MX 7

By Tim Buntel

I get excited every time we release a new version of ColdFusion. I usually hit the road and talk about the release to customers at their companies, user group events, and conferences.

I start the conversation by saying, "This is the best ColdFusion ever! Wait till you see what you can do with this!" Well, ColdFusion MX 7 is here and this time, I won't need to say a word. All you need to do is take a look at this release to see that I am not exaggerating whatsoever. More than just being the best release of ColdFusion ever, this is the release that will make you a hero again - just like the first ColdFusion did when we began this incredible journey ten years ago.

Initial feature research began over two years ago and now, after countless hours of customer interviews, scores of development hours numbering in the thousands, 10 months of alpha and beta testing with more than 3000 testers, and over 15,000 regression tests, we are incredibly proud to introduce ColdFusion MX 7, the easiest way to build and deploy powerful Internet applications.

ColdFusion MX 7 will dramatically improve the experience your users have with your web applications, save you hours of time and complexity in creating those applications, and allow the applications to run in the most easy-to-manage, scalable, and reliable environment ever. In addition, this latest release will allow you to take your ColdFusion skills into an incredible new world of opportunity with development for mobile devices, instant messaging clients, and more. Let's take a look at each of these areas.

A Better Experience for Both Users and Developers

There are some chronic problems with the experience of web applications today, but ColdFusion MX 7 delivers an incredible collection of key features that can help you overcome them.

Printing web content is poor at best using today's browsers. We've all experienced trying to print from the web - content gets chopped off in the margins, page breaks are in the wrong spots, you have no control over headers or footers, and so forth.

The new CFDOCUMENT tag feature in ColdFusion MX 7 will change all of that by allowing you to dynamically transform web content into high-quality printable, portable documents in PDF or FlashPaper 2 formats using a single tag. Your users will be able to print web pages as they see them in the browser, or your can write documents to disk for offline viewing or to send as attachments to email messages. Best of all, it works with all of your content – there's no need to re-write pages or use carefully formatted XHTML.

Most web applications also suffer from the poor capabilities of HTML to display structured, formatted data. Classic dynamic database-driven applications presenting repeating-groups of data are today displayed in simple HTML tables and scrolled into long, poorly presented pages. This poor display impacts the productivity of users, as they attempt to consume this information in simple HTML rows and columns. And, once again, the ability to print, save, or e-mail this information is almost non-existent.

In order to solve this problem and deliver data in high-quality reports, many developers have had to resort to third-party product add-ons outside the web application development environment. But often this approach is costly and doesn't integrate well with the rest of the web application.

With this release, ColdFusion delivers integrated business reporting capabilities, providing unprecedented access to important business data. This will enable high-quality, structured reports to fully integrate into your web applications, providing users with

well-formatted data that is easy to understand, print, and email. Also included, the ColdFusion Report Builder - a free tool that you'll use to design your reports. It lets you include sub-reports, charts, and graphs, as well as automatically generate subtotals and totals, and much more.

ColdFusion reports integrate seamlessly into your web applications. You can use ColdFusion queries to create exactly the data needed for a dynamic report or pull data from anything else that ColdFusion can access including UDFs, CFCs, web services, XML, databases, and so on. The reports are then delivered in PDF, Macromedia FlashPaper 2, or Microsoft Excel format. You can also generate and save reports to disk for reuse and portability, and easily add the ability to e-mail reports from within applications or let users view the reports offline.

Finally, when it comes to capturing information through the browser, users are constrained by the use of web forms. They have to wait for page refreshes to get simple field-level validation, and often lose data between multi-step forms. For many ColdFusion developers, creating these HTML forms is one of the most time consuming and difficult parts of your job. You spend more time trying to make form inputs and labels line up nicely in HTML table cells than you do in writing the form's logic. ColdFusion MX 7 delivers a better experience for both you and your users by allowing you to build accessible, high-quality forms in minutes using the new, rich Flash and XML forms capabilities. Using familiar CFML tags, you can easily create complex, multi-step forms with tabbed or accordion interfaces. Your applications can use Flash controls that aren't available in regular HTML such as data grids, tree controls, and calendar date pickers. And best of all, your forms will look good and be intuitive for your users without the need to code any of the presentation-tier information.

These are just a few of the features in ColdFusion MX 7 that will make your life easier in developing web applications. I think you'll find plenty of other gems as well, including application events, a full administrator API, form masking and parameter validation, a powerful new version of CFCHART, improved web services and XML support, and incredible new Verity search capabilities.

Better Management, Scalability, and Availability

All of the great applications you build with ColdFusion MX 7 will only be successful if you deploy them into a scalable, reliable, and secure environment without requiring tremendous complexity or incurring greater administrative costs. So we've made key enhancements to the way that applications are managed and deployed too.

As we looked closely at how customers were deploying ColdFusion, we realized that it was much too complex to create multiple instances. So we created a simple, web-based user interface that anyone can use to generate new instances of applications and application servers on individual physical servers. This Enterprise Manager will provide companies the ability to easily add higher levels of availability to their applications through application isolation and clustering.

EAR and WAR files are a single file used to package an application for easy deployment, and they're the standard for deploying applications in J2EE environments. New in ColdFusion MX 7 is support for packaging and deploying applications in this standard J2EE manner expected by system administrators. With an EAR or WAR file, you can easily package your entire application into a single file for deployment onto any J2EE certified app server. The archive contains all of the application's logic, resources, and the ColdFusion runtime required to run it. From the J2EE perspective, it's just another Java application.

Finally, while ColdFusion has compiled CFML templates to Java byte code since version 6, the human-readable source code was still required on the server. ColdFusion MX 7 allows you to deploy your applications as Java byte code only - you can now choose to leave out the unencrypted CFML source files. We call this Sourceless Deployment, and it allows you to protect your source code investments and allay security concerns when sensitive information may be present in an application's logic.

A New Class of Applications - Introducing Event Gateways

ColdFusion applications today respond to HTTP requests coming most often from a web browser or web service call. Event Gateways in ColdFusion MX 7 allow your ColdFusion applications to respond to or initiate events from a nearly limitless range of other protocols: SMS text messages on mobile applications, instant messaging clients, TCP/IP sockets, file systems, and more. And if the technology that you would like to integrate with ColdFusion doesn't have an Event Gateway already, you can create your own. With the same, easy-to-use CFML used to create web applications today, you can now build gateway applications that can interact with virtually any device or protocol common in today's networked world.

One particularly powerful and exciting gateway included in ColdFusion MX 7 is for creating SMS applications. SMS, or Short Message Service, is a way to send short text messages to and from wireless devices such as mobile phones, PDAs, and pagers. And with projections that the world will have almost two *billion* mobile telephone users by 2006, there are a lot of potential users of your applications! Event gateways will allow you to leverage the productivity of ColdFusion to bring applications to this vast new market with unparalleled ease.

What's to Come in This Series

These brief explanations of course can't begin to explain all the power of this release. So we're going to have this regular "Macromedia Speaks Out" column here in CFDJ that will go more in-depth into these and many other feature areas over the next few months. Best of all, these feature articles are going to be written by the development and QA engineers here on the ColdFusion team who actually built them. It will give you an opportunity to learn firsthand why this time I can really say, "This is the best ColdFusion ever!"

About the Author

Tim Buntel is senior product manager for ColdFusion at Macromedia.

tbuntel@macromedia.com

Introducing... ColdFusion MX 7

After an entire year spent meeting with and speaking to thousands of ColdFusion developers, the CF team at Macromedia are unleashing this month the feature-rich new release, CFMX 7

By Ben Forta

It's been three years since we released Macromedia ColdFusion MX, the most dramatic and ambitious ColdFusion update ever. ColdFusion MX marked an important milestone in the ColdFusion story. It was a chance for the team to take a big step back and rebuild ColdFusion from the ground up, taking into account everything we had learned about web applications and how they are built.

ColdFusion MX was primarily an architectural release. It featured things like the following:

- A brand-new, Java-based engine
- A true compiler
- Deployment on top of industry standard J2EE servers
- Better support for XML, SOAP, and other standards
- Access to the world of Java

Of course, ColdFusion MX (and ColdFusion MX 6.1) also boasted important new features, language enhancements, improved performance, as well as greater scalability and reliability. But at its core, ColdFusion MX was all about architecture, an incredible investment in the inner workings of ColdFusion so as to facilitate a world of new functionality. ColdFusion MX has been an incredibly successful product, and a large portion of the ColdFusion user base is already taking advantage of all it has to offer. And so with ColdFusion's new engine proving its mettle and developers busily exploring the opportunities it presents, the ColdFusion team was able to spend time building new features and functionality that were not possible in the past.

For over a year we met with and spoke to thousands of ColdFusion developers. We presented ideas and previews to hundreds of user groups worldwide, brainstormed with countless partners and customers, waded through mountains of wish-list feedback, and chatted with numerous users (both current and potential). When the dust settled, a series of goals emerged:

- Make new users far more successful. ColdFusion has always appealed to new developers. There is no other language or product as well suited to their needs as ColdFusion. New users (primarily those with a background in building web pages and static sites) are an important part of the ColdFusion user base, and ColdFusion must remain dedicated to making successful development easier for these users. This involves the creation of Dreamweaver extensions and configuration screens, providing better out-of-the-box education, delivering more usable value, and more.
- Provide existing users with feature and functionality that they can use immediately. Developers are never shy about what they want. We need to deliver the features and functionality they ask for.
- Help developers (our users) make their users happier. Consumers of ColdFusion applications have common requests – things they'd like to see in the applications created for them. Many of these requests revolve around how the application captures and presents data. You said that ColdFusion must provide powerful new capabilities for forms, reporting, and printing.
- Improve reliability and deployment options. ColdFusion's Java internals opened up all sorts of powerful and important deployment and reliability options. Now ColdFusion needs to make this more available to you and your applications, more than ever before.
- Innovate, innovate, innovate. ColdFusion pioneered rapid development on the web. Indeed, there still is no quicker way to build web-based applications. The ColdFusion develop-

ment experience needs to be applied to new platforms and technologies, making them just as readily usable.

These are significant, even lofty, goals. We invested tens of thousands of development hours, launched the largest beta test program to date, and maintained regular contact with our customers so that they could keep us focused and on track.

The result is the most customer-driven ColdFusion version ever, a feature-rich release that solves real problems for real developers building real applications, a product that meets and exceeds the enumerated goals.

And so I'd like to take this opportunity to formally introduce you to ColdFusion MX 7.

Improved Form Field Validation

Data-entry forms have long been the Achilles heel of web-based applications. Without bad-mouthing HTML forms (actually, there is little need to – you've all experienced the pain firsthand), ColdFusion MX 7 improves forms in several ways, starting with improved form field validation.

For starters, ColdFusion contains additional validation types, including the oft-requested validations for e-mail and URLs. In addition, the JavaScript error message that appears when using client-side validation displays all of the validation errors at once, not just the first validation error.

Perhaps more importantly, it is now simpler to perform both client-side and server-side validation at once. The cfinput tag has a new attribute named validateAt that accepts three values:

- onSubmit (the default value) specifies client-side validation when the client submits the form, just like client-side validation in the current cfinput tag.
- onBlur specifies client-side validation as soon as a field loses focus (user tabs to the next field or clicks another field, for example).
- onServer specifies server-side validation, the same type of validation that hidden form fields perform. However, it does not require you actually to define those fields (the fields are still present, but are generated and embedded automatically).

Look at the following code:

```
<!--- Client-side validation on submit --->
<cfinput type="text"
        name="quantity"
        validate="integer"
        validateAt="onSubmit"
        required="yes"
        message="Numeric quantity is required!">

<!--- Client-side validation on loss of focus --->
<cfinput type="text"
        name="quantity"
        validate="integer"
        validateAt="onBlur"
        required="yes"
        message="Numeric quantity is required!">

<!--- Server-side validation --->
<cfinput type="text"
        name="quantity"
        validate="integer"
        validateAt="onServer"
        required="yes"
        message="Numeric quantity is required!">
```

All three cfinput tags perform the same validation but at different points within the form submission process. The best part is that validation methods may be mixed. So to validate on the client and server you could do the following:

```
<!--- Client-side validation on submit --->
<cfinput type="text"
        name="quantity"
        validate="integer"
        validateAt="onSubmit,onServer"
        required="yes"
        message="Numeric quantity is required!">
```

Here the validateAt attribute specifies two values (onSubmit and onServer) so that ColdFusion generates client-side validation code and embeds hidden form fields for server-side validation.

Another validation enhancement is input masking. ColdFusion MX 7 introduces a new attribute to solve this problem: mask takes an input mask and uses it as a data input filter. A mask is a string comprised of special characters which are used to validate data entry: A question mark (?) allows any character, the letter A allows only alphabetical characters, the number 9 allows digits, and an X allows alphanumeric characters. Any other character is a literal and is itself embedded in the input.

For example, to validate a three-digit age, you could do the following:

```
<cfinput type="text"
        name="age"
        maxlength="3"
        mask="999">
```

The mask filter "999" would only accept digits. If a user entered anything other than a digit, the tag would simply ignore that input. Similarly, to validate a US-style phone number in the format (123) 456-7890, you could use the following code:

```
<cfinput type="text"
        name="phone"
        maxlength="13"
        mask="(999) 999-9999">
```

Again, the mask attribute value allows only digits, but inserts the other characters automatically.

For a Canadian postal code you could use the following:

```
<cfinput type="text"
        name="postcode"
        maxlength="7"
        mask="A9A 9A9">
```

You get the idea. Input masking does not negate the need for input validation, but it does make for a far better user experience.

Flash Forms

Another important enhancement to forms is less of an enhancement and more of a drop in replacement. Macromedia Flash has long been a potential replacement for HTML forms, enabling developers to leverage the capable Flash Player to deliver a better user experience. Of course, that has meant learning Flash or deploying Macromedia Flex. For ColdFusion developers who simply want better forms, there needs to be a simpler solution.

To make the creation of Flash-based forms easier for coders, ColdFusion MX 7 introduces a series of tags that make building powerful and sophisticated data-entry forms as simple as, well, ColdFusion. For example, if you need to prompt a user for a date (perhaps a date of birth), you can replace the HTML code:

```
<form action="" ...>
<input type="text" name="dob">
...
</form>
with the following:
<cfform format="flash" action="" ...>
<cfinput type="datefield" name="dob">
...
</cfform>
```

This creates a form with a text field, just like the HTML text field, except that this one displays a pop-up date chooser when a user selects the field. It's that simple. Using a combination of cfinput tags to create controls, and cfformgroup tags to group them as needed, ColdFusion developers can generate Flash forms without knowing (or even owning) Flash.

A pop-up calendar is just the start of it. Other features include the following:

- Tree control
- Data grid
- Multipane forms (using tab or an accordion-style interface)
- Input masking
- Data binding between form controls
- Client-side events
- Integrated error checking and validation feedback

How does this all work? When ColdFusion processes a page containing these tags, it generates the Flash ActionScript needed to create the form and then compiles that code into a SWF file and embeds it in the page. All of that is hidden from ColdFusion developers who simply use CFML tags, just as it is already.

In other words, you can create forms that leverage the power of Flash while retaining the productivity and simplicity that has become the hallmark of ColdFusion. Stay posted to the ColdFusion Developer Center, as it will feature an article from the ColdFusion engineers who worked on Flash forms.

Printable Web Pages

Considering that most ColdFusion development involves applications that search and display data, it should come as no surprise that printing (or the inability to do so easily) has long been a source of aggravation for ColdFusion developers. After all, if you were to generate a web page and then select File > Print in your browser, well, you'd never really know what to expect the printed output to look like.

Developers have had to resort to all sorts of tricks to control printed output generation, with varying degrees of success – until now.

ColdFusion MX 7 introduces a new tag (a family of tags, actually) that makes turning web pages into printable content as painless as you'd expect from CFML. Look at the following code snippet:

```
<cfdocument format="pdf">
Here is some text.<br>
<img src="image.gif">
</cfdocument>
```

The cfdocument tag takes whatever code you provide and generates printable documents in Adobe PDF and Macromedia FlashPaper formats. In this example, the PDF would contain a single line of text followed by an image on the next line. That's all it takes.

The cfdocument tag is designed to work with any web pages. There is no need for XHTML or specific formatting. You can use inline formatting or CSS, you can embed images and links, you can use tables and <p> tags for alignment – it'll just work.

In addition, cfdocument tag supports the following, among other things:

- Page orientation of portrait or landscape
- Different (and custom) page sizes
- Headers and footers
- Different formatting options (page size, headers, and footers) for different sections of a document
- Saving generated output to disk

If you have web pages that you'd like to print – just about any pages – one simple set of tags will solve the problem for you quickly and efficiently.

Reporting

In addition to free-form page printing, ColdFusion developers often have another printing-related need: structured reporting. Third-party reporting tools have long been difficult to integrate with ColdFusion applications. ColdFusion MX 7 introduces its own reporting solution, a Report Writer and a reporting engine.

ColdFusion MX 7 introduces a new file type, the CFR (ColdFusion Report) file. CFR files are report templates you create with the new ColdFusion Report Writer, which looks a lot like other report-building tools you may have used. Once you have created a report, you can embed it in your applications using a cfreport tag, like this:

```
<cfquery datasource="mydsn" name="myQuery">
SELECT * FROM myTable
</cfquery>
```

```
<cfreport format="FlashPaper"
          template="myReport.cfr"
          query="myQuery">
```

As you can see, queries are passed to the cfreport tag at runtime. This means that a CFR file is actually more of a report template than an actual report, but you can use it to construct any SQL – dynamically if required. You have complete flexibility and control over how your ColdFusion application creates reports.

The ColdFusion reporting features the following:

- Reports that can be generated in PDF, FlashPaper, and Excel formats
- Intuitive report creation tool featuring report bands, data aggregation functions, embedding charting, and more
- Toolbars used to insert images, text, lines labels, as well as to manage data alignment
- Integrated SQL query builder
- Support for subreports (reports embedded within other reports)
- Report creation wizard
- Charting wizard
- Full zoom and preview support

ColdFusion reporting is available in all editions of ColdFusion, and ColdFusion Enterprise provides greater control so as to manage report creation in high-usage environments. Read more about reporting in the other CFMX 7 article in this issue Collin Tobin & Dean Harmon's article, "Building Reports in ColdFusion MX 7."

