



# BlueDragon: Another Option for CF Developers

**I**magine going to buy a new car. You are down to specifying the options for the vehicle and the salesman says, "You can have it in any color as long as it's black." If black is your favorite color that's great; but if it's not, then maybe they have just lost a customer. These days it's all about giving people choices.

Until recently, if you were developing ColdFusion applications you didn't have much choice. Macromedia was the only vendor providing a ColdFusion application server. Now, BlueDragon from New Atlanta has arrived to change all that. [Note: BlueDragon was originally a product called tagServlet developed by n-ary Limited. So it's not entirely new.] I have taken some time to check out BlueDragon, and here is what I have seen.



Reviewed By  
Phil Cruz

## What Is BlueDragon?

In a nutshell, BlueDragon is a family of Java-based products that implement the CFML (5.0) language. (There are a few CF5 tags that are not supported in BlueDragon, and support for CFMX language features is under development.) This allows you to develop and deploy CFML applications on an application server other than that provided by Macromedia. In addition, you have the ability to extend your applications with Java servlets, JavaServer Pages (JSP), or other J2EE technologies.

The product family consists of three products: BlueDragon Server, BlueDragon Server JX, and BlueDragon for J2EE. The products are priced competitively to Macromedia ColdFusion Server products, and range in cost from \$249/server-\$2499/

cpu. BlueDragon is free for development (like ColdFusion MX).

Check the New Atlanta site for details, but basically the server product is an application server to be used in conjunction with your Web server (it supports IIS, Apache, and Netscape/iPlanet) to serve CFML pages. Similar to CFMX,

it has a built-in Web server and can be used in standalone mode. The Server JX product adds the ability to serve JSP in addition to CFML. (While CFMX can execute JSPs as well, this is only available in the Enterprise Edition.)

The BlueDragon for J2EE product is quite different from the server products. Rather than being a complete application server, it is a portable CFML runtime module that allows for packaging your application into a Web application archive or WAR file and deploying onto any J2EE server. One interesting thing to note about the WAR file packaging is that BlueDragon/J2EE allows you to deploy your files as compiled binaries – without source – thereby allowing you to protect your intellectual property and source code. For a CF developer looking to migrate more seamlessly to J2EE and/or protect his or her source code, BlueDragon provides an answer.

## Installation and Administration

Since I wanted to try the BlueDragon/J2EE product, I needed to obtain a J2EE compatible server. (Again, the two more basic versions are themselves complete application servers, like CFMX.) I chose to download Tomcat, which is readily available and free for deployment. Installing Tomcat went without a hitch; I installed it on default port 8080 so as not to conflict with IIS or CFMX running on the same box. Installing BlueDragon/J2EE was straightforward as well.

Essentially, BlueDragon/J2EE ships as a set of files that you drop into the Web application directory of the J2EE server. The template files include an index.cfm that displays a page so that you can verify your installation. You access the BlueDragon Administration screen by browsing to the /bluedragon/admin.cfm relative to your application URL, for example <http://localhost:8080/myapp/bluedragon/admin.cfm>.

The BlueDragon Administrator is laid out similarly to the CF Administrator and should feel familiar to any CF developer (see Figure 1). As you would expect, there are functions to set data sources, mappings, application/session variable timeouts, scheduled tasks, and mail server settings.

The BlueDragon Administrator differs from the CF Administrator in that it does not have functions for Verity (not supported in BlueDragon), debugging settings, log files, tag restrictions, and a few other miscellaneous settings. It does include a Deploy WAR File function that ColdFusion Server does not support. As for database support, the popular databases are supported and BlueDragon ships with New Atlanta's JTurbo, which is

