

# FusionDebug review

Posted At : August 6, 2007 8:29 PM | Posted By : Michael Smith

Related Categories: Reviews

## Fusion Debug review

*By George Murphy*

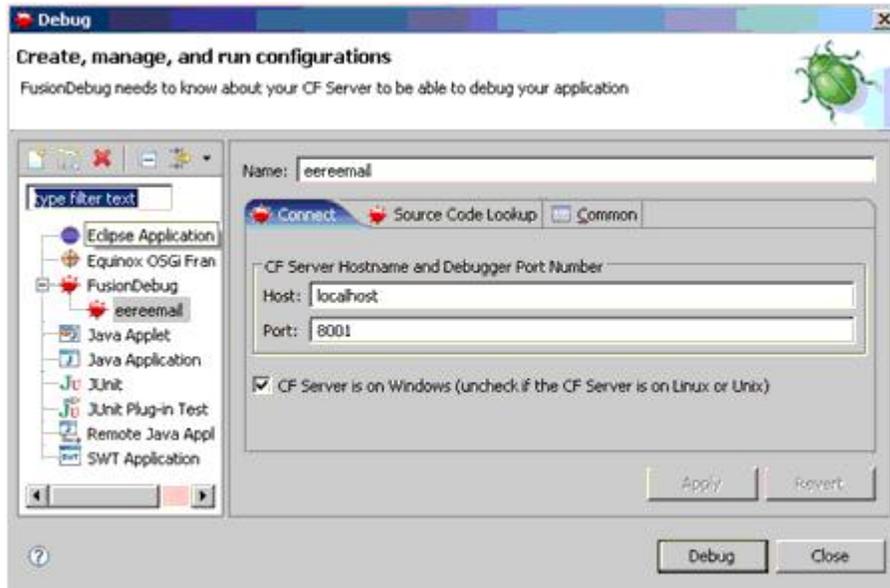
This is my third year in a row attending CFUnited. My name is George Murphy and I am a ColdFusion developer and contractor doing work for the Department of Energy through Technology & Management Services. One of the things that I have come to enjoy most about attending CFUnited each year is being able to discover new software and little snippets of code which I can use throughout the year. My first year was working with ColdFusion developer Selene Bainum to come up with functions to dynamically process form submissions. This year it was discovering the usefulness and value of FusionDebug 2.0.1. This plugin runs inside of CFEclipse and allows the user to set breakpoints and then have FusionDebug step through the code line by line. The coolest aspect of the entire application is being able to see every variable that is being created from when you start running your application until the breakpoint and even after if you step through the code.

- **Installer – Complete IDE Environment**

Another interesting feature of FusionDebug is that it comes bundled with Eclipse 3.2 and CFEclipse 1.3. This is a handy feature if a developer is not running Eclipse on their machine. You don't have to fool with downloading and installing Eclipse separately. However, if you would like to install FusionDebug in your copy of Eclipse that option is available.

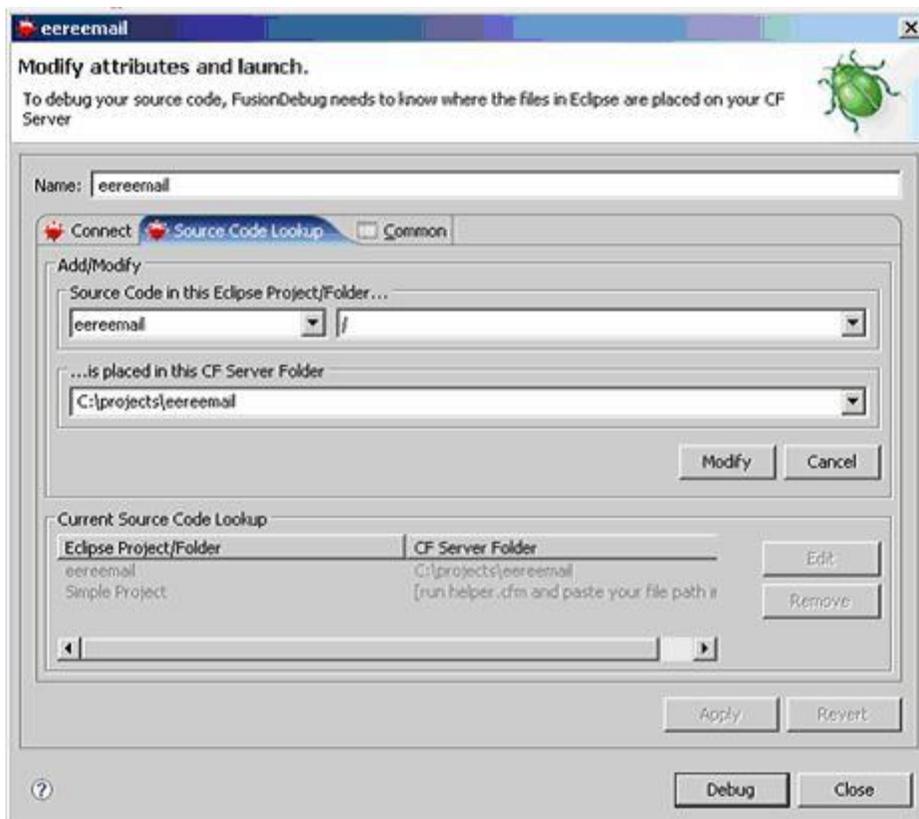
- **Server Configuration Wizard**

There is also a Server Configuration wizard which is capable of updating the ColdFusion /JRun server configuration file in order to allow ColdFusion to accept connections from FusionDebug.