Event Gateways

ColdFusion has long been an incredibly simple way to build web applications. That's why we all use ColdFusion – for its simplicity (and therefore productivity). But ColdFusion is not exclusively tied to the web. In fact, ColdFusion does not even talk to web browsers; that is the web server's job. ColdFusion simply executes scripts on the server in response to requests – requests which (thus far) have been originated typically through HTTP.

Could ColdFusion respond to other requests – for example, data sent to a specific port, or changes in a folder, or inbound SMS and IM messages, or database table changes, or...? The answer is yes: ColdFusion can respond to any and all of those; it just needs a way to know when those events occur. It needs gateways.

Gateways are interfaces to other systems, ways for events to trigger ColdFusion processing. A gateway watching a folder on a server can trigger ColdFusion execution when folder contents change. A gateway connecting to an SMS provider can respond to inbound SMS messages (and send SMS messages as well). A database trigger can ping a gateway so that a database event forces ColdFusion processing (imagine being able to generate static HTML pages automatically and dynamically whenever back-end databases change).

Among the gateways included with ColdFusion MX 7 are the following:

- Asynchronous processing gateway
- Folder watcher gateway
- Socket gateway

fast. easy. affordable.

CommonSpot™ Content Server

Efficient Content Management

100 % browser-based
Content object architecture
Template driven pages
50+ standard elements
Extensible via ColdFusion
Content reuse
Content scheduling
Flexible workflow
Granular security
CSS support
508 compliance
Personalization
Replication
Custom metadata
Custom authentication
Static site generation
Multilanguage support

With CommonSpot Content Server you get it all. CommonSpot's exceptional blend of rapid deployment, ease of use, customization and scalability make it the leading ColdFusion content management solution.

Our rich Web publishing framework empowers developers with out-of-the-box features such as template-driven pages, custom content objects, granular security, flexible workflow and powerful ColdFusion integration hooks (just to name a few), allowing you to rapidly build and efficiently deploy dynamic, high performance Web sites.

For the past six years, PaperThin has been a leader in the ColdFusion community, and CommonSpot has been the solution of choice for organizations of all sizes, including Architect of the Capitol, AFL-CIO, Boeing, Kaiser Family Foundation, Ohio University, PGA.com and hundreds of others. CommonSpot's sophisticated feature set and affordable pricing are an unbeatable combination.

Call us today at 800.940.3087 to schedule a live demonstration of your site running under CommonSpot, or visit www.paperthin.com to learn more.

Paper | Thin

- JMS gateway
- Lotus Sametime gateway
- XMPP protocol gateway
- SMS gateway

Third-party vendors are already hard at work creating ColdFusion MX 7 gateways. You could write your own gateways, too. The result is that ColdFusion is now able to interact with just about any back-end and technology available to you. You will be able to read more about event gateways and what you can do with them in Jim Schley and Tom Jordhal's article next issue, "Writing and Using Event Gateways in ColdFusion 7."

Improved Deployment

ColdFusion (as of ColdFusion MX) is a Sun-verified Java application, and is installed on top of J2EE servers like any other Java applications. Well, kind of. You can deploy ColdFusion MX (including ColdFusion MX 6.1) on top of a J2EE server but the complete process is one that does not excite J2EE administrators.

In J2EE-land, administrators are typically given an application to deploy, and they don't pay a whole lot of attention to what that application is and how it works. Nor should they. Developers should worry about applications and J2EE administrators should worry about servers staying up and running well.

How does this work? Applications to be deployed on a J2EE server are packaged as a single file, a Java archive file (usually with an EAR or WAR extension). The archive file contains everything needed for an application to run: source code, configuration settings, supporting files – everything. Once an application has been tested and is ready for deployment, it is packaged—the package itself being test-deployed – and handed off to the J2EE administrator, who drops it onto the J2EE server. (I am simplifying things a bit, but the basic flow is accurate.) What J2EE administrators don't do, or don't like doing, is running through a post-installation to-do list containing things like creating a data source, setting up some mappings, installing these extensions, and so on.

Yet J2EE administrators deploying ColdFusion MX must do just that. You can deploy ColdFusion itself – core engine, compiler, and runtime services – like any other Java application, but that is just ColdFusion itself. Once you deploy ColdFusion, someone still needs to move all the CFML and CFC files over and use the ColdFusion Administrator to define data sources and mappings and more. In other words, while ColdFusion itself is deployed like any other J2EE application, the total experience of deploying a ColdFusion application is not.

ColdFusion MX 7 changes this by giving you the ability to build complete J2EE deployment packages. It comes with a packaging tool that creates a complete EAR or WAR file that can contain the ColdFusion runtime (with or without specific features), application code, data sources, and more. The tool can take some time to run because building a complete, deployable EAR or WAR file is not a quick process. When it's finished, however, you can give that package to a J2EE administrator to deploy just like any other Java application. This means you can deploy ColdFusion applications on a J2EE server that is not running ColdFusion because your Java package file will contain the ColdFusion engine.

This is an important, and much-needed enhancement. >From a J2EE administrator's perspective, deploying ColdFusion MX 7 applications will be just like deploying any Java applications. Actually, J2EE administrators don't even have to know that it is a ColdFusion application. To them, it's Java pure and simple.

Multiple Instances Made Easier

ColdFusion MX Enterprise supports multiple instances of ColdFusion – although that is less a ColdFusion feature and more a J2EE-server feature that ColdFusion users can take advantage of. After all, ColdFusion MX is a Java application. I've discussed the benefits and importance of using multiple instances before but, simply put, using multiple ColdFusion instances provides greater security, stability, and scalability. It's almost like having ColdFusion installed on multiple physical servers, except it's all on one server.

If you have an existing J2EE server, you can create multiple EAR or WAR files and then deploy them as you would any other Java application. If you do not have an existing J2EE server, the ColdFusion installer can install JRun 4 for you. In doing so, it creates and deploys the first ColdFusion instance so that you can be up and running immediately. But when you want to deploy additional instances, things get a little tricky for users without experience in J2EE server administration. You need to use the J2EE server administration tools to create a new server, run the ColdFusion installer to create the EAR or WAR, expand the files (if using JRun), make tweaks to an XML file, and then copy the expanded files into the server folders. This is all feasible, but not exactly a trivial process. Unfortunately, this is why so many users have yet to deploy multiple instances.

ColdFusion MX 7 make this process much simpler. It has the same three installation options as ColdFusion MX 6.1, but selecting the JRun + ColdFusion option in ColdFusion MX 7 installs additional administration screens that make the deployment of instances (and even the creation of clusters of instances) as simple as any other ColdFusion administration processes. You'll be able simply to fill in a form and click a button to create a new instance – without needing to use the JRun management tools or the ColdFusion installer, without needing any XML tweaks, and without even knowing what an EAR or WAR file is.

How could you use this new functionality? Consider these use cases:

- You deploy a brand-new application, one that uses its own data sources and is built by a different development team (that needs ColdFusion Administrator access). You want the new application to be safely isolated from your existing production applications. Simply create a new instance, launch the ColdFusion Administrator for that new instance, define the data sources and any other needed settings, copy the code, and you are good to go.
- You are about to deploy an update to your application code and you need to maintain the existing application as a fallback, just in case something goes wrong. Simply create a new instance (you could even create a CAR file using the old instance to save data sources and any other needed configuration, launch the ColdFusion Administrator for the new instance, and import the CAR file to import those settings), copy the code, associate your web server to the new instance, stop the old instance (to

prevent resources from being used unnecessarily), and you are done. If you then need to roll back, start the old instance and associate your web server to it again. Clean and simple.

- You have an existing application that spikes in load (holiday shoppers maybe), and you want an additional server running the same application so that you can handle a greater load, and also provide failover in the event that a server problem occurs. Simply create a new instance, point to the Java package containing the code and settings used for the first instance, and let ColdFusion do its thing. You'll have a second instance created, configured like the first, and containing the same application as the first. You can then use a second screen to create a cluster – perhaps to enable session sharing between the instances.

You Get the Idea.

Of course, for those who want more control, the installer will still install JRun with its own management software just as it does now – and you can deploy and manage applications just as you can now. But for those of you who simply want to leverage what is undoubtedly the most significant benefit of ColdFusion Enterprise over ColdFusion Standard, ColdFusion MX 7 makes life much simpler. Stay posted to the ColdFusion Developer Center; it will feature an article from the ColdFusion engineers who worked on the Enterprise Manager.

Lots of Other Goodies

This is just the start of it. Other ColdFusion MX 7 goodies include the following:

- XForms support
- Administration API Improved web services support
- Dreamweaver extensions
- Simplified authentication against NT domains and Active Directory

ColdFusion MX is the most eagerly anticipated version of ColdFusion in a long time. This is ColdFusion designed by developers, for developers; ColdFusion created in response to vast user feedback; and thus ColdFusion that solves real problems right out of the box.

If you are using an older version of ColdFusion, now is the time to upgrade. If you have yet to experience ColdFusion development, there has never been a better or more exciting time to start than now. Check out the upgrade options for ColdFusion MX 7.

About the Author

Ben Forta is the Macromedia senior product evangelist and the author of numerous books, including ColdFusion Web Application Construction Kit and its sequel Advanced ColdFusion Application Development, as well as books on SQL, JavaServer Pages, WAP, Windows development, and more.

ben@forta.com

Develop Powerful Web Applications

with ColdFusion MX 7

- Easily build and deploy multi-step data entry forms with CFML tags
- Solve business reporting problems with Integrated Reporting
- Manage your account with HostMySite's Control Panel
- Call 24x7x365 for expert ColdFusion support

Visit hostmysite.com/cfdj for a special offer

HostMySite.com

1-877-215-4678

– continued from page 8

Advanced Deployment with Ant (and DbUnit)

2. *index.cfm:* User form, user can choose the directory where checksum files shall be generated.
3. *directorylist.cfm:* Lists the entries in a directory tree in a format compatible with CFDirectory using recursion.
4. *ant.cfm:* We receive the directory entered by the user in index.cfm. Then, after calling the customtag directorylist, we loop through the directory list entries. For every list entry a java file object is created:

```
<!--- set file to check --->
<cfobject type="JAVA" action="Create" name="oFile" class="java.io.File">
<cfset oFile.init(qFileList.path)>
```

Then we create a Ant Checksum object

```
<!--- checksum task --->
<cfobject type="JAVA" action="Create" name="oAntChecksum" class="org.
apache.tools.ant.taskdefs.Checksum">
```

and evaluate the checksum using the method eval()

```
<!--- evaluate checksum --->
<cfset oAntChecksum.setFile(oFile)>
<cfset sReturnval = oAntChecksum.eval()>
```

If the return value sReturnval is true, the file was unchanged, otherwise the file was changed. This simple application could be easily extended and used to check if files can be overwritten from a hotfix. This should only be the case when they were not changed by customers.

Ant Productivity Tips

Use Properties

By using properties you will make your build files reusable with minimal effort. In time, you can develop a collection of targets that you can reuse to develop new build files quickly. Use external property files to keep per-user settings out of the build files – especially passwords.

Use Relative File Paths

Relative paths are a good way to ensure your project's portability. Absolute file paths are much less flexible and will make changes in the directory structure of your project more painful than is necessary.

CFDJ Advertiser Index

ADVERTISER	URL	PHONE	PAGE
ACTIVEPDF	WWW.ACTIVEPDF.COM	866-468-6733	29
BLOG-N-PLAY	WWW.BLOG-N-PLAY.COM	888-303-5282	46
CFDYNAMICS	WWW.CFDYNAMICS.COM	866-233-9626	4
CFUNITED	WWW.CFUNITED.COM	301-424-3903	COVER III
COLDFUSION DEVELOPER'S JOURNAL	WWW.SYS-CON.COM/COLDFUSION	888-303-5282	35
EV1 SERVERS	WWW.EV1SERVERS.NET	800-504-SURF	3
HAL HELMS, INC	WWW.HALHELMS.COM		33
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM/CFDJ	877-215-4678	23
INTERAKT ONLINE	HTTP://KTML.INTERAKTONLINE.COM/	4031 401.68.19	6
INTERMEDIA.NET	WWW.INTERMEDIA.NET	800-379-7729	COVER IV
MACROMEDIA	WWW.MACROMEDIA.COM/GO/VIDEO5	415.252.2000	COVER II
MACROMEDIA	WWW.MACROMEDIA.COM/GO/WEBUPDATE	415.252.2000	13
PAPERTHIN	WWW.PAPERTHIN.COM	800-940-3087	21
SAVVY SOFTWARE	WWW.BESAVVY.COM	866-870-6358	25
SEAPINE SOFTWARE	WWW.SEAPINE.COM/WHITEPAPER.PHP	888-683-6456	9
SYS-CON REPRINTS	WWW.SYS-CON.COM	201-802-3026	43
WEBAPP CABARET	WWW.WEBAPPCABARET.COM/CDJ.JSP	866-256-7973	15

Issuing SQL Commands

Ant provides the SQL task for executing simple SQL statements as well as script files containing multiple SQL statements. The SQL task can be used for running SQL statements against any data source for which you have a Java database connectivity API (JDBC)-compliant driver

Get

The <get> task can fetch any URL, so it can be used to trigger remote server-side code during the build process, from remote server restarts to sending SMS/pager messages to the developer cellphones.

i18n

Internationalization is always trouble. Ant helps here with the native2ascii task which can escape out all non-ASCII characters into unicode. You can use this to write files which include strings (and indeed comments) in your own non-ASCII language and then use native2ascii to convert to ASCII.

Conclusion

Our team successfully introduced Ant and DbUnit to a client ColdFusion development environment. Since then, we've written several build files and automated processes that in the past took too much time and were error prone. Adding a bit of automation to the deployment process is a start to putting you in control of your development process. You should have less to worry about, a shorter build/test/deploy cycle, and more time to spend on new features.

Resources

- *Ant Homepage:* http://ant.apache.org
- *Ant Download:* www.artfiles.org/apache.org/ant/binaries/apache-ant-1.6.2-bin.zip
- *DbUnit Homepage:* http://dbunit.sourceforge.net
- *DbUnit Download:* http://prdownloads.sourceforge.net/dbunit/dbunit-2.1.zip?download
- *Ant article by John Ashenfelter (CFDJ, February 2004):* www.sys-con.com/story/?storyid=43787&DE=1
- *Effective Unit Testing with DbUnit:* www.onjava.com/pub/a/onjva/2004/01/21/dbunit.html
- *Canoo – Free tool for automated testing of Web applications:* http://webtest.canoo.com
- *Anthill – Free automated build tool:* www.urbancode.com/projects/anthill/default.jsp

About the Author

Harry Klein is cofounder and CTO at CONTENS Software GmbH, a leading supplier of enterprise content management software. He is a Certified Advanced ColdFusion developer and Microsoft MSCE.

klein@contens.de

Savvy™ CONTENT MANAGER

Powerful
Web Content Manager
Faster and Easier

Call for DEMO
1.866.870.6358

www.BeSavvy.com
Savvy Software Inc.

macromedia® ALLIANCE PARTNER

CF Six Pack

New developers should be using these tags

By Greg Cerveny

This article is for you if you are a beginning coder. You know how to do marvelous things like query databases and output the results. You can take data submitted from a form, manipulate it, and perform calculations. After mastering these fundamentals, you are ready to expand your knowledge of ColdFusion. This article explains six ColdFusion tags that you can easily start using today.

These tags are beyond the basics, are quick to learn, and provide utility. The tags listed here have a brief explanation of their use, a detailed look at the attributes, and an example of use.

The most important tag in this article is cfqueryparam. Most of you are not using cfqueryparam - I know, because I have worked on your code! If you take one thing away from this article, please take that tag. But in order to understand cfqueryparam, we should first take a look at cfparam. So let's start there.

cfparam

```
<cfparam
  name = "param_name"
  type = "data_type"
  default = "value">
```

At the most basic level of cfparam, it determines whether a variable exists. It can also verify that the variable is of the correct type. If neither of these conditions is met the tag will throw an error, or optionally set the variable to a default value. As it sounds, it can be used in a few different ways. These uses correspond with the attributes, so let's start by taking a look at them.

The first attribute is name. This is the name of the variable whose existence you want to determine. It's the only required attribute. If this is the only attribute used, it operates in this way: it checks if a variable of that name exists, and if it doesn't, it throws an error:

```
<cfparam name="myVariable">
```

```
Essentially, it is the equivalent of this code

<cfif NOT isDefined("myVariable")>
   <cfthrow />
</cfif>
```

The second attribute is type. This defines what data type the variable is supposed to be. There are a few options: any, array, binary, boolean, date, guid, numeric, query, string, struct, UUID, variableName. Most of these values are self explanatory, for a more detailed description of these types please see the ColdFusion documentation. If the type attribute is not specified, the default is any. Meaning, the variable has to be a variable of any kind, just as long as it as a variable:

```
<cfparam name="myBoolean" type="boolean">
```

Essentially it is the equivalent of this code

```
<cfif NOT isDefined("myBoolean") AND NOT isBoolean("myBoolean")>
   <cfthrow />
</cfif>
```

The last attribute is default. This attribute defines what value the variable holds, if it does not meet the previous conditions. Looking at our previous code representations, it changes the cfthrow, to a cfset. Instead of throwing an error, it assigns the variable to a default value. In other words, if the variable does not exist or is not of the right type, then the variable is created and set to this default value:

```
<cfparam name="myBoolean" type="boolean" default="false"/>
```

Essentially is the equivalent of this code

```
<cfif NOT isDefined("myBoolean") AND NOT isBoolean("myBoolean")>
   <cfset myBoolean = false />
</cfif>
```

It is good coding practice to cfparam your variables at the beginning of a document. By doing this you are checking for the existence of variables, and if they don't exist you create them. This avoids several isDefined() functions, or worse, missing variable errors. It also lets the coder know what variables will be coming into the page - that is, if you aren't documenting your code.

You have probably realized that simply checking for a variable's existence doesn't guarantee that there is valuable information stored in it. By using cfparam you are ensuring the variable exists and is the right type. Now you are left with only the task of performing operations to determine the quality of the information. Which is a lot simpler.

Example of use:

You have a page that pulls a list of users from a database. You pass the order in the url (ascending or descending). At the top of the processing page you put:

```
<cfparam name="url.order" type="string"
default="ASC">
```

This code looks for order in the url, if it doesn't exist or isn't a string, it creates it with the value of "ASC".

cfqueryparam

```
<cfqueryparam value = "parameter value"
   CFSQLType = "parameter type"
   maxLength = "maximum parameter length"
   scale = "number of decimal places"
   null = "Yes" or "No"
   list = "Yes" or "No"
   separator = "separator character">
```

Cfqueryparam is a lot like cfparam in that it validates the incoming value, but it's more robust and also offers binding. We'll tackle these one at a time. As the name implies, it's part of a query. Think of it as this: as cfparam is to cfm files, cfqueryparam is to query. There is one essential difference however: cfqueryparam has no default value.

So you might ask yourself, why do you need cfqueryparam when the following code works:

```
<cfquery datasource="yourDatasource">
INSERT INTO mailinglist (name, emailAddress)
VALUES( '#form.name#', '#form.emailAddress#')
</cfquery>
```

There are a few reasons this is bad. Communication with the database is inefficient. None of the form values have been validated to contain correct data. None of the string data has been escaped. Strings passed in can contain things like single quotes and semicolons that can cause innocent values to error upon insertion, and malicious values to delete or change information.

Now that you've seen how cfqueryparam can help you, let's take a look at the attributes.

The first attribute is value. This is the value that you want to pass into your query. If you are still using CFMX 6.0, there is an issue with using a function within this attribute, but that has been fixed in later patches.

The second attribute is CFSQLType. You have a lot of options, they are listed in Table 1. The CFSQLType is the sql type that the paramater value will ultimately bind to (see sidebar). In other words, this is the column type that was setup in your database.

CF_SQL_BIGINT
CF_SQL_BIT
CF_SQL_CHAR
CF_SQL_BLOB
CF_SQL_CLOB
CF_SQL_DATE
CF_SQL_DECIMAL
CF_SQL_DOUBLE
CF_SQL_FLOAT
CF_SQL_IDSTAMP
CF_SQL_INTEGER
CF_SQL_LONGVARCHAR
CF_SQL_MONEY
CF_SQL_MONEY4
CF_SQL_NUMERIC
CF_SQL_REAL
CF_SQL_REFCURSOR
CF_SQL_SMALLINT
CF_SQL_TIME
CF_SQL_TIMESTAMP
CF_SQL_TINYINT
CF_SQL_VARCHAR

Table 1

The max length attribute protects you from attempting to insert too large of a variable into your database. If you have a varchar(15), but you attempt to insert a string of length 25, your database throws an error. If you put a value into the max length attribute, cfqueryparam will then validate the information and throw an error before sending it the database. The errors thrown by the cfqueryparam are often easier to interpret than those thrown by the database.

The scale attribute measures the number of permitted digits after the decimal point. This only applies to CFSQLTypes CF_SQL_DECIMAL and CF_SQL_NUMERIC. This is similar to max length in that it validates the value and throws an error if it does not pass.

The null attribute determines whether the value to be inserted is NULL, meaning nothing is to be inserted. When you specify null to equal yes, then the value attribute is ignored.

The final two attributes deal with delimited lists. The list attribute identifies the passed in attribute value as a delimited list. The separator attribute specifies which delimits the list, it defaults to ','. Cfqueryparam will automatically add the necessary parenthesis and quotes to the value.

Example of use:

You have a form that has name and e-mail address, and you would like to insert it into a database. On your action page you have the code:

```
<cfparam name="form.name" type="string">
<cfparam name="form.emailAddress"
type="string">

<cfquery datasource="yourDatasource">
INSERT INTO mailinglist (name, emailAddress)
VALUES(
<cfqueryparam value="#form.name#"
maxlength="50" cfsqltype="cf_sql_varchar">,
<cfqueryparam value="form.emailAddress#"
maxlength="50" cfsqltype="cf_sql_varchar">
    )
</cfquery>
```

cfsettings

```
<cfsetting
  enableCFoutputOnly = "Yes" or "No"
  showDebugOutput = "Yes" or "No"
  requestTimeOut = "value in seconds" >
```

Cfsetting is a page level control of general settings. There are only a few attributes, and each performs a unique operation. As we examine the attributes, the use of the associated settings will become clear.

The first attribute is enableCFoutputOnly. This attribute blocks any output unless it is explicitly stated within cfouput or writeOuput. In other words, if it's not in between <cfoutput> </cfoutput>, or within writeOutput(), it won't be displayed to the user. This attribute is used to eliminate the white space and generally untidy output produced by ColdFusion (or any tag based language for that matter).

The second attribute is showDebugOutput. This manually turns on and off debugging output for the current page. This is really used for turning debugging off. You have to have debugging turned on in the cf administration in order for you to be able to use showDebugOutput="yes". More effective is the cfsetting showdebugoutput="no". This is good if the debugging information is affecting your layout.

The last attribute is requestTimeOut. The first two attributes are Yes / No values, but this attribute takes the time out in seconds. This is a self-imposed time limit in which the page will time out after the specified limit. This overrides any time out settings made in the cf administrator.

Example of use:

This tag is easily adapted to be controlled via the URL. Using the following code you can change settings for your debugging on the fly via the URL.

```
<cfparam name="url.debug" type="boolean"
value="true">
<cfparam name="url.timeout" type="numeric"
value="20">
<cfparam name="url.cfoutputonly" type="boolean"
value="false">
<cfsetting showdebugoutput="#url.debug#"
enablecfoutputonly="#url.cfoutputonly#"
requestTimeOut="#url.timeout#">
```

cfsavecontent

```
<cfsavecontent variable = "variable name">
  content
</cfsavecontent>
```

Cfsavecontent is an easy tag that is really useful. It saves the content rendered in between the pair of tags to the variable specified in the attribute. This tag is used for variable assignment, like cfset. Instead of <cfset varaible = "value">, the value is placed between the cfsavecontent tags instead of quotation marks.

This tag has a lot of applications. It's good for setting variables containing large amounts of text. It improves readability for long expressions. By surrounding the value with tags, you eliminate issues of escaping quotation marks. Another good application of this is putting javascript in here.

If your web applications make use of including templates, you have noticed that sometimes certain elements require a javascript function. Using cfsavecontent, it's easy to keep this javascript code organized. The code is outputted at the top of your page, instead of placing the javascript in the middle of your page whenever you need it.

Example of use:

Here we have a variable that will contain a hyperlink. The hyperlink calls a javascript function to open a browser window. We can't easily use cfset because the link contains single and double quotes. Using cfsavecontent we do not have that problem

```
<cfsavecontent variable="popUpLink">
<a href="javascript://" onclick="MM_
openBrWindow('http://www.test.com','test','widt
h=200,height=75')">test</a>
</cfsavecontentcfsilent
```

<cfsilent>

```
...
</cfsilent>
```

Cfsilent is an easy tag with no attributes, so let's get right into its functionality. All output is suppressed between a pair of cfsilent tags. Even if you have <cfoutput> or writeOutput(), it won't display anything.

So why do we want to suppress white space? It's all about the final display. Cfsilent helps combat rendered pages that have a lot of extra space. This is because ColdFusion is a tag-based language, and it renders output not necessarily in the CFML language. So even though you don't have any code, you still have line returns that are output.

Example of use:

Here we are performing some operations and variable assignment. Nothing is being displayed, so we will place it between cfsilent to tidy up the final output.

```
<cfsilent>
   <cfparam name="url.action" default="">
   <cfif len(url.action)>
         …
   <cfelse>
         …
   </cfif>
   <cfloop>
         …
   </cfloop
</cfsilent>
```

cfdump

```
<cfdump
  var = "#variable#"
```

General Overview of Binding

The database management system receives SQL in plain text and has to interpret how to retrieve the data. Suppose your database gets a lot of requests like this:

SELECT email, password from users where ID = 18;
SELECT email, password from users where ID = 45;
SELECT email, password from users where ID = 7;

The database first looks up to see if it has processed this request before. If it has, it just uses that method of finding the data. If it hasn't, then it has to interpret the SQL, which takes longer. With the example above, each select statement is unique. If you are using bind variables, instead of ID = 12, it would have ID = @variable. That way the SQL statement is the same and it doesn't have to do as much processing.

```
expand = "Yes or No"
label = "text">
```

Cfdump outputs the value of the variable in a nicely formatted version. The difference between <cfoutput>#variable#</cfoutput> and cfdump is that you can output complex variables like queries or even objects. A bit of warning first: don't use this is as part of your final application. Cfdump is used strictly as part of your debugging process to expose variables.

Let's dig into the parameters a bit more.

The first attribute is var. This takes a ColdFusion variable surrounded by pound signs. The pound signs are important. If you call this tag with var="myvariable" instead of var="#myvariable#", it's going to output the string "myvariable", and not the value contained within the variable of the same name.

The second attribute is expand. This takes either a Yes or No. When cfdump outputs complex variables such as structs, XML, or queries it generates collapsible DHTML. The individual branches and sub variables can be expanded and contracted to allow you to drill down the data. When set to yes, the variable will be fully expanded. When set to no, it will be contracted.

This is useful when dumping more than one large complex variable type. For example if you had 4 queries with up to 20 rows in each, if you dumped them all expanded that would take up a large amount of space. So you would want to dump them contracted and just click on the queries as you want to explore them.

The last parameter is label. This takes a string that titles the output.

Example of use:

This is a form action page that processes and inserts the passed information into the database. After looking at the database you have determined that there is an issue with this page. Your first step is look at the data that is being sent in, so you cfdump the form scope and abort processing so you can take a closer look at the data:

Top of page:

```
<cfdump var="#form#"><cfabort>
```

Conclusion

You now have six new tags in your ColdFusion toolbelt. Start implementing them into your existing code and new projects. You will find that after using each a few times, they come quite easily and you won't remember how you ever lived without them.

About the Author

Greg Cerveny is an applications developer living in Colorado.

geg.cerveny@gmail.com

What's your PDF?

Precise Document Formatting

With activePDF Server , you gain full control over your PDF output with conversion options that allow you to specify page size, compression and resolution options, embed text, create bookmarks, concatenate to existing files, and more. Licensed per server, you can easily add PDF generation to virtually any Windows application.

Populate Dynamic Forms

With activePDF Toolkit 's form-filling capabilities, you can dynamically populate PDF forms with data and images from a database, allow users using only Adobe Reader to fill-in and save forms and use PDF forms as document templates to precisely control image placement and resizing. With Toolkit's robust API, the automation of virtually any PDF manipulation task becomes possible - append, stamp, stitch, merge, paint, secure PDF and more.

Promote Digital Fidelity

Do you need to standardize PDF output within your enterprise? With DocConverter, you can easily use built-in support for "watched" folders to implement server-side PDF generation in a matter of minutes, with full control over the PDF output at the server level. Or, use DocConverter's programmable COM object to integrate convert-to-PDF functionality within your enterprise application.

Present Data Fashionably

Ensuring precise layout of an HTML document can be a nightmare, especially when printing. PDF guarantees pixel-perfect layout every time as what you see is what you print. With activePDF WebGrabber, you can dynamically convert any URL, HTML stream, or HTML file to PDF on the fly, while maintaining embedded styles.

Download your free trial version today at www.activePDF.com

activePDF® Leading the iPaper Revolution

Deploying Applications with ColdFusion MX 7

Introducing sourceless and EAR/WAR file deployment

By Dave Carabetta

The release of ColdFusion MX 7 provides developers with several options for deploying their CFML applications. While the current option of using ColdFusion Archives (CAR files) has worked to this point, a fundamental problem still remains that the source code needed to be included.

However, the two new options available in this release, sourceless and EAR/WAR file deployment, truly give developers the flexibility to deploy their code while optionally preserving their valuable intellectual property. (It is important to note that New Atlanta's BlueDragon Server line has supported sourceless deployment of CFML source code since the introduction of their 6.1 product line in June 2004. All references to sourceless and EAR/WAR deployments as "new features" apply solely to Macromedia ColdFusion MX 7.) Coupled with the streamlined ability to add new ColdFusion instances to an environment via the ColdFusion MX Administrator (Enterprise edition only), much of the manual work associated with application deployment has been automated, allowing the developer to focus on development instead.

Pre-compiled and Sourceless Deployment

Before ColdFusion MX 7, the only option for companies to protect their commercial CFML code distribution was the cfencode utility. However, anybody who has worked with ColdFusion for any reasonable amount of time knows that a simple Internet search will turn up a utility for decompiling the templates, which essentially rendered cfencode pointless. So for many years, all a developer could realistically do was release their code to the public and simply hope that the end user would abide by the license agreement. Further, many companies, not just those selling commercial applications, simply weren't comfortable running production applications without some sort of source code protection.

ColdFusion MX 7 ships with a new cfcompile utility that has two main uses. First, it can be used to simply compile your ColdFusion source code into the resulting Java class files so that the initial load time is quicker. Second, it allows for pre-compiling CFML code (CFM, CFC, and CFR templates) to Java byte code, which can then be deployed without the source templates. This utility, which can be found in the *cfroot*/bin (for the server configuration) and in the *cf_webapp_root*/WEB-INF/cfusion/bin (for the multi-server and J2EE configuration) directory, can be run on one template (i.e., a custom tag) or an entire directory structure containing a complete CFML application (i.e, a blogging application).

In both cases, its usage is incredibly simple. To pre-compile CFM, CFC, and CFR templates (note that this is *not* for sourceless deployment, which will be covered shortly), the general usage is

```
cfcompile webroot [directory-to-compile]
```

where *webroot* is the absolute path to the site's web root and the *directory-to-compile* is an optional parameter specifying the directory to compile (Note: all sub-directories are automatically compiled as well). If no directory is specified, the utility will compile all CFM, CFC, and CFR templates from the web root, recursively. This usage of cfcompile is the same process that the ColdFusion runtime uses when first loading a page: it parses the file, compiles it, and stores the appropriate class files in the cfclasses directory under your ColdFusion installation. For a small set of files, this might not seem very useful due to the massive compiler optimizations that the ColdFusion MX runtime has received since its first release. However, for large sites, this initial overhead savings could prove valuable.

As an example, the Mach-II framework files are used as the directory structure (the freely available code is located at http://www.mach-ii.com/code.cfm). Assuming that the web root is C:\Inetpub\wwwroot, the following command-line statement would pre-compile all the templates in the framework:

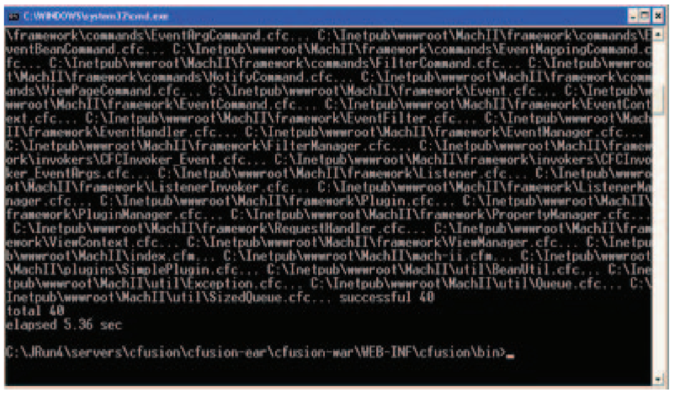
```
cfcompile C:/Inetpub/wwwroot C:/Inetpub/wwwroot/MachII
```

As seen in Figure 1, the forty templates comprising the base framework were compiled in 5.36 seconds. Figure 2 shows the resulting class files in Windows Explorer. A count of the number of resulting class files output shows that 345 files were generated for a total directory size of just over 1 MB (remember, the number of compiled class files is not one-to-one with the number of ColdFusion templates. ColdFusion Components, for example, have their methods broken out into individual class files). Running this utility has just saved the ColdFusion runtime from having to generate over 300 files when they are first loaded!

The second use of this utility lies in its ability to pre-compile templates for sourceless deployment. What this means is that applications can now be distributed without the underlying source code and without the fear that a vendor's intellectual property will be compromised. The general usage is

```
cfcompile -deploy webroot directory-to-compile output-directory
```

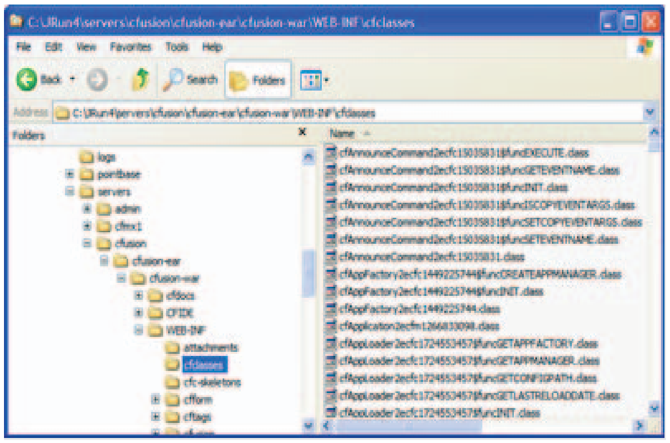
Once again using the Mach-II core file distribution, the sourceless deployment syntax is



```
\framework\commands\EventArgCommand.cfc... C:\Inetpub\wwwroot\MachII\framework\commands\E
ventBeanCommand.cfc... C:\Inetpub\wwwroot\MachII\framework\commands\EventMappingCommand.c
fc... C:\Inetpub\wwwroot\MachII\framework\commands\FilterCommand.cfc... C:\Inetpub\wwwroo
t\MachII\framework\commands\NotifyCommand.cfc... C:\Inetpub\wwwroot\MachII\framework\comm
ands\ViewPageCommand.cfc... C:\Inetpub\wwwroot\MachII\framework\Event.cfc... C:\Inetpub\w
wwroot\MachII\framework\EventCommand.cfc... C:\Inetpub\wwwroot\MachII\framework\EventCont
ext.cfc... C:\Inetpub\wwwroot\MachII\framework\EventFilter.cfc... C:\Inetpub\wwwroot\Mach
II\framework\EventHandler.cfc... C:\Inetpub\wwwroot\MachII\framework\EventManager.cfc...
C:\Inetpub\wwwroot\MachII\framework\FilterManager.cfc... C:\Inetpub\wwwroot\MachII\framew
ork\invokers\CFCInvoker_Event.cfc... C:\Inetpub\wwwroot\MachII\framework\invokers\CFCInvo
ker_EventArgs.cfc... C:\Inetpub\wwwroot\MachII\framework\Listener.cfc... C:\Inetpub\wwwro
ot\MachII\framework\ListenerInvoker.cfc... C:\Inetpub\wwwroot\MachII\framework\ListenerMa
nager.cfc... C:\Inetpub\wwwroot\MachII\framework\Plugin.cfc... C:\Inetpub\wwwroot\MachII\
framework\PluginManager.cfc... C:\Inetpub\wwwroot\MachII\framework\PropertyManager.cfc...
 C:\Inetpub\wwwroot\MachII\framework\RequestHandler.cfc... C:\Inetpub\wwwroot\MachII\fram
ework\ViewContext.cfc... C:\Inetpub\wwwroot\MachII\framework\ViewManager.cfc... C:\Inetpu
b\wwwroot\MachII\index.cfm... C:\Inetpub\wwwroot\MachII\mach-ii.cfm... C:\Inetpub\wwwroot
\MachII\plugins\SimplePlugin.cfc... C:\Inetpub\wwwroot\MachII\util\BeanUtil.cfc... C:\Ine
tpub\wwwroot\MachII\util\Exception.cfc... C:\Inetpub\wwwroot\MachII\util\Queue.cfc... C:\
Inetpub\wwwroot\MachII\util\SizedQueue.cfc... successful 40
total 40
elapsed 5.36 sec

C:\JRun4\servers\cfusion\cfusion-ear\cfusion-war\WEB-INF\cfusion\bin>_
```

Figure 1: Command-line output after pre-compiling the Mach-II framework



```
C:\JRun4\servers\cfusion\cfusion-ear\cfusion-war\WEB-INF\cfclasses
cfAnnounceCommand2ecfc15035831$funcEXECUTE.class
cfAnnounceCommand2ecfc15035831$funcGETEVENTNAME.class
cfAnnounceCommand2ecfc15035831$funcINIT.class
cfAnnounceCommand2ecfc15035831$funcISCOPYEVENTARGS.class
cfAnnounceCommand2ecfc15035831$funcSETCOPYEVENTARGS.class
cfAnnounceCommand2ecfc15035831$funcSETEVENTNAME.class
cfAnnounceCommand2ecfc15035831.class
cfAppFactory2ecfc1449225744$funcCREATEAPPMANAGER.class
cfAppFactory2ecfc1449225744$funcINIT.class
cfAppFactory2ecfc1449225744.class
cfApplication2ecfm1266833098.class
cfAppLoader2ecfc1724553457$funcGETAPPFACTORY.class
cfAppLoader2ecfc1724553457$funcGETAPPMANAGER.class
cfAppLoader2ecfc1724553457$funcGETCONFIGPATH.class
cfAppLoader2ecfc1724553457$funcGETLASTRELOADDATE.class
cfAppLoader2ecfc1724553457$funcINIT.class
```

Figure 2: Pre-compiled class file output

```
cfcompile -deploy C:/Inetpub/wwwroot C:/Inetpub/wwwroot/MachII C:/Inetpub/
wwwroot/MachII_Sourceless
```

The major usage difference is the presence of the "-deploy" switch that tells the cfcompile utility to compile the CFML source code to Java byte code. Further, the *output-directory* is required since the resulting files will still have the same name and file extension as the source files. Once this process is done, it is simply a matter of backing up the original CFML source code and copying the generated byte code templates into the original directory. That's it! You now have a copy of the Mach-II framework that can be run without the original source code!

EAR/WAR Deployment

While the cfcompile utility satisfies the long-standing enhancement request for protection of source code, it does not answer the other request that applications be easier to deploy. ColdFusion MX has been a pure J2EE application since its initial release. A traditional J2EE application is bundled as a EAR (Enterprise Application Archive) or a WAR (Web Application Archive) file. At companies that run a pure J2EE environment, system administrators simply take the EAR/WAR file and drop it into the J2EE instance. However, traditional ColdFusion deployments meant that a server first had to be installed (the ColdFusion runtime) and then the CFML source code installed separately. This did not sit well with system administrators, who felt this two-step process was awkward, and hence limited ColdFusion's adoption in those environments.

To solve this disconnect, ColdFusion MX 7 introduces the option of packaging CFML applications as a EAR or WAR file so that one file can be handed off to the system administrator for deployment. This option means that deployment and management is easier for the system administrator while subsequently allowing ColdFusion to be a viable development alternative when building J2EE applications. Found under the "Packaging & Deployment" section in the ColdFusion MX Administrator (Enterprise version only), the "J2EE Archives (.ear/.war)" option provides a variety of options such as whether or not to include the source code or whether or not to include the ColdFusion Administrator. Figure 3 shows an example of bundling the Mach-II framework into a WAR file (MachII.war).

Having created your EAR or WAR file, it is now literally as easy as sending the resulting file to your system administrator for deployment – no separate server installation is needed!

J2EE Instance Management

Running ColdFusion on multiple instances has many benefits such as application isolation, greater stability, and greater scalability (for more information on the benefits of multiple J2EE instances, see Ben Forta's July 2003 CFDJ article entitled "When One ColdFusion Is Not Enough" at http://sys-con.com/story/?storyid=42050&de=1). Before the release of ColdFusion MX 7, the creation of new J2EE instances in which ColdFusion server would be deployed (remember, ColdFusion MX is simply a standard Java application deployed into a J2EE server) left a lot of additional setup work for administrators. From a separate interface for actually creating the instance (i.e., the JRun Admin console) to the manual editing of the underlying XML configuration files (to get features such as session replication properly working), creating a new instance was too tedious of a process, and one that caused many administrators to forgo the feature.

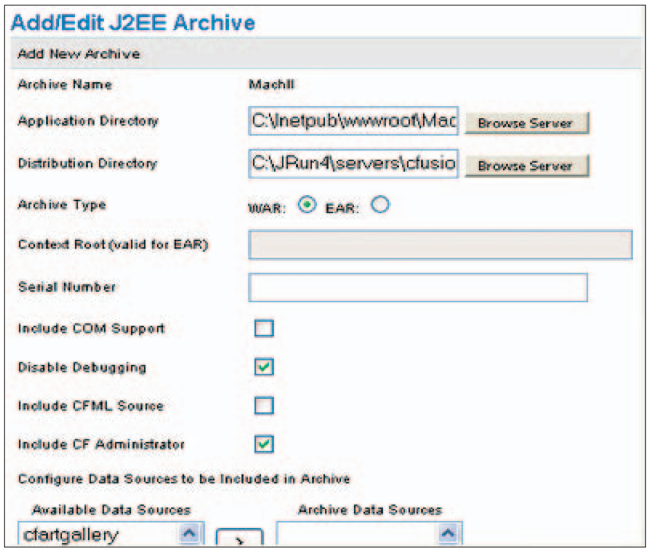
Recognizing this barrier to multiple instance deployment adoption, Macromedia has included a new feature in the ColdFusion Administrator for creating new J2EE servers with no more than a couple of mouse clicks (Note: this feature is only available if the integrated JRun + ColdFusion option was selected while running the installer. Macromedia did not backport this option for existing JRun installations.) Found in the ColdFusion MX Administrator under the "Enterprise Manager" navigation bar heading, setting up a new J2EE instance really could not be any easier. From the main screen, simply click the "Add New Instance" button. As seen in Figure 4, only a few pieces of information are needed for ColdFusion to create the new instance.

The "Server Name" field is required, as this is the name of the J2EE instance. The next two fields are optional. The "Server Directory" field (which is automatically filled in based on the "Server Name" value filled in) is where the instance will reside on the file system. The "Create From EAR/WAR" option is where the EAR/WAR packaging feature mentioned previously really shines. Remember the MachII.war file created earlier that had the ColdFusion runtime bundled with it? Well, if it needs to de deployed to a separate instance, point to the WAR file created and the Manager will deploy the ColdFusion runtime from the bundled archive. If the JRun installation is on a Windows server, there are also options to create the new instance as a Windows service and to define the service with an auto restart recovery option. Clicking the "Submit" button will start the process of installing the new instance, deploying the ColdFusion server within it, optionally deploying the application to run within it, and starting the instance. When everything is complete, the screen will look similar to Figure 5 below:

It is important to realize that the setup of a J2EE instance and the subsequent deployment of ColdFusion MX within it require a lot of files to be laid down. Therefore, it might be tempting to think that the process is not responding. Rest assured that the process is indeed running under the hood and simply take several minutes to complete.

Licensing and Other Restrictions

The pre-compile and sourceless deployment licensing and restrictions are pretty straightforward. The ability to run applications without the source code is only compatible with ColdFusion MX 7 installations – it is not backwards compatible with ColdFusion MX 6.1 and earlier. That being said, the ability to both compile and run applications without the source code


Add/Edit J2EE Archive

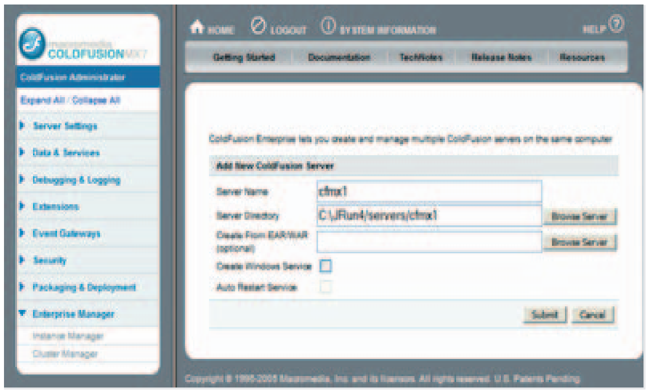
Add New Archive

Archive Name	MachII
Application Directory	C:\Inetpub\wwwroot\Mac
Distribution Directory	C:\JRun4\servers\cfusio
Archive Type	[✓ WAR:] EAR:
Context Root (valid for EAR)	
Serial Number	
Include COM Support	☐
Disable Debugging	✓
Include CFML Source	☐
Include CF Administrator	✓

Configure Data Sources to be Included in Archive

Available Data Sources	Archive Data Sources
cfartgallery	

Figure 3: Defining what to include in the EAR/WAR archive



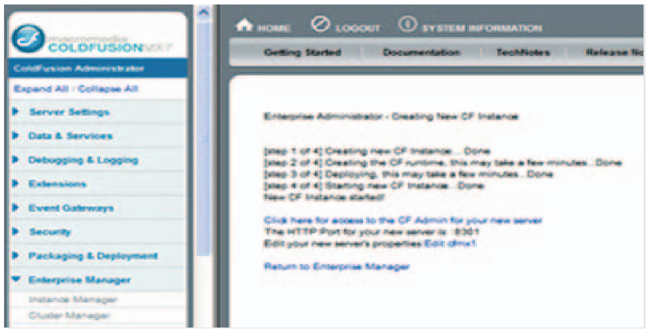
ColdFusion Enterprise lets you create and manage multiple ColdFusion servers on the same computer

Add New ColdFusion Server

Server Name	cfmx1
Server Directory	C:\JRun4/servers/cfmx1
Create From EAR/WAR (optional)	
Create Windows Service	☐
Auto Restart Service	☐

Copyright © 1995-2005 Macromedia, Inc. and its licensors. All rights reserved. U.S. Patents Pending

Figure 4: Defining new J2EE instance details



```
Enterprise Administrator - Creating New CF Instance

[step 1 of 4] Creating new CF Instance... Done
[step 2 of 4] Creating the CF runtime, this may take a few minutes...Done
[step 3 of 4] Deploying, this may take a few minutes...Done
[step 4 of 4] Starting new CF Instance...Done
New CF Instance started!

Click here for access to the CF Admin for your new server
The HTTP Port for your new server is: 8301
Edit your new server's properties Edit cfmx1

Return to Enterprise Manager
```

Figure 5: Creating a new J2EE instance

is available across all editions of the ColdFusion MX 7 product line – Developer, Standard, and Enterprise. This means that you do not have to have an Enterprise license to compile your application for Enterprise customers.

The EAR/WAR deployment option is a little trickier. First off, the ability to *create* the package is available in any edition of ColdFusion MX 7. However, because EAR/WAR files can only be deployed to J2EE servers, ColdFusion MX 7 Enterprise is the only edition that supports the deployment of these packages. (This deployment restriction is consistent with New Atlanta's BlueDragon product line, as seen at http://www.newatlanta.com/products/bluedragon/product_info/features.cfm#FCMatrix) Further, deploying an EAR or WAR package requires the purchase of ColdFusion MX 7 Enterprise licenses to match the number of CPUs on the deployment server. For example, if the package is being deployed to a 4-CPU server, then a 4-CPU Enterprise license of ColdFusion MX is required. (Note that Macromedia has changed its server licensing policy with the release of ColdFusion MX 7 such that a 2-CPU license can no longer be split across two 1-CPU machines. Each physical server now requires a separate license. Further, the licensing is based on the number of *physical* processors in the server only.) If a serial number is not applied when the EAR/WAR package is first created, then either the ColdFusion Administrator must be included with the package, or the Administrator API must be used so that the user can apply their license upon deployment (See the Macromedia LiveDocs for more information on the Administrator API). If the Administrator is not included and no serial number is applied via the Administrator API, the application is deployed as a 30-day trial that reverts to the Developer version upon expiration (which is restricted to localhost access plus 2 client machines).

While seemingly a hassle on the surface, the EAR/WAR deployment option is still much easier than previous versions' deployment experience. As an example, if a company wanted to purchase a bug tracking system with ColdFusion, that company would have to do the following using pre-MX 7:

1. Buy the bug tracking system application itself
2. Purchase ColdFusion and the appropriate number of licenses
3. Install ColdFusion server
4. Deploy the bug tracking system to the new server

With ColdFusion MX 7's EAR/WAR packaging option, that same company would have to do the following:

1. Buy the bug tracking system with the ColdFusion runtime bundled as one cost
2. Purchase and apply the appropriate number of ColdFusion licenses

Much simpler, isn't it?

The Enterprise Manager feature for creating new J2EE instances is limited to the Enterprise edition. However, because the licensing is a "per CPU" agreement, there is no limit to the number of instances that can be created. Further, the Instance Manager can only create instances on the same server from which the ColdFusion MX Administrator is installed as opposed to across a network to another installation.

Summary

After years of developer demand, Macromedia has come through with an intelligent, straightforward way to confidently deploy CFML applications. The cfcompile utility improves initial template load time and gives a developer confidence that the innovation and hard work that they have put into a product will not be compromised while the ability for ColdFusion to be packaged up into one clean installation step opens it up to an increased number of companies who previously would have shunned the idea because of the fragmented install process. The J2EE Instance Manager vastly simplifies this process because system administrators now simply have to click a few buttons as opposed to running several installers.

About the Author

Dave Carabetta has been working with ColdFusion for over 8 years now. Currently, he is the lead developer for a commercial real estate financial services company in New York City, where he develops CF applications as well as managing the web site's cluster setup. He's currently working on ColdFusion/Flex integrated applications.

cbetta@hotmail.com

ColdFusion Everywhere PART 2

Run any application on the desktop

By Phil Cruz

& Dick Applebaum

When we last left our two intrepid heroes they had just shown us how to create and run our first CFEverywhere application (including installing and configuring its requisite CFML server, application server, and Web server).

Once installed, we were able to move our "package" (application and accompanying servers) to any hard drive and run it without change (we will exploit this capability later).

In Part 1 we:

1. Discussed the reasons and advantages of deploying applications as CFEverywhere packages
2. Examined some applications that would benefit from this approach
3. Installed the Jetty J2EE server
4. Deployed the BlueDragon/J2EE CFML engine
5. Wrote and installed our first CFEverywhere application

As with most example applications, ours was not much to brag about. However, we did validate that CFML programs can be made to run in a small footprint on the desktop – a noteworthy accomplishment.

In this article, we will add a relational database server to our CFEverywhere package. This will allow us to run most any CFML application we desire on the desktop. A simple database-driven CFML application and sample data is provided so that we can validate the package and demonstrate it in action. Let's get started!

The first step is choosing the database server component. When selecting a database we had these requirements:

- Free for commercial use and distribution
- Cross-platform
- Embeddable, small footprint
- Easy to install/configure/administer
- Strong developer community

Based on these criteria we selected Derby. You may not have heard of Derby but you may have heard of Cloudscape. Cloudscape was one of the earlier Java-based database servers released in 1997. IBM acquired the technology in 2001, and in August 2004 they contributed the source code to the Apache Software Foundation as the Derby project. As an Apache project, you can feel comfortable that Derby will have strong developer support and continue to be a robust database platform.

Derby is not the only database that is suitable for CFEverywhere. You are free to choose any database that meets your requirements. If you are targeting only the Windows platform you may choose to use an Access database. Daffodil One$DB (www.daffodildb.com/one-dollar-db.html) is another Java-based database that looks suitable.

The Derby project Web site is located at http://incubator.apache.org/derby/. There's a lot of good documentation about the database on the Web site. We recommend reading it to become more familiar with the product. Follow the links on the site and download the Derby package from http://cvs.apache.org/dist/incubator/derby/10.0.2.1/incubating-derby-10.0.2.1-bin.zip. Extract the zip file into your CFEverywhere folder and rename the folder to Derby. IBM hosts the Derby JDBC drivers. Go to http://www.ibm.com/developerworks/db2/downloads/jcc/ to download the drivers. You will have to follow several links and complete a free registration. The download package is called "Released product: IBM Cloudscape (IBM DB2 JDBC Universal Driver, for Cloudscape/Derby)" and the filename is db2jcc_for_derby.zip.

After you have downloaded the file, extract the contents and copy db2jcc_license_c.jar and db2jcc.jar to cfeverywhere\server\wwwroot\WEB-INF\lib. The lib directory is on BlueDragon's classpath so you won't have to add the driver files to the classpath.

To see if Derby is working let's try to invoke the sysinfo utility. At the command line, create an environment variable pointing to the Derby directory:

On Windows

```
set DERBY_INSTALL=c:\cfeverywhere\derby
```

Note: Be careful not to put any spaces around the equal sign. Then add the Derby JARs to the classpath:

```
set CLASSPATH=%DERBY_INSTALL%\lib\derby.jar;
```

AN ADVANCE LOOK AT BLAC

COLDFUSION

ColdFusionJournal.com

Leveragin
or Just F

Editorial
Blackstone
Robert Diamond page 5

CF Community
Tales from the List
Simon Horwith page 7

cf101
What's the Best Approach to Software Development?
Jeffry Houser page 32

CFUGs
ColdFusion User Groups
page 42

Training: Fast Track to Fl
Macromedia Training 'Flexes' I

CFML: Harnessing the P
SQL Server Using Store
Use of stored procedures wi
architecture in your CFML

Foundations: A We
A perfect marriage betw

<BF> on <CF>: C
and Data Abstra
basic object function

PDFs: Creatin
Your CF Appli
only functional,

RETAILERS PLEASE DISPLAY UNTIL SEPTEMBER 30, 2004

SYS-CON MEDIA

SPECIAL ISSUE: COLDFUSION SECURITY

COLDFUSION Developer's Journal

ColdFusionJournal.com

September 2004 Volume: 6 Issue: 9

web services EDGE conference
2005 East
February 15-17, 2005
Hynes Convention Center
Boston, MA

SANDBOX SECURITY
By Alex Hubner

Editorial
Reaching Deeper
Simon Horwith page 5

Community Focus
Problems Securing Your Applications?
Simon Horwith page 7

CF101
ColdFusion's Application Framework
Jeffry Houser page 28

CFUGs
ColdFusion User Groups
page 40

FOUNDATIONS
'Selling' ColdFusion
Hal Helms page 46

Security: ColdFusion Security Best Practices Knowing the security risks are there is half the battle — Bryan Murphy 16

Security: Public-Key Encryption Making strong encryption nearly painless — Wayne Graham 20

cfcUnit: ColdFusion Unit Testing Framework Paul Kenney's version — Harry Klein 30

<bf> on <cf>: Defending ColdFusion Against... — Ben Forta 36

Real-World Uses: Using ColdFusion for Network Monitoring Find the exact piece of data you need — Ryan Emerle, Matthew I. Fusfield 42

Security: Top Ten Web Security Tips You can't be too careful — Michael Smith 48

RETAILERS PLEASE DISPLAY UNTIL NOVEMBER 30, 2004

SYS-CON MEDIA

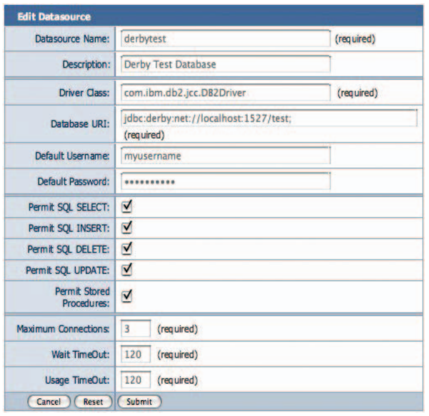
```
    %DERBY_INSTALL%\lib\derbytools.
jar;%CLASSPATH%
```

Note: That should all be entered as one line.

On Mac OS X and Linux

```
export DERBY_INSTALL=/Applications/cfevery
    where/derby

export CLASSPATH=$DERBY_INSTALL/lib/derby.jar:
$DERBY_INSTALL/lib.derbytools.jar:$CLASSPATH
```



Edit Datasource	
Datasource Name:	derbytest (required)
Description:	Derby Test Database
Driver Class:	com.ibm.db2.jcc.DB2Driver (required)
Database URI:	jdbc:derby:net://localhost:1527/test; (required)
Default Username:	myusername
Default Password:	••••••••••
Permit SQL SELECT:	✓
Permit SQL INSERT:	✓
Permit SQL DELETE:	✓
Permit SQL UPDATE:	✓
Permit Stored Procedures:	✓
Maximum Connections:	3 (required)
Wait TimeOut:	120 (required)
Usage TimeOut:	120 (required)

Figure 1: Derby datasource

Note: That should all be entered as one line.

Invoke sysinfo with:

```
java org.apache.derby.tools.sysinfo
```

If you did everything correctly, you should see information about Java and Derby. In addition to sysinfo, Derby comes with a command-line utility called "ij" for interacting with the database. We'll use ij to create an empty database. At a command line type:

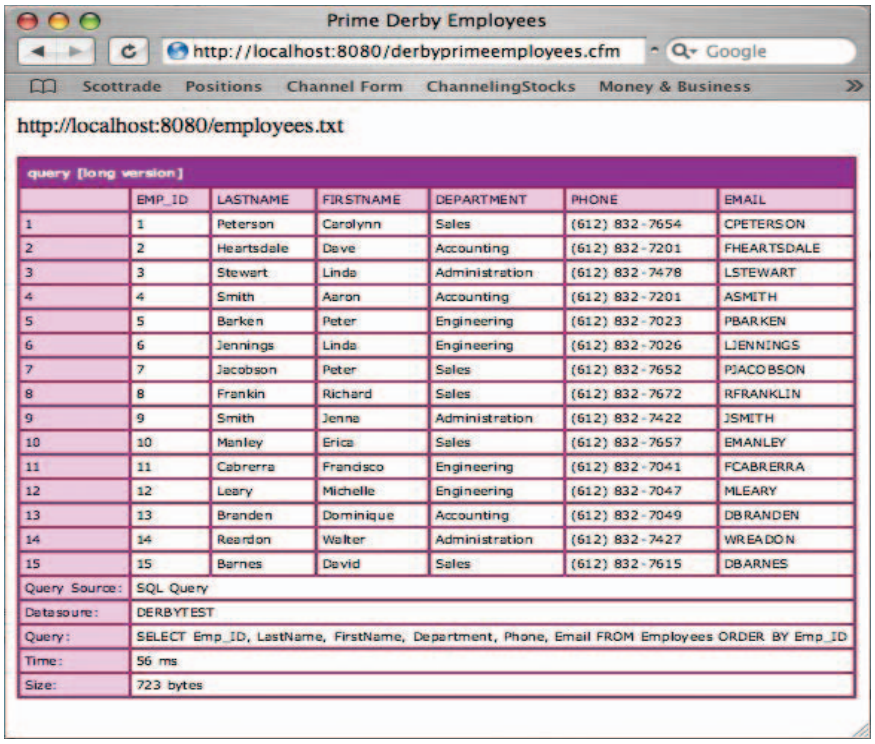
On Windows

```
    cd \cfeverywhere\derby
    java org.apache.derby.tools.ij
```

On Mac OS X and Linux

```
cd /Applications/cfeverywhere/derby
java -Dderby.storage.fileSyncTransactionLog=true\
org.apache.derby.tools.ij
```

Note: The extra system parameter for the transaction log on the command line is actually only required for OS X due to a JVM issue with Derby on that platform. Then at the ij> prompt type:



http://localhost:8080/employees.txt

query [long version]

	EMP_ID	LASTNAME	FIRSTNAME	DEPARTMENT	PHONE	EMAIL
1	1	Peterson	Carolynn	Sales	(612) 832-7654	CPETERSON
2	2	Heartsdale	Dave	Accounting	(612) 832-7201	FHEARTSDALE
3	3	Stewart	Linda	Administration	(612) 832-7478	LSTEWART
4	4	Smith	Aaron	Accounting	(612) 832-7201	ASMITH
5	5	Barken	Peter	Engineering	(612) 832-7023	PBARKEN
6	6	Jennings	Linda	Engineering	(612) 832-7026	LJENNINGS
7	7	Jacobson	Peter	Sales	(612) 832-7652	PJACOBSON
8	8	Frankin	Richard	Sales	(612) 832-7672	RFRANKLIN
9	9	Smith	Jenna	Administration	(612) 832-7422	JSMITH
10	10	Manley	Erica	Sales	(612) 832-7657	EMANLEY
11	11	Cabrerra	Francisco	Engineering	(612) 832-7041	FCABRERRA
12	12	Leary	Michelle	Engineering	(612) 832-7047	MLEARY
13	13	Branden	Dominique	Accounting	(612) 832-7049	DBRANDEN
14	14	Reardon	Walter	Administration	(612) 832-7427	WREADON
15	15	Barnes	David	Sales	(612) 832-7615	DBARNES

Query Source:	SQL Query
Datasoure:	DERBYTEST
Query:	SELECT Emp_ID, LastName, FirstName, Department, Phone, Email FROM Employees ORDER BY Emp_ID
Time:	56 ms
Size:	723 bytes

Figure 2: Prime Derby employees

```
  ij>connect 'jdbc:derby:test;create=true';
  ij>exit;
```

(Don't forget the semicolons at the end of the command line.) If you go to the Derby directory you will notice a test subdirectory was created. This directory contains all the files that represent the test database. Ij is just a command-line utility but it doesn't actually start Derby in network mode. To do that we need to call the NetworkServerControl class; we'll create a batch file (or a shell script) to simply the process.

On Windows

Create a file startdb.bat that looks like this:

```
@echo off
chdir>~.tmp
FOR /F "eol=; tokens=1-17 delims=#" %%a in
(~.tmp) do set dirnow=%%a
del ~.tmp
SET DERBY_INSTALL=%dirnow%\derby
SET DERBY_SYSTEM_HOME=%dirnow%\derby

FOR %%X in ("%DERBY_INSTALL%") DO SET
DERBY_INSTALL=%%~sX

set CLASSPATH=%DERBY_INSTALL%\lib\derby.
jar;%DERBY_INSTALL%\lib\derbytools.jar;%DERBY_
INSTALL%\lib\derbynet.jar;%CLASSPATH%

cd derby
java org.apache.derby.drda.NetworkServerControl
start
```

Note: Each command is on a separate line.

On Mac OS X and Linux

Create a file startdb.sh that looks like this:

```
dirnow=`pwd`

export DERBY_INSTALL=$dirnow/derby

export CLASSPATH=$DERBY_INSTALL/lib/derby
  jar:$CLASSPATH

export CLASSPATH=$DERBY_INSTALL/lib/derbytools
  jar:$CLASSPATH

export CLASSPATH=$DERBY_INSTALL/lib/derbynet
  jar:$CLASSPATH
```

```
cd derby

java org.apache.derby.drda.NetworkServerControl
  start
```

Note: Each command is on a separate line.

Make sure the file has proper permissions. Execute the bat file (or run the shell script) and you should see:

```
   Server is ready to accept connections on
port 1527.
```

Derby is up and running!

In Part 1 of this series we ran a simple "Hello CFEverywhere" CF application to demonstrate that CF was installed and working. Now that we have the Derby database system installed, we need a simple application that will utilize this capability and demonstrate a database-enabled CFEverywhere environment. Download the following three files from the URL at the end of this article and copy them to your wwwroot directory:

- derbyprimeemployees.cfm
- employees.txt
- derbyemployeesdrilldown.cfm

Next, we must define a datasource. To get to the BlueDragon Administrator we need to start Jetty. Go to a command line, change to the cfeverywhere\server directory and issue the start command, i.e.,

```
   cd \cfeverywhere\server
java -jar start.jar etc/cfeverywhere.xml
```

Point your browser to the BlueDragon administrator at http://localhost:8080/bluedragon/admin.cfm.

Click the datasources link and set up a datasource with the following parameters (see Figure 1):

```
Datasource Name:  DerbyTest
         Driver:  com.ibm.db2.jcc.DB2Driver
            URL:  jdbc:derby:net://local-
host:1527/test;
      User Name:  myusername
       Password:  mypassword
```

After you create the datasource make sure to verify the connection. With our datasource defined, we can use CF to create and populate the database tables (see Figure 2).

Point your browser to: http://localhost:8080/derbyprimeemployees.cfm.

This template:

1. Drops any existing Employees Table
2 Creates a new Employees Table
3. Reads the employees.txt CSV file
4. Iterates over the CSV file inserting records into the database
5. Reads all the records from the database and dumps them into the browser

The program is set up so you can run it at any time to refresh the Employees Table, regardless of its current status.

Next, point your browser to (see Figure 3): http://localhost:8080/derbyemployeesdrilldown.cfm.

This is a simple program that allows you to page-through the records in the database.

Ta-dah! We've met the objective of this article – we have a database-driven CFEverywhere application running locally, giving the appearance of a traditional desktop application.

Experiment! Add your own applications and databases. As before, move the cfeverywhere directory around – even to a different platform. It's also worth noting that although the sample applications we've demonstrated have been very simple, it is only due to the scope of this article. CFEverywhere applications can utilize all CFML tags/functions including ColdFusion components and Web services. They can also use popular application frameworks such as Mach-II and Fusebox.

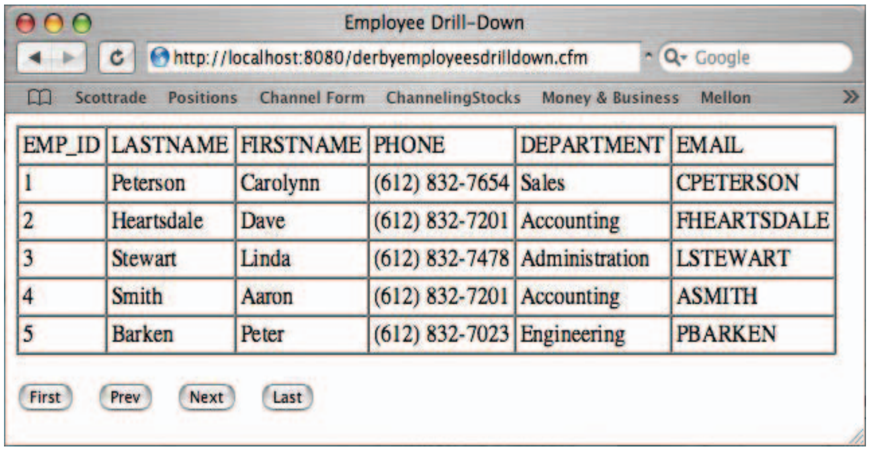
And, as if what we've shown so far isn't enough, come back next month for Part 3. We'll show you how to package your CFEverywhere application to deliver the ultimate user-experience.

To access your application, the user will be able to:

1. Download the application (or insert a CD).
2. Double-click an icon.
3. That's it! There is no Step 3.

Enjoy!

To download the files used in this article go to http://philcruz.com/cfeverywhere/downloads/cfeverywhere_files_part2.zip.


Employee Drill-Down
http://localhost:8080/derbyemployeesdrilldown.cfm
Google
Scottrade Positions Channel Form ChannelingStocks Money & Business Mellon
First Prev Next Last

EMP_ID	LASTNAME	FIRSTNAME	PHONE	DEPARTMENT	EMAIL
1	Peterson	Carolynn	(612) 832-7654	Sales	CPETERSON
2	Heartsdale	Dave	(612) 832-7201	Accounting	FHEARTSDALE
3	Stewart	Linda	(612) 832-7478	Administration	LSTEWART
4	Smith	Aaron	(612) 832-7201	Accounting	ASMITH
5	Barken	Peter	(612) 832-7023	Engineering	PBARKEN

Figure 3: Employee drill-down

About the Authors

Phil Cruz is a Macromedia Certified Advanced ColdFusion developer and has over 12 years of experience in the computing industry. He is responsible for www.mach-ii.info, a community site for the Mach-II framework. As a micro-ISV, he created Tracking Tools, an easy-to-use bug tracking application built with Mach-II (www.tracking-tools.com).

phil@philcruz.com

Dick Applebaum has been involved with computers (and their predecessors) since 1956, including 16 years with IBM. Dick and two partners opened the fifth retail computer store in Silicon Valley in 1978. Eleven years later, he sold the stores and didn't touch a computer for seven years (when he discovered the Web). Since 1997, Dick has been developing applications using ColdFusion.

dicklacara@mac.com

New Globalization Features in CFMX 7

All good things come to those who wait

By Paul Hastings

Lost amongst the hubbub over CFMX's super-cool new features like PDF generation, reporting, event gateways, admin API, and so on, was a pretty major enhancement to ColdFusion's globalization (G11N) functionality. In one "stealthy" upgrade Macromedia has quite literally managed to bring the whole world to ColdFusion developers. And while it's been some time in coming, it's certainly worth the wait. In this article I'll review some the new G11N features in ColdFusion MX 7 and hopefully launch you on your way towards developing ColdFusion applications for the world.

ColdFusion Through the Ages

Let me kick things off with a brief history of ColdFusion G11N. To me, ColdFusion has always been useful for developing localized web applications, even when it didn't particularly want to. I started using ColdFusion with version 1.5, well before I ever heard of internationalization (I18N) or Unicode or any of the other gobbledygook that I now see and use on a daily basis. At that time, getting it to work with Thai language data was about all we cared about and it was surprisingly easy considering there was nothing built-into ColdFusion that was even remotely I18N. We did encounter some obstacles, in particular, Thai dates used to vex us no end. But this was all just part of the general sea of I18N chaos you had to swim in back in those "good old days." For instance, hardly any databases or editors supported Thai.

ColdFusion 3 and 4 made some great strides in I18N even though these were mainly indirect. While there were still no directly related I18N tags or functions, the introduction of C++ CFX tags solved several nagging localization issues, in particular Thai dates and collation (our main backend database at the time was MS SQL Server 6.5, which couldn't sort Thai properly even if you held a gun to its head). So we made do.

The release of ColdFusion 5 saw the first set of purpose-built I18N functions (see Table 1 for a list) along with the introduction of the concept of "locales" to ColdFusion. Pretty nifty, but these "official" locales were nothing to write home about in terms of global coverage. While these "official" CF locales did cover most of the major Internet markets at the time (see Table 2), they didn't support Thai (which was a major Internet market for us). Another indirect upgrade in functionality, the easier use of Java classes via the createObject function, did help quite a bit as we were then able to leverage Java's I18N resources.

DateConvert	LSCurrencyFormat	LSIsNumeric	LSParseNumber
GetHttpTimeString	LSDateFormat	LSNumberFormat	LSTimeFormat
GetLocale	LSEuroCurrencyFormat	LSParseCurrency	
GetTimeZoneInfo	LSIsDate	SetLocale	
LSIsCurrency	LSParseDateTime	LSParseEuroCurrency	

Table 1: ColdFusion 5 I18N Functions

Pretty locale name	Java locale
Dutch (Belgian)	nl_be
Dutch (Standard)	nl_NL
English (Australian)	en_AU
English (Canadian)	en_CA
English (New Zealand)	en_NZ
English (UK)	en_GB
English (US)	en_US
French (Belgian)	fr_BE
French (Canadian)	fr_CA
French (Standard)	fr_FR
French (Swiss)	fr_CH
German (Austrian)	de_AT
German (Standard)	de_DE
German (Swiss)	de_CH
Italian (Standard)	it_IT
Italian (Swiss)	it_CH
Norwegian (Bokmal)	no_NO
Norwegian (Nynorsk)	no_NO_nynorsk
Portuguese (Brazilian)	pt_BR
Portuguese (Standard)	pt_PT
Spanish (Modern)	es_ES
Spanish (Standard)	es_ES
Swedish	sv_SE

Table 2. ColdFusion 5 supported locales

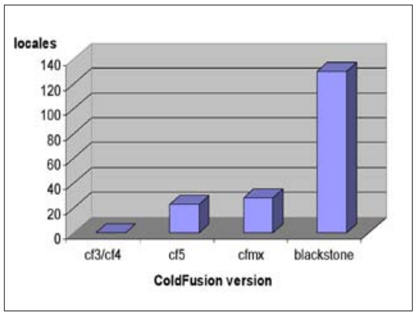


Figure 1. Officially supported ColdFusion supported locales by version

The switch to Java with ColdFusion MX was a bellwether change for ColdFusion G11N developers. First it introduced Unicode to ColdFusion, which brought a certain order out of the chaos of character encodings that was the name of the game in ColdFusion 5 and earlier. It added the important and complex CJK (Chinese, Japanese, Korean) locales to the officially supported ColdFusion locales. And being based on Java certainly made for easier and tighter integration with Java I18N libraries like IBM's killer ICU4J (http://oss.software.ibm.com/icu4j/).

A few flies in ColdFusion MX's G11N ointment included: no Thai locale (well, something we cared about anyway); the initial problems with MS Access and Unicode; and the fact that the built-in Verity engine didn't support Unicode. The ColdFusion MX 6.1 release quickly took care of the MS Access problem with the introduction of the "Access for Unicode" driver.

And then along came ColdFusion MX 7.

CFMX 7 Comes to Town

Take a good look at Figure 1. Something's sure happened to the locale support in CFMX 7; it's like ColdFusion MX on steroids. No, it's not steroids, its Java's locales.

As of MX 7, ColdFusion locale support follows Java's. Any locale that Java supports like Arabic, Russian, Hebrew, Hindi and *finally even Thai*, CFMX 7 now also supports. Further, CFMX 7 now accepts standard Java-style locale identifiers like ar_AE for Arabic (United Arab Emirates), th_TH Thai (Thailand), and so on for the setLocale() function while maintaining backwards compatibility with the old "pretty" locale names like English (US). The use of Java style identifiers is fairly significant as it's more or less a standard for identifying locales and makes for closer integration with important Java I18N classes like ResourceBundle. For a list of all the new locales, simply output the server variable Server.ColdFusion.SupportedLocales. I would like to claim the new locale support is a result of all the years of my subliminal mind-control on Macromedia staff but I'm not sure – that's the nature of subliminal mind-control techniques, you just never know.

Another significant G11N improvement in ColdFusion MX 7 is the addition of Unicode to the Verity supported encodings. You can now maintain all aspects of your ColdFusion application in one encoding, Unicode. Table 3 lists

Asian Language Pack					
Japanese	Korean	Chinese	Traditional Chinese		
Multilanguage Language Pack					
Unicode					
Western European Language Pack					
Bokmal	Finnish	Italian	Spanish	Danish	French
Nynorsk	Swedish	Dutch	German	Portuguese	
Eastern European/Middle Eastern Language Pack					
Arabic	Greek	Polish	Turkish	Bulgarian	Hebrew
Russian	Czech	Hungarian	Russian2		
Source: CFMX 7 documentation					

Table 3: Supported Verity languages in CFMX 7

Tag	Parameter	Usage
cfprocessingdirective	pageencoding	Specifies the encoding of a ColdFusion page
Cfcontent	type	Specifies the encoding in which to return the results to the client browser.
Cffile	charset	Specifies how to encode data written to a file, or the encoding of a file being read.
Cfheader	charset	Specifies the character encoding in which to encode the HTTP header value.
Cfhttp	charset	Specifies the character encoding of the HTTP request.
cfhttpparam	mimeType	Specifies the MIME media type of a file; can also include the file's character encoding.
Cfmail	charset	Specifies the character encoding of the mail message, including the headers.
cfmailpart	charset	Specifies the character encoding of one part of a multipart mail message.
Cfprocessingdirective	pageEncoding	Identifies the character encoding of the contents of a page to be processed by ColdFusion MX.
Source: CFMX 7 Documentation		

Table 4: CFMX 7 G11N tags

Function	Parameter	Usage
getLocale	-	Returns the current locale setting.
getLocaleDisplayName	-	Returns the name of a locale in the language of a specific locale. The default value is the current locale in the locale's language..
lsCurrencyFormat	-	Converts numbers into a string in a locale-specific currency format.
lsDateFormat	-	Converts the date part of a date/time value into a string in a locale-specific date format.
lsEuroCurrencyFormat	-	Converts a number into a string in a locale-specific currency format.
lsIsCurrency	-	Determines whether a string is a valid representation of a currency amount in the current locale.
lsIsDate	-	Determines whether a string is a valid representation of a date/time value in the current locale.
lsIsNumeric	-	Determines whether a string is a valid representation of a number in the current locale.
lsNumberFormat	-	Converts a number into a string in a locale-specific numeric format.
lsParseCurrency	-	Converts a string that is a currency amount in the current locale into a formatted number.
lsParseDateTime	-	Converts a string that is a valid date/time representation in the current locale into a date-time object.
lsParseEuroCurrency	-	Converts a string that is a currency amount in the current locale into a formatted number. Requires euro as the currency for all countries that use the euro.
lsParseNumber	-	Converts a string that is a valid numeric representation in the current locale into a formatted number.
lsTimeFormat	-	Converts the time part of a date/time value into a string in a locale-specific date format.
SetLocale	-	Specifies the locale setting.
CharsetDecode	encoding	Converts a string in the specified encoding to a binary object.
CharsetEncode	encoding	Converts a binary object to a string in the specified encoding.
GetEncoding		Returns the character encoding of text in the Form or URL scope.
SetEncoding	charset	Specifies the character encoding of text in the Form or URL scope. Used when the character set of the input to a form, or the character set of a URL, is not in UTF-8 encoding.
ToBase64	encoding	Specifies the character encoding of the string being converted to Base 64.
ToString	encoding	Returns a string encoded in the specified character encoding.
URLDecode	charset	Specifies the character encoding of the URL being decoded.
URLEncodedFormat	charset	Specifies the character encoding to use for the URL.
CharsetDecode	encoding	Converts a string in the specified encoding to a binary object.
Source: CFMX 7e Documentation		

Table 5: CFMX 7 G11N functions

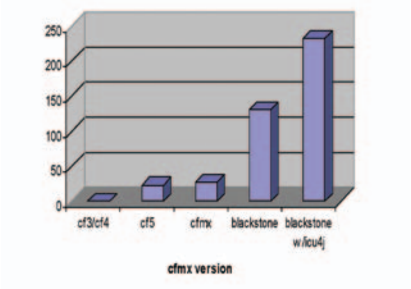
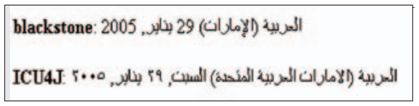


Figure 2. ColdFusion vs ColdFusion/ICU4J locales


blackstone: 2005 ,العربية (الإمارات) 29 يناير

ICU4J: العربية (الامارات العربية المتحدة) السبت, ٢٩ يناير, ٢٠٠٥

Figure 3. Comparison of ICU4J/CFMX locale date format output for ar_AE locale

the new languages included with the built-in Verity.

Needless to say, the new G11N support flows pretty much throughout all of the functionality in CFMX 7. For example, the new CFDOCUMENT tag handles all the new locales including the so-called BIDI (bidirectional) locales like Arabic and Hebrew that have right-to-left writing systems. The vastly improved CFFORM and its related tags are now also locale sensitive (though the Flash format forms still have issues with BIDI text). As far as tags/functions directly related to G11N, these are briefly explained in Tables 4 and 5.

But...

Just prior to ColdFusion MX 7 being released in February, the ICU4J project released version 3.2 of its Java library. One of the more trumpeted parts of this release was the inclusion of the Unicode Consortium's Common Locale Data Repository (CLDR, http://www.unicode.org/cldr/), which contains significantly more locales (232 with 60 more now in beta testing) than core Java (130). Figure 2 graphs the locale coverage for plain ColdFusion and ColdFusion with ICU4J. Besides the CLDR's increased locale coverage, there were some also differences between locale specific formatting in locales that both CLDR and Java supported. This is especially noticeable in Arabic locales where core Java uses European instead of Arabic-Indic digits, though there are other, sometimes subtler, format differences. See Figure 3 for an example using a simple date format for the ar_AE Arabic (United Arab Emirates) locale. Notice that I'm using the term "core Java" here, not CFMX 7. Since it derives its locale information from Java, any locale issues present in Java will also "bubble-up" in the new version of CFMX.

This leaves ColdFusion G11N developers with a choice to make. Use the Java supported locales and get the elegant simplicity of built-in ColdFusion tags and functions? Or use ICU4J with ColdFusion and get the larger, perhaps "better" locale coverage provided by the CLDR? It's a real head-scratcher.

Conclusion

ColdFusion MX 7 marks a significant improvement in G11N functionality for CF developers. It provides a huge increase in locale coverage compared to ColdFusion MX and improved Unicode in Verity. If you're a G11N developer, it's well worth the upgrade.

About the Author

Paul Hastings, who after nearly 20 years of IT work is now a perfectly fossilized geologist, is CTO at Sustainable GIS, an agile consulting firm specializing in Geographic Information Systems (GIS) technology, ColdFusion Internet and intranet applications for the environment and natural resource markets, and of course globalization. Paul is based in Bangkok, Thailand, but says that's not nearly as exciting as it sounds.

paul@sustainableGIS.com

Create a Blog, Win ColdFusion MX 7

The first of a new series

By Simon Horwith

This is the first instalment of a new column that is designed to challenge developers, show-off their solutions, and give away some cool prizes in the process.

In addition to describing the challenge as well as the winning entry from the submissions to the last month, each month's instalment will also describe the prize at stake, the deadline for entries, and the criteria that will be used for judging. In the event that I don't receive any entries or any entries worthy of winning a challenge, I'll either write my own solution or write a partial solution for everyone to build on. (In case you're wondering, no I will not keep the prize for myself if I end up having to write the solution myself.)

Though the criteria will change with each challenge, there are a few criteria that will always play a part in the judging. The code should be flexible, portable, and easy to read and maintain. In other words, it shouldn't be difficult to install, to run multiple copies of, to integrate with another application, or to modify. I recommend including a short "readme" instructing how to install and configure the application. Note that any submission to a challenge will be judged by installing it on a ColdFusion MX Developer Edition J2EE installation on JRun with Apache. That shouldn't make a difference if the submissions are well written, but I thought I'd mention it. If you include reasoning behind the decisions made in coding a solution, it may help me to judge – but that's up to you when you submit your solution.

The Challenge

This month, the challenge is to create a blog application. Now, I know that creating a blog is a lot of work… no; I don't expect anybody to do this from scratch. Ray Camden has donated his blog code base as a starting point. Go to http://ray.camdenfamily.com/blog352.zip to download the current version of his blog. Your challenge is to add any new features you feel useful and to rework the existing code in any way that you see fit. Is that all? No. The new blog code base should also take advantage of as many new features in ColdFusion MX 7 as developers see fit.

That's one of only two requirements – you must make use of new features in ColdFusion MX 7 in order to be considered for the prize. I'm not looking for specific enhancements – I'm curious to see what everyone comes up with. Certainly, I anticipate entries that take advantage of the new <CFFORM> functionality, document and/or report generation, and the new application event framework. There are some great new localization, validation, and XML features in ColdFusion MX 7 that I also anticipate seeing entries make use of.

The sky's the limit… how about using the new event gateway to allow IM and/or SMS blog entries? These are just a few suggestions…knock yourself out and have fun with it!

The second requirement is that anyone using the existing blog code base must be able to easily upgrade to this new version. If you want to add support for other databases, for saving all of the data in XML files, or continue to leave the existing database options that's fine, but be sure that there's a utility or easy migration path for anyone wishing to upgrade. This challenge has a submission deadline of April 1, so no submissions received after April first will be considered. In order to enter the contest, send your submission in zip format to myself at simon@horwith.com.

The Prize

For this first challenge, I'm offering a super-cool prize package. Dave Gruber and the rest of the product team at Macromedia have graciously donated a copy of ColdFusion MX 7 standard edition for our lucky winner! That's right – a licensed copy of the newest version of ColdFusion!

To help our lucky winner learn more about all of the great new features in their prize, the winner will also receive free entrance to the CFUnited conference in Washington, DC! Michael Smith and Liz Fredrick of TeraTech (the conference organizers) were kind enough to donate a free entrance to give away with the copy of ColdFusion MX 7 donated by Macromedia. For those of you not familiar with the conference, CFUnited was formerly known as "CFUN" and is the premier annual ColdFusion event. This year they expect 1,000 attendees, and the speaker and topics roster looks very promising. You can go to http://www.cfunited.org/ to find out more about this terrific 3-day event.

What could be better than a free copy of the best version of ColdFusion to date and complimentary entry to the largest ColdFusion conference of the year? I look forward to seeing what everyone submits… good luck!

About the Author

Simon Horwith is the editor-in-chief of ColdFusion Developer's Journal. Simon is a Macromedia Certified Master Instructor and a member of Team Macromedia. He has also been a contributing author of several books and technical papers. You can read his blog at www.horwith.com
simon@horwith.com

XML Gets a Facelift in CFMX 7

Macromedia's latest offering includes sorely needed enhancements to existing XML tools

By Alex Sherwood

Developers rejoiced and glowed with XML elation when CFMX was released: finally, a set of native XML functions that eliminated the need for flaky CFXs, heart pounding instantiations of the MSXML COM object, and unspeakably complex string parsers that sent shivers down the spines of even the most seasoned developers.

Elation, however, quickly turned to frustration. No sooner were developers' parched lips whetted with the power of XPATH searches and attribute plucking via the XMLAttributes structure, than cries of "More" could be heard. ColdFusion's XML tags and functions were enough to allow developers to build serious XML processing into their web applications, but not enough to hit the "big time." It was akin to buying a Corvette with no windshield – the engine got you up to 180MPH, but without the windshield, it ain't much fun.

Simple tasks such as returning attributes from XPATH searches and native XSD/DTD validation were not possible. Hacks, work-arounds, and yet more installation of third-party software were needed to address these shortcomings.

Enter "Blackstone" – or, as of this month, CFMX 7. Macromedia's latest offering includes sorely needed enhancements to existing XML tools, as well as a host of new functions that bring the total ColdFusion XML toolkit into the 21st century.

Put Down Your CFFILE Tags

A significant enhancement to both existing and brand new functions is the ability in CFMX 7 to pass XML data to functions in a variety of ways. CFMX previously required developers to represent XML as a string variable or a native CFML XML document object. CFMX 7 introduces the ability to specify both absolute and relative paths, as well as the use of the HTTP, HTTPS, FTP and FILE protocols.

With CFMX 7, you can take your pick of a string, a local path, a relative path (using the current template as the root) such as "xmlfiles/data/myfile.xml", or a pick of any valid URL that uses the HTTP, HTTPS or FTP protocols. All of the following uses of the XMLParse() function are now perfectly acceptable:

```
XMLParse("C:\somefolder\somefile.xml");
XMLParse("somefolder\somefile.xml");
XMLParse("ftp://xmldata.foocompany.com/myfile.xml");
XMLParse("http://#application.appxmlsore#");
```

Making a Good Thing Better

Many of the pre-Blackstone XML tags and functions you've come to know and love (or hate) have been updated to make them a little more palatable to the discerning tastes of today's more sophisticated developers. Here are some highlights:

XmlParse(xmlText [[, caseSensitive], validator])

A new parameter, *validator*, accepts a reference to a DTD or XML schema in all of the ways mentioned above (string, path, URL, etc.), and will validate the *xmlText* parameter against the schema or DTD. If an empty string is passed as *validator*, CFMX will attempt to validate the XML documents against a DTD or schema embedded within the XML document itself. If no *validator* parameter is specified, the function will simply parse the document without attempting any validation.

If the document fails validation, ColdFusion stops processing and throws a structure of arrays. This structure is the same structure thrown by the XmlValidate() function, which is covered in detail later in this article.

XmlTransform(xml, xsl[, parameters])

This enhancement is sure to put smiles on some faces. Passing parameters to an XSL style sheet is now as simple as building a struct with the name/value pairs of the XSL:PARAM elements you want to update, and then passing this structure as the third argument of the *XmlTransform()* function.

XmlSearch(xmlDoc, xPathString)

One of the biggest gripes voiced by developers was the inability of the *XmlSearch()* function to return anything but element nodes. There were several ways to work around this limitation – all of them about as fun as a root canal without Novacaine.

CFMX 7 introduces support for both element and *attribute* nodes being returned from an *XmlSearch()* function call. Together with the new ***IsXMLAttribute()*** and ***XmlGetNodeType()*** functions, it's now a breeze to both identify and work with XML attribute nodes.

XmlElemNew(xmlObj[, namespace], childName)

XML element nodes can now include namespaces at creation time. You may specify an optional URI of a namespace for which the newly created element should belong. For applications that contain documents with a blend of XML vocabularies, this is a welcome addition to the toolset.

<CFXML>

Another sorely needed enhancement is support for allowing an XML declaration when parsing text into a native XML object via the <CFXML> tag. Pre-Blackstone, <CFXML> would throw an exception if an XML document contained an opening declaration. Both <CFXML> and ***XmlNew()*** now support XML declarations.

New Kids on the Block (Not Those New Kids...)

CFMX's shortcomings in areas of XML document validation and low-level document manipulation have been addressed in CFMX 7 with a handful of new functions that seriously galvanize the overall processing package.

XmlValidate(xmlDoc[, validator])

The *XmlValidate()* function does exactly what is says – validates an XML document against a reference to a DTD or XML schema. The XML document supplied in the *xmlDoc* argument is validated against a valid DTD or schema supplied in the *validator* argument. If the *validator* argument is omitted, the function looks for a reference to a DTD or XML schema embedded in the *xmlDoc* argument.

This function always returns a structure of arrays containing a validation status and optional validation/warning errors. The structure contains the following keys:

Errors **–** An array containing any validator messages. These errors occur when the supplied XML document *does not* conform to the specified DTD or schema.
FatalErrors **–** An array containing any validator fatal error messages.
Status **–** a Boolean value: **true** if the document is valid and **false** if the validation fails
Warning **–** An array containing validator warnings. A well-formed and valid document can produce warning messages.

As you can see, there's plenty here for you to chew on. The power of validation with DTDs and XML schemas is now only a function call away.

XmlGetNodeType(xmlNode)

This little gem returns the type of XML document node passed as the *xmlNode* argument. Some of the possible node types returned: ATTRIBUTE_NODE, ELEMENT_NODE, DATA_SECTION_NODE, TEXT_NODE, and COMMENT_NODE. For applications that require extremely granular XML processing functionality, this function will come in handy.

IsXmlNode(value)

This function performs a test for several XML node types, such as the document object itself, element nodes in the document object, and most *XmlNode* objects contained in an element's *XmlNodes* array.

IsXmlAttribute(value)

This new addition is a nice side dish served with the **XmlSearch()** function. As its name states, this function will return **true** if the *value* argument is a XML DOM attribute node. For XPATH searches for which the element type returned is not known, this is an efficient method to determine if the search returned an attribute node rather than an element node.

Next Steps

There's still more. In addition to the new and enhanced features in this article, CFMX 7 also introduces over a half dozen new functions specific to building SOAP functionality into your applications. For a full and complete listing of the new CFML XML library, be sure to check out the Release Notes and updated CFML Language Reference for CFMX 7.

About the Author

Alex Sherwood is the Director of Internet Services at Pierce, Hamilton & Stern, one of the nation's largest collection agencies. Alex has been working with ColdFusion and other Internet technologies since 1997.

asherwood@phs-net.com

ProjectFusion: ColdFusion Project Server Integration

Manage tasks, resources, and projects

By Jon Hirschi

Most people are familiar with Microsoft Project – project management software that allows you to manage many aspects of projects including tasks, due dates, progress, and people assigned to specific tasks.

Microsoft Project Server is an enterprise storage product that facilitates sharing project plans created with Microsoft Project with other people in a company, especially other project managers. Project Server enables you to manage a workforce by tracking all projects in common and giving you a view of all projects that an individual may be assigned to. Chances are that any large organization will have more than one project manager and using a centralized storage point allows for better group management of projects. In this kind of environment, Project Server acts as the central repository and Microsoft Project acts as a client to access that information. Project Server also provides a Web interface for viewing the status of projects.

The principal (recommended) way to integrate with Microsoft Project is by using the integrated Project Data Service (PDS) that's included with Microsoft Project Server. PDS is a SOAP Web service that translates incoming requests and manages their entry into the system. The most common integration pieces are external user management, timesheet systems, and ticketing systems. The PDS exposes pieces of the puzzle that allow you to integrate with these external systems and to integrate all the parts of your organizational project systems.

The PDS will allow you to create projects, tasks, resources, as well as log time against projects and run administrative reports on information in the system. Granted, for most integration you'll want to use the Microsoft hooks into other Microsoft products. For example, Project Server boasts excellent integration with Microsoft's Outlook Exchange, Active Directory, and Sharepoint. If those are your only integration points, any work you would need to do is done by Microsoft, but if you need to work with a non-Microsoft product, read on.

Often times the largest integration obstacle is simply learning how a product works, and that can easily be the case with Microsoft Project. The initial learning curve of Microsoft Project may be steep but it's worth the hurdle to reach familiarization with the application before and during your implementation. You can retrieve a copy of the Microsoft Project Server API help files from this location: www.microsoft.com/downloads/details.aspx?FamilyID=4D2ABC8C-8BCA-4DB9-8753-178C0D3099C5&displaylang=en.

I'll now focus on the basics of the PDS and share some code that you can use in your organization to hook into Project Server.

Authentication

The first thing you have to do in any integration is authenticate with the other system. In Project Server land authentication is a three-part process: authenticate with Project Server, create the session, and authenticate with PDS Web services. If you're trying this on your local system, the best way to debug and observe the authentication steps is to turn on prompting for all cookies and turn off automatic forwarding in your Web browser.

To authenticate with Project Server:

```
 <cfhttp method="get"  url="http://#projectserver#/projectserver/lgnPSAu.
asp"
   redirect="No" timeout="60">
    <cfhttpparam name="un" value="#pjUname#" type="url">
    <cfhttpparam name="pwd" value="#pjPwd#" type="url">
</cfhttp>
```

Next take the cookie from the Project Server login page and pass it to the session manager:

```
<cfhttp method="post" url="http://#projectserver#/projectserver/SesStart.
asp?ref=lgnpsau.asp&au=0"
  redirect="No" timeout="60">
  <cfhttpparam name="PjSessionID" value="#urldecode(PjSessionCookie
)#"
type="Cookie">
</cfhttp>
```

Passing in the cookie to Project Server instantiates the session on Project Server and initiates the security system.

```
<cfhttp method="post"  url="http://#projectserver#/projectserver/lgnpsau.
asp?err=0"
 redirect="No" timeout="60">
   <cfhttpparam name="PjSessionID" value="#urldecode(PjSessionCookie)#"
type="Cookie">
</cfhttp>
```

After the third cfhttp call, you'll receive an XML session cookie to use with Web services. Each and every Web service request must be accompanied with the XML session cookie provided in the last step. This cookie serves to associate your session with the system and Project Server uses this to manage the security and show you the correct projects that your username has access to. To make things easier, create an object and function to send the cookie and the XML request to Project Server.

```
<cfobject webservice="http://sjn-msp-02/projectserver/pds.wsdl"
name="pdshandle">

  // Function to make it easier to send requests to the web service
    function pdsSend(myxml) {
       var returnedxml = "";
        returnedxml = xmlparse(pdshandle.soapxmlrequest(sessioncookie,m
           yxml));
        return returnedxml;
   }
```

From here you just have to send SOAP requests to the Project Server to manage whatever project you're working on. The first step is to create a project using the PDS <CreateProject> method:

```
 initialProject = pdsSend("<Request><ProjectCreate><AutoPublish>1</Auto
Publish><Project><ProjectName>#projectname#</ProjectName><Title>MY Test
ColdFusion Project</Title><Company>eBay</Company><StartDate>#dateformat(n
ow(),'yyyymmdd')#</StartDate></Project></ProjectCreate></Request>");
```

You can use this method to fully create a project and tasks, if you know all of the details about the project, such as tasks, resources, and assignments. For this example, though, project creation is broken down into each respective method call to show a wider variety of methods associated with the PDS. This example creates only one project but the PDS allows you to create multiple projects in one request. A large number of PDS methods allow you to create or modify several items in one request, which can be very convenient; however, if one item fails, all other items will fail as well.

Replies from the PDS Web service always come back in the same format: an XML response with an hresult and status node. A status code of 0 means that your request was successful; anything greater than a 0 means that something went wrong and the request failed. Figure 1 shows a <CFDUMP> of the resultant XML that is returned by PDS.

Next, check out the project you just created using the Web service call <ProjectsCheckout> to add tasks, resources, and assignments to it.

```
projectcheckout =
pdsSend("<Request><ProjectsCheckout><Project><ProjectID>#projectID#</
ProjectID></Project><Project><ProjectID>1</ProjectID></Project></Proj-
ectsCheckout></Request>");
```

In the <ProjectsCheckout> Web service call, there are two projects being checked out: the project you just created from the PDS system and the ID for the global resources project. To add resources or users to Project Server, it's necessary to check out the global project because the resources will also be added to the global project. In the default configuration, the global project should have an ID of 1 and the name of resglobal, but this may be different for your configuration. It helps to keep in mind that within Project Server, users are a type of resource.

Once the global resource project is checked out, create a resource by calling the <ResourcesAdd> method.

```
resourceinfo =
pdsSend("<Request><ResourcesAdd><Resources><Resource><Name>#resource-
name#</Name></Resource></Resources></ResourcesAdd></Request>");
```

The default type of resource created is the "work" type, meaning a person. When a work resource is created with this PDS method, it automatically creates a Project Server Web access login for the resource as well. PDS returns the enterprise resource ID or Enterprise User ID (EUID) of the user that is created. The EUID is needed for successive method calls and it must be parsed out.

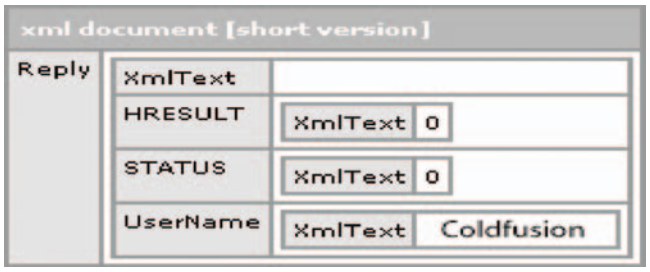
Although the resource has been added to Project Server, it still needs to be added to the project that was just created in order to assign the resource to a task. Using <ProjectResourcesCreate> you can add one or several resources to the project.

```
ProjResource = pdsSend("<Request><ProjectResourcesCreate><AutoPublish>1</
AutoPublish><ProjectID>#projectID#</ProjectID><Resources><Resource><EUID
>#resourceUID#</EUID></Resource></Resources></ProjectResourcesCreate></
Request>");
```

At this point there is a little bit of a problem. The resource was added to Project Server and an EUID was created, but unfortunately the EUID can't be used in the project for assignments. For resources in Project Server there are three IDs to choose from when trying to assign a resource to a specific task in a project. Resources have an enterprise resource ID or EUID (one that corresponds to the resource ID of the resource global project), a project resource Unique ID (UID), and a Resource ID.

To assign the resource to the task, you need to find the UID for the resource. You can accomplish this by making another request to get the details for the project using the <ProjectData> method and then taking the EUID received after adding the resource to Project Server and matching it with the EUID that's returned with the project data. The <ProjectData> method returns a complete resource object that includes the EUID and the UID for that particular project.

```
projectdata =  pdsSend("<Request><ProjectData><ProjectID>#projectID#</
ProjectID><ReturnUIDs>1</ReturnUIDs></ProjectData></Request>");
```


xml document [short version]

Reply	XmlText		
	HRESULT	XmlText	0
	STATUS	XmlText	0
	UserName	XmlText	Coldfusion

Figure 1. XML CFDUMP

Sound confusing? It helps to think of Project Server organizing everything into projects. When a resource is added to a project, it creates a Unique ID for that project. The EUID corresponds to the Unique ID for the resource in the global project.

There's one other thing that you need the project data for: to find out what the next Task ID should be. If you have installed the latest service pack from Microsoft for Project Server, you won't need to do this step. For previous versions of Project Server you'll need to find out what the next Task ID is and include it with the Project ID in the task creation method <ProjectTasksCreate>.

```
Taskxml = pdsSend("<Request><ProjectTasksCreate><AutoPublish>1</AutoPublish><ProjectI
D>#ProjectID#</ProjectID><Tasks><Task><Name>#taskname#</Name><ID>#projectTaskID#</ID><Finish>#dateformat(ta
skenddate,'yyyymmdd')##timeformat(taskenddate,'HHmmss')#</Finish><Start>#dateformat(taskstartdat
e,'yyyymmdd')##timeformat(taskstartdate,'HHmmss')#</Start></Task></Tasks></ProjectTasksCreate></
Request>");
```

Once the task has been added to the project, assign a resource to the task using <ProjectAssignmentsCreate>.

```
ProjAssign = pdsSend("<Request><ProjectAssignmentsCreate><AutoPublish>1</AutoPublish><ProjectI
D>#ProjectID#</ProjectID><Assignments><Assignment><TaskID>#projectTaskID#</TaskID><ResourceID>
#ProjectResourceID#</ResourceID><Units>100</Units><Finish>#dateformat(taskenddate,'yyyymmdd')#
#timeformat(taskenddate,'HHmmss')#</Finish><Start>#dateformat(taskstartdate,'yyyymmdd')##timef
ormat(taskstartdate,'HHmmss')#</Start></Assignment></Assignments></ProjectAssignmentsCreate></
Request>");
```

A resource can only be assigned to a task once. However, several resources can be assigned to complete a task. Assignments have their own start dates and end dates as well, because specific assignments can be less than the duration of the task. To finish it off, check the project back in and voilà, you've created your own project in Project Server!

```
projectcheckin = pdsSend("<Request><ProjectsCheckin><Project><ProjectID>#projectID#</
ProjectID></Project><Project><ProjectID>1</ProjectID></Project></ProjectsCheckin></Request>");
```

Conclusion

Once all of the above steps have been completed, you have a project in which you can manage tasks, resources, and projects. Using the Web service calls discussed in this article, you can do anything from tracking the status of projects, to managing resources and resource availability on an enterprise scale with your custom applications. Integrating with Project Server using ColdFusion's built-in processing tools is a great way to connect your enterprise applications to Microsoft's project management tools. If you're interested in doing more with Project Server, you'll find more information at these Web sites:

- The official developer's network that includes tools and information for dealing with Microsoft Project including developer guides and helpful information about integrating with Microsoft Project Server and Microsoft Project: http://msdn.microsoft.com/project
- Microsoft Project Users Group is an independent Web site that features Microsoft Project and project management resources for those interested in finding out more about how other project management professionals are doing things in the real world: www.mpug.org

About the Author

Jon Hirschi is a Senior Web Developer at eBay Inc. specializing in content and knowledge management systems. He has been using ColdFusion since version 4.0.

jhirschi@ebay.com

What Are You Searching For?

Verity helps you find it

By Ray Camden

In the January issue of *CFDJ* (Vol. 7, issue 1), Joe Cronin wrote an excellent article about Verity entitled "Optimize, Extend, and Enhance the Search Functionality in ColdFusion." In this article, Joe talked about the powerful features of the Verity search service provided in the ColdFusion MX server.

With the release of ColdFusion MX 7, a new version of the Verity engine is provided, allowing for even more powerful searching. In this article we will cover just some of the new Verity-based features. Check the documentation for more information.

Getting Started

Before we do anything, let's create a Verity collection we can use for our tests. Listing 1 creates a Verity collection based on your ColdFusion documentation. If you haven't installed the ColdFusion documentation, you can modify the code to point to any other collection of files.

This script is pretty self-explanatory so I won't go into much detail. Again, you may have to modify it if you haven't installed the ColdFusion documentation. Now that we have a simple collection, let's start looking at some new features in ColdFusion MX 7.

May I Offer a Suggestion?

Nothing is as frustrating as when a search that you think should work doesn't return anything. You know you're using the right search terms…or at least you think you are, but you simply can't get the site to return any results. What if the search engine itself could help you? The Verity engine in ColdFusion MX 7 now offers a suggestion feature. While primarily targeted to spelling mistakes, it can certainly be of use to your site visitors. The <cfsearch> tag now takes a suggestions attribute. The possible values are *always*, *never*, or a number. Obviously, if the value is *always*, suggestions will always be returned. If the value is *never*, suggestions are never returned. If you specify a number, suggestions are only returned when the search results are less than the number specified. This feature does have a small impact on performance, so you want to consider before turning it on, but most users will probably appreciate the help. Instead of simply getting a "0 results found" message, they can given a bit of help. Listing 2 works with the collection created in Listing 1.

There is a lot going on here – so let's take it line by line. First, I <cfparam> both a URL and Form variable called searchTerms, because I want to be able to search via either the URL or Form scopes. Lines 4–9 define a simple form that will let users enter search terms. Line 11 is where things become interesting. We first check to see if anything interesting exists in the searchTerms variable. If so, we perform the search using the <cfsearch> tag. There are two new attributes passed in: status and suggestions. We discussed the suggestions attribute already. Since we passed a number, we will get suggestions if the results are less than or equal to our number. We specified 0, which basically means, just provide suggestions if no results are found. The status attribute tells the <cfsearch> tag to return a structure of information related to our search. There is a bunch of useful information here, but for our purposes, we are interested in the keywords value. If it exists, it is a structure of arrays. Each key in the structure matches a word in our search query. Each key then will have an array of suggestions. The status structure also contains a score for each suggestion, but we aren't going to work with that now.