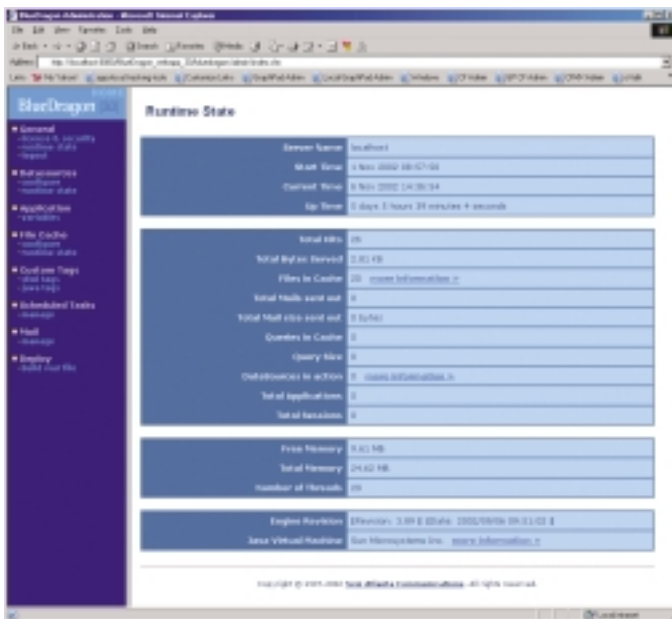


Figure 1: Screen shot of BlueDragon Administrator

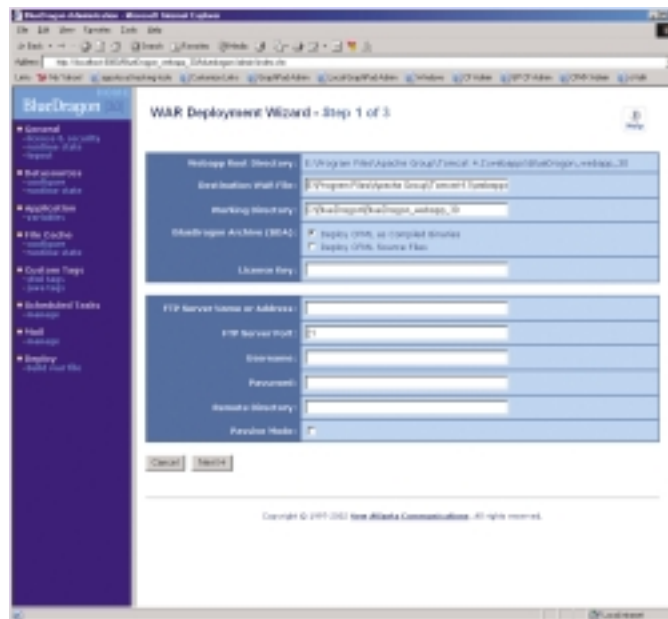


Figure 2: Screen shot of BlueDragon Administrator WAR deployment wizard

a Type 4 JDBC driver for connecting to Microsoft SQL server. I had no problems creating a data source to a Microsoft SQL server database.

### Running Some Code

For my initial test, I tried running the employee directory application for the fictitious New Line Software company that ships as one of the example applications with ColdFusion 5. I selected this application because most people would have access to the code and it does all the basic interactions with a database: add, edit, delete, and search records. I copied the code from the CF 5 examples into the Web application directory with the other BlueDragon/J2EE files, created the data source, and launched the application – it just worked.

There's really not much to report. I tested all the functionality of the application and it worked flawlessly. Anyone browsing the application would have no indication that BlueDragon, not a ColdFusion Server, was serving pages.

Creating the WAR file for deployment was straightforward. From the BlueDragon Administrator, click "build war file" to launch the WAR deployment wizard (see Figure 2). You must provide a destination for the WAR file, select if you want to deploy source files or compiled binaries, and provide a license key. The following screen will ask which database drivers and optional tags you want to

include in your WAR file. Press finish and your WAR file is created.

Upon inspecting the contents of the WAR file (which is basically a zip archive), I confirmed the source code was not included in the package. In place of the source was a compiled binary file, webapp.bda. As for deployment, WAR deployment is handled slightly differently depending on the particular J2EE server. Tomcat supports "auto-deployment" in which you simply copy the WAR file to the \webapps directory of Tomcat and restart. I was able to auto-deploy the WAR file for my sample application with no problem.

Granted, the example application is very simple under the hood. I was eager to try BlueDragon with a "real" application. I tried to run a bug-tracking application that is built with Fusebox 3, has user authentication/authorization, extensive use of custom tags, and user-defined functions.

Before running the application, I did read the product documentation, which includes a CFML Compatibility Guide that discloses certain CFML tags that may be unsupported in BlueDragon or are supported, but with differences. I found the documentation to be clear, informative, and well written. Upon trying to run the bug-tracking application I ran into some issues that were both documented and undocumented.

As for documented issues, the application utilizes CFASSOCIATE and CFOBJECT with COM objects that are not sup-

ported in BlueDragon. I also found a custom tag that still uses the deprecated parameterExists() function. In CF5, parameterExists() is deprecated but still valid syntax; it is not valid syntax in BlueDragon. None of these are showstoppers and the code could be modified to remove dependence on them. One suggestion for New Atlanta is to provide a "code compatibility analyzer" utility that you could run on your source code to flag such issues, similar to what CFMX has.

As for undocumented issues, I ran into a bug involving structure deserialization using WDDX. Also, I ran into an issue related to <cfmodule> syntax. In CF5/CFMX, you can call a tag like <cfmodule template="mytag.cfm"> </cfmodule> and because it has an end tag it will be executed twice. You can also code it like <cfmodule template="mytag.cfm"/> (like XHTML syntax); that implies the end tag and will also execute the tag twice. Using the latter form in BlueDragon, the code was executing only once. I raised both of these issues on the BlueDragon mailing list and New Atlanta was quick to confirm the bugs as well as indicate that they would be resolved in an upcoming service pack.