- **Source Code Lookups**

Point to the exact project paths in Eclipse of the code that users need to debug on the ColdFusion server. This is part of the setup process for debugging. If you are having any issues this is the first place where you should look to make sure that your path information for your particular project is correct.



- **Debug Perspective**

This is a window which contains all of the views needed to begin debugging your ColdFusion Applications. The variables tab allows you to see all of the variables that are running in your application. The breakpoint tab lets you see all of the active breakpoints that are set in the application. Finally the expressions tab allows you to monitor and settings that have been applied to any expressions. It is nice to have all of this information in one area.

Name	Value
VARIABLES	
CLEANURLACTIVATE	"http://localhost/eereemail/admin/emailmanage.cfm"
EMAILANTISPAM	ColdFusion Function
EQUIP	ColdFusion Component
EQUIPRETURN	Query (3 Columns, 1 Rows, Current Row: None)
Column Information	
EERE_EQUIP_ID	int identity
EERE_EQUIP_NAME	varchar
EERE_EQUIP_ACTIVE	bit
Rows	
[1]	
EERE_EQUIP_ID	"2"
EERE_EQUIP_NAME	"Printer"
EERE_EQUIP_ACTIVE	"true"
FINDFILE	"action_updatecontractor.cfm"
ISCFUID	ColdFusion Function
ISEMAIL	ColdFusion Function
ISINITIALIZED	"false"
ISSQLINJECT	ColdFusion Function
KEY	"dsn"

## • Inspect Expressions

FusionDebug allows you to create multiple expression watchers. What this does is allow FusionDebug to evaluate those expressions. To simplify it creates a window into your expressions to allow you to see the values going into and coming out of the expressions that you need to know.

The screenshot shows the FusionDebug Expressions window. At the top, there is a tab for 'Expressions' and a search bar containing the expression: `"cleanURLActivate" = "http://localhost/eereemail/admin/emailmanage.cfm"`. Below this, the main window displays the code for `action_updatecontractor.cfm`. The code is as follows:

```

1 <cfset cleanURLActivate = Replace(cgi.HTTP_REFERER, "&Msg=Printer&2Gupdated&2Gsuccessfully&2Espal=1_2", "") >
2 <cfif cgi.HTTP_REFERER EQ application.settings.users.cfcpath10 OR cgi.HTTP_REFERER EQ application.settings
3 <cfset contractor = createObject("component", application.settings.users.cfcpath7).init() />
4 <cfset contractor.setEERE_CID(form.EERE_CID) />
5 <cfset contractor.setEERE_CONTRACTOR_NAME(form.EERE_CONTRACTOR_NAME) />
6 <cfset contractor.setEERE_CORID(form.EERE_CORID) />
7 <cfset contractor.setEERE_ACTIVE(form.EERE_ACTIVE) />

```

Those are just a few of the many features available in this application. To learn more about these features see:

<http://www.fusion-reactor.com/fusiondebug/features.html>

I have never used a debugger before; but after sitting through Charlie Arehart's presentation about FusionDebug and the new debugger that comes bundled inside of ColdFusion 8, I immediately saw the potential value of the two applications.

After the presentation I decided to skip the next presentation and visit the Fusion team to get FusionDebug installed and running on my laptop. The install process was very uneventful which made me quite happy.

I commissioned FusionDebug to solve a nagging problem that I have not been able to solve for the past month. I am in the process of building a LAN E-mail request application for the EERE (Energy Efficiency and Renewable Energy) division of DOE. When users passwords are reset or new users are created I have a component which generates the password and stores it in the session scope. I then read that session scope back to the Administrator who copies it and gives it to the user. When the Administrator leaves the page I clear the session scope of the last variable that I created. I was using the variable `CGI.path_info` to check for the existence of the page I was reading to. And if the `CGI.path_info` was not equal to the page I was reading I cleared that session variable. This worked flawlessly while I was on ColdFusion 6.1. However when I ported the application to ColdFusion 7 or ColdFusion 8 the session variable was never available when I went to read it.

Before I worked with FusionDebug I attempted the classic methods to try and figure out why it was not working. I verified that the session was getting created and that `CGI.path_info` was available and equal to the page I wanted to read to. Right after setting the session variable I did a cflocation back to page I was reading to. I verified that `CGI.path_info` was available right before the redirect and right after. So, it was a mystery as to why the session variable was not available right after the cflocation. Well FusionDebug helped me solve the problem because I was able to see the value of `CGI.path_info` during the cflocation and the value was an empty string. That is why the session scope was getting cleared