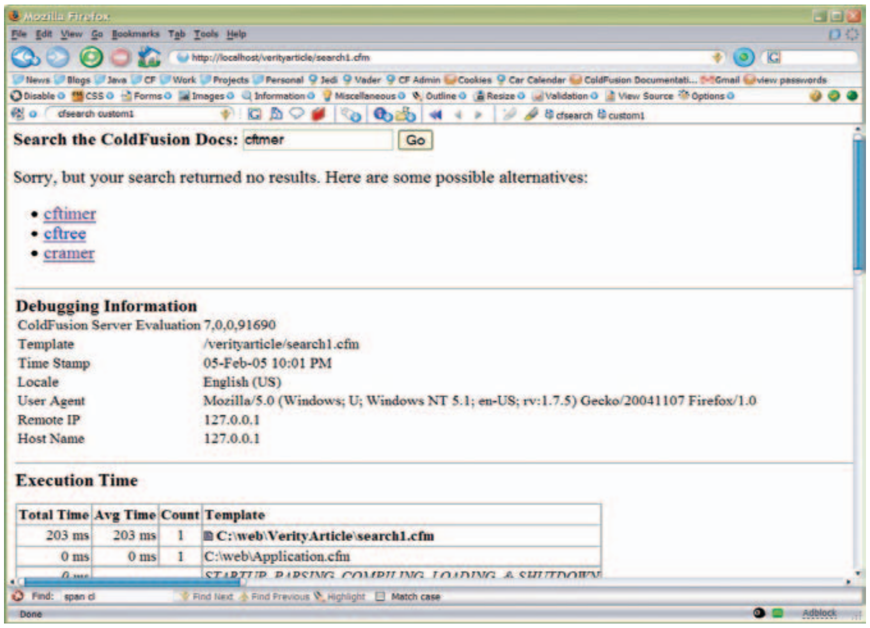
First we check to see if the keywords portion of the status structure exists. If so, we loop through each key and then through each suggestion in the array, and provide a link back to the search page. This allows users to quickly click on a link to perform a new search. Figure 1 demonstrates how a failed search for "cftmer" shows a set of suggestions, with the first one being what we probably intended, "cftimer."

Working with Results

Sometimes the problem isn't that you can't get results from your searches, but that you get too many. It would be helpful if the results told you a bit more about themselves. Verity has always returned a summary column with the results. In ColdFusion MX 7, you can now get a set of text that contains the context of the matched item. This is also useful if you are searching for multiple items and want to see which one was matched.

Three new attributes to <cfsearch> enable the use of context passages. The first is contextPassages. This is a numerical value that tells Verity the number of context passages to return. Obviously if you provide a number larger than the amount of matches in the document, you won't get additional context passages. You can also specify how Verity marks up the matched items. By default, Verity will surround the match with bold tags. However, you can specify your own markup with contextHighlightBegin and contextHighlightEnd. This can be useful for specifying your own stylesheet or other design. Listing 3 is a modified version of Listing 2. Let's focus just on the things that changed.

This is the same as Listing 2, except for the three new context-related items.



Search the ColdFusion Docs: cftmer Go

Sorry, but your search returned no results. Here are some possible alternatives:

- cftimer
- cftree
- cramer

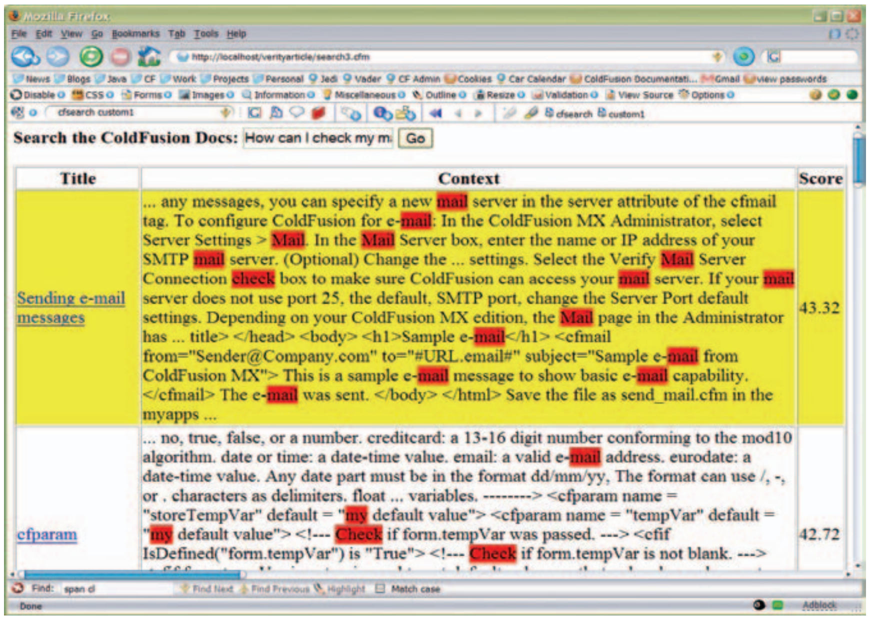
Debugging Information

ColdFusion Server Evaluation	7,0,0,91690
Template	/verityarticle/search1.cfm
Time Stamp	05-Feb-05 10:01 PM
Locale	English (US)
User Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.5) Gecko/20041107 Firefox/1.0
Remote IP	127.0.0.1
Host Name	127.0.0.1

Execution Time

Total Time	Avg Time	Count	Template
203 ms	203 ms	1	**C:\web\VerityArticle\search1.cfm**
0 ms	0 ms	1	C:\web\Application.cfm

Figure 1



Search the ColdFusion Docs: How can I check my m Go

Title	Context	Score
Sending e-mail messages	... any messages, you can specify a new mail server in the server attribute of the cfmail tag. To configure ColdFusion for e-mail: In the ColdFusion MX Administrator, select Server Settings > Mail. In the Mail Server box, enter the name or IP address of your SMTP mail server. (Optional) Change the ... settings. Select the Verify Mail Server Connection check box to make sure ColdFusion can access your mail server. If your mail server does not use port 25, the default, SMTP port, change the Server Port default settings. Depending on your ColdFusion MX edition, the Mail page in the Administrator has ... title> </head> <body> <h1>Sample e-mail</h1> <cfmail from="Sender@Company.com" to="#URL.email#" subject="Sample e-mail from ColdFusion MX"> This is a sample e-mail message to show basic e-mail capability. </cfmail> The e-mail was sent. </body> </html> Save the file as send_mail.cfm in the myapps ...	43.32
cfparam	... no, true, false, or a number. creditcard: a 13-16 digit number conforming to the mod10 algorithm. date or time: a date-time value. email: a valid e-mail address. eurodate: a date-time value. Any date part must be in the format dd/mm/yy, The format can use /, -, or . characters as delimiters. float ... variables. --------> <cfparam name = "storeTempVar" default = "my default value"> <cfparam name = "tempVar" default = "my default value"> <!--- Check if form.tempVar was passed. ---> <cfif IsDefined("form.tempVar") is "True"> <!--- Check if form.tempVar is not blank. --->	42.72

Figure 2

Notice that I do something a bit with the markup for the context matches. Instead of simple HTML or CSS, I specify the string __MATCHBEGIN__ and __MATCHEND__. What is the reason for that? In our collection, we have indexed HTML documents. That means our context passages will have HTML in them. I don't want the HTML from the results to mess up my page, therefore, I do some massaging on the value:

```
<cfset cleanedContext =
htmlEditFormat(context)>
<cfset cleanedContext =
replace(cleanedContext,"__MATCHBEGIN__","<span
style=""background-color:red"">","all")>
<cfset cleanedContext =
replace(cleanedContext,"__MATCHEND__","</
span>","all")>
```

First we use htmlEditFormat to escape all HTML in the result. I then search for and replace the strings I defined earlier. I insert <span> tags to highlight the matches with a lovely red color. (Hey, I'm a coder, not a designer.)

Keep It Simple, Stupid

Out of the box, Verity has very serious searching powers. If you know the syntax, you can perform many particular, special types of searches. However, if you aren't aware of how Verity works, things can be confusing. For example, searching for "email cfmail" means to search for the phrase, "email cfmail." Some users may assume that any document with either "email" or "cfmail" should return a match. You can enable a simpler understanding of your search criteria by specifying a type attribute to your search tag. There are a set of possible values, but we are going to focus on the "Internet" type that allows for free-form questions in your search criteria. Listing 4 is yet another modification to our Listing 3. There is only one item changed, and that is the <cfsearch> tag.

As we just said, the only difference here is the type attribute. To get an idea of how powerful it is, consider Figure 2. This is the result of entering: "How can I check my mail with cfmail?" The first result concerns sending e-mail – which isn't exactly what we wanted but is close – and should lead us down the right path. The point is – the same query used in our Listing 1 would have returned nothing.

Categorize This!

Verity is an amazing tool when it comes to searching text. However, filtering the results of a Verity search has never been as easy as it is in SQL. While you can perform any number of specialized textual searches, you can't filter your results based on properties such as date, numbers, etc. For example, it's trivial in SQL to return results where the records fall within a date range, a price lies between two values, and a related column points to a particular foreign key. With all that power, you don't mind so much that your text searches are typically a bit limited.

Verity does have some workarounds for this. In the previous version of ColdFusion, you could index two custom fields. This meant that if your data came from a database, you could stuff two extra values into a collection. ColdFusion MX 7 even adds two new custom fields for a total of four. But using this feature requires modifying your search criteria and isn't very simple. Enter another new feature in ColdFusion MX 7 – Categorization. To use this feature, you must enable categories when you create your collection. Listing 5 creates a new collection and indexes a custom set of data related to sports.

As before, let's focus on the differences. Just like with Listing 1, we only create our collection if it doesn't already exist. However, this time we specify categories="yes" when we create our collection. We create two queries that we'll use to add to our collection. Note that in each case, we specify a category and a categoryTree, new features to the <cfindex> tag. The category attribute can be either a string, or a list of strings, that represents the category of the data being added to the collection. The categoryTree value represents the "path" of the category. In our case, we specify that the "football" category is "beneath" either the "American Sports" root or "English Sports" root. Another example could be:

```
category="Vocal Trance"
categoryTree="Music/Electronic/Trance"
```

Note the use of slashes in the categoryTree. This defines a hierarchy for our data. This hierarchy can contain any information. You're completely free to set up your hierarchy any way you want. Listing 6 shows a simple example of searching a collection with categories.

In the first two lines of code we search against our collection using criteria="*". This simply means to match everything. When run, those lines will match both records we entered into our collection. However, the next set of lines specify both the category and the categoryTree. In this case, the result will have only the American football team.

Last but not least – your search form can provide information about the categories in your collection. A new action to the <cfcollection> tag, categoryList, returns a structure of all the categories and category trees in your collection. This information could be used to populate a dropdown or other form control in your search form. A file included in the zip, get_cats.cfm, shows an example of the categoryList. You can download this from www.sys-con.com/coldfusion/sourcec.cfm.

But Wait, There's More…

This article covers only some of the new Verity features in ColdFusion MX 7. Please check the documentation for other new features. If you have any further questions, please feel free to contact me at ray@camdenfamily.com. Last, if you have never worked with Verity, I encourage you to do so, even if you don't immediately upgrade to ColdFusion MX 7.

About the Author

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

jedimaster@mindseye.com

Listing 1: setup.cfm

```
<!--- get all collections --->
<cfcollection action="list" name="currentCollections">

<!--- get names --->
<cfset collections = valueList(currentCollections.name)>

<!--- do we have "cfdocs" yet? --->
<cfif not listFindNoCase(collections, "cfdocs")>

  <cfcollection action="create" collection="cfdocs" path="#server.
coldfusion.rootdir#/verity/collection">

  <cfset webroot = expandPath("/")>
  <cfindex collection="cfdocs" type="path" extensions=".htm,.html"
urlPath="/cfdocs/htmldocs/" key="#webroot#/cfdocs/htmldocs"
 action="update">

</cfif>
```

Listing 2: search1.cfm

```
<cfparam name="url.searchTerms" default="">
<cfparam name="form.searchTerms" default="#url.searchTerms#">

<cfform action="#cgi.script_name#" method="post">
  <b>Search the ColdFusion Docs:</b>
  <cfinput type="text" name="searchTerms" value="#form.searchTerms#"
     required="true"
    message="Enter your search terms.">
          <cfinput type="submit" name="search" value="Go">
```

```
</cfform>

<cfif len(trim(form.searchTerms))>

  <cfsearch collection="cfdocs" criteria="#trim(form.searchTerms)#"
     suggestions="0" name="results" status="status"
        maxrows=10>

  <cfif results.recordCount gte 1>
    <table border="1">
      <tr>
        <th>Title</th>
        <th>Score</th>
      </tr>
    <cfoutput query="results">
      <cfset path = listLast(key, "\/")>
      <tr
        <cfif currentRow mod
                         2>style="background-color: yellow"</cfif>
      >
        <td><a href="/cfdocs/htmldocs/
                             #path#">#title#</a></td>
        <td>#score*100#</td>
      </tr>
    </cfoutput>
    </table>
  <cfelse>
    Sorry, but your search returned no results.
    <cfif structKeyExists(status, "keywords")>
      Here are some possible alternatives:
      <ul>
      <cfloop item="keyword" collection="#status.
        keywords#">
        <cfloop index="x" from="1" to="#arrayLen(status.
         keywords[keyword])#">
          <cfset suggestion = status.keywords[keyword][x]>
          <cfoutput>
          <li><a href="#cgi.script_name#?searchTerms=#urlEncodedFormat
            (suggestion)#">#suggestion#</a></li>
          </cfoutput>
        </cfloop>
      </cfloop>
      </ul>
    </cfif>
  </cfif>

</cfif>
```

Listing 3

```
<cfsearch collection="cfdocs" criteria="#trim(form.searchTerms)#"
 suggestions="0" name="results" status="status"
    maxrows=10 contextPassages="3"
    contextHighlightBegin="__MATCHBEGIN__"
    contextHighlightEnd="__MATCHEND__">
```

Listing 4

```
<cfsearch collection="cfdocs" criteria="#trim(form.searchTerms)#"
   suggestions="0" name="results" status="status"
     maxrows=10 contextPassages="3"
     contextHighlightBegin="__MATCHBEGIN__"
     contextHighlightEnd="__MATCHEND__"
     type="internet">
```

Listing 5: setup_categories.cfm

```
<!--- get all collections --->
<cfcollection action="list" name="currentCollections">

<!--- get names --->
<cfset collections = valueList(currentCollections.name)>

<!--- do we have "veritytest" yet? --->
<cfif not listFindNoCase(collections, "veritytest")>

  <cfcollection action="create" collection="veritytest" path="#server.
   coldfusion.rootdir#/verity/collection" categories="yes">

</cfif>

<!--- fake set of data --->
<cfset q = queryNew("id,name,body")>
<cfset queryAddRow(q,1)>
<cfset querySetCell(q,"id","1",1)>
<cfset querySetCell(q,"name","Saints",1)>
<cfset querySetCell(q,"body","The New Orleans Saints",1)>

<cfindex action="update" collection="veritytest" type="custom"
 query="q" key="id" body="body" category="football"
categoryTree="American Sports">

<!--- another fake set of data --->
<cfset q2 = queryNew("id,name,body")>
<cfset queryAddRow(q2,1)>
<cfset querySetCell(q2,"id","2",1)>
<cfset querySetCell(q2,"name","London Foo",1)>
<cfset querySetCell(q2,"body","Sorry, I don't know any English
  football teams.",1)>

<cfindex action="update" collection="veritytest" type="custom"
  query="q2" key="id" body="body" category="football"
  categoryTree="English Sports">
```

Listing 6: cat_search.cfm

```
<cfsearch collection="veritytest"  category="football" criteria="*"
 name="results">

<cfdump var="#results#">

<cfsearch collection="veritytest"  categoryTree="American Sports"
 category="football" criteria="*" name="results">

<cfdump var="#results#">
```

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

Matt Liotta

Advanced CF

Deployment/Platform

Simon Horwith

Michael Dinowitz

Tim Buntel

Charles Arehart

Christian Cantrell

Jeff Peters

Accessibility / Usability

Manager Empowered Programming

Raymond Camden

coldfusion

Steve Drucker

Michael Smith

Ben Forta

Sean Corfield

MX Integration

Lifecycle

Hal Helms

Dave Watts

Geoff Snowman

Shlomy Gantz

Boot Camp

June 29 - July 1, 2005
Washington DC Area
7th Annual ColdFusion Conference

3 full days!
New Venue
75% Bigger

TeraTech
New Atlanta
CFDynamics
Fog Creek Software
Synthis
Paper Thin
AboutWeb
MX developer's journal
ColdFusion Developer's Journal

www.cfunited.com

CFUN has become the premier CF specific event, and Michael Smith and his team deserve all sorts of praise for their hard work in pulling it all off yet again.

Ben Forta

"...this event really is the best gathering in the world for people developing or managing CF systems. It's here that we can understand what happened, hear what's happening, and learn what's going to happen. You can't beat it."

Chuck Hoffman

"Great place to network yourself and pick up new techinques and ideas. Also to meet ones peers, and see what the future holds for all those involved with ColdFusion"

Daniel Gregorio

"Introductions to the latest developing techniques. Get inspiration in new ways to develop projects. Ge"erally, to "re-kindle" your cf "fire" by being part of a group excited by and interested in cf development."

Kathleen Ballard

Produced by
TeraTech
Programming
405 East Gude Drive
Ste 207
Rockville MD 20850
www.teratech.com
301.424.3903
info@cfunited.com

Intermedia.NET...

Now serving the hottest ColdFusion.

At Intermedia.NET we go beyond the industry standard by supporting the hottest new Coldfusion software, offering power like never before. For nearly a decade, we've been providing reliable, secure hosting to thousands of companies across the globe. We can do the same for you.

Intermedia.NET's premier hosting services include:

- ColdFusion MX hosting
- Competitive plans
- Security sandboxes
- Custom tag registration
- Verity collections search engine
- Guaranteed service levels

Unprecedented power, unmatched reputation...
Intermedia.NET is your hosting solution.

INTERMEDIA.NET

Call us at: **1.888.379.7729**
e-mail us at: **sales@intermedia.NET**
Visit us at: **www.intermedia.NET**