So, while I did run into some issues, none proved to be insurmountable and I'm quite pleased at the prospect of having more deployment options for my application. I have not done any performance/load testing. From what I hear from

## product review

New Atlanta, BlueDragon generally outperforms ColdFusion Server 5 and is on par with CFMX. As usual, do your own testing and your mileage may vary.

### CFML – The New Standard?

I think we can agree that competition is a “good thing.” Competition can put pressure on vendors to provide quality service/products at competitive prices. However, now that BlueDragon has entered the game, what happens if New Atlanta extends the CFML language with their own features? In fact, they have already indicated their intent to do this by implementing a <CFFORWARD> tag that would allow you to forward the request to a JSP, similar to how the <jsp:forward> tag does.

The potential is there for a “fracturing” of the CFML “standard” and having code that will run on BlueDragon, but have issues on ColdFusion Server or vice versa. Having endured the browser-compatibility wars, I for one would not look forward to having to write conditional code all throughout my applications to deal with “CF server compatibility” issues.

It is possible, though perhaps unlikely, that Macromedia will implement a standards process similar to the Java Community Process that Sun has, and allow other vendors to provide input to the future of CFML. Lacking that, New Atlanta has said that there is one standard for CFML and that is what is provided by Macromedia in terms of published documentation, and what is implemented in the actual server products.

### Summary

BlueDragon is not a “ColdFusion killer” and it’s not meant to be. If anything it’s an affirmation and indication of maturity of the ColdFusion market. It addresses some issues that Macromedia’s ColdFusion does not. BlueDragon is not 100% compatible with ColdFusion (and doesn’t claim to be) or bug free (no product is), but I found it to deliver as advertised. It’s another choice available to us as developers, and one worth checking out.



### About the Author

*Phil Cruz is a certified ColdFusion developer. He has over 10 years of experience in the computing industry and specializes in the design and development of Web-based applications.*

[philcruz@yahoo.com](mailto:philcruz@yahoo.com)

### Vitals

BlueDragon • New Atlanta Communications, LLC  
100 Prospect Place • Alpharetta, GA 30005  
Phone: 678 256-3011 • Fax: 678 256-3012  
E-mail: [info@newatlanta.com](mailto:info@newatlanta.com) general [sales@newatlanta.com](mailto:sales@newatlanta.com) sales  
Test Environment: Windows 2000 Server



## Editorial —continued from page 5

to mention, because it’s viewed by many in our always-interesting industry as a touchy subject, and one worth covering.

We’ve commissioned a product review of New Atlanta’s BlueDragon, a Java-based server that also compiles CFML pages. The current release, 3.0, supports most of the standards of the CFML 5.0 language, with an MX-compatible release under construction. Phil Cruz’s in-depth review hits upon some of the high and low points of the server, and it’s worth a read.

We’ve had several e-mails in the past few months from people asking us to give the product a good look, and even more e-mails from people wondering where it fits into the ColdFusion market. I think that the product is good for the ColdFusion market because it shows the embrace of CFML as a standard language and, on the server side, the ColdFusion world as a maturing market.

Till next month – happy coding!



## Q&A —continued from page 7

**the user input “Tom, Dick, Harry”, I’m trying to get the query to execute something like “SELECT \* FROM TABLE WHERE Name LIKE '%Tom%' OR Name LIKE '%Dick%' OR Name LIKE '%Harry%'”.**

**How can I create the query to do what I want?**

A: The question is really, “How do I dynamically build queries?” and it’s an issue that CF developers need to address in almost every application. The bottom line is this: you can use CF tags nested inside your CFQUERY tag to dynamically generate the necessary SQL. For your scenario, there are several ways to solve the problem, but here’s my suggestion (see Listing 3).

### Listing 3

```
<CFQUERY NAME="qSearch" DATASOURCE="YourDNS">
SELECT * FROM TABLE
WHERE Name = 'PlaceholderText'
<CFLOOP LIST="#Form.SearchText#" INDEX="SearchItem">
    OR Name LIKE '%#Trim(SearchItem)#%'
</CFLOOP>
</CFQUERY>
```

First, you’ll need to create what I call a “placeholder” in your SQL. This is a valid SQL statement that you put immediately after the WHERE clause, but it has no real impact on your query; it’s just there to make the rest of your query easier to generate. In this case, I’ve used the statement “WHERE Name = ‘PlaceholderText’”. The likelihood of having a name equal to “PlaceholderText” is pretty slim.

Next, you need to dynamically build your OR statements by looping over the list that the user submitted. For example, if your form field is called “SearchText” and they submit “Tom, Dick, Harry”, you would use CFLOOP to loop over the values in Form.SearchText. Inside your loop, simply create the OR condition for your query using the value of the list index as your search criteria. Be sure to use the Trim() function to eliminate any spaces that the user may have placed after the commas. The end result will be a SQL query that contains an OR clause for each term that the user supplies.



### Info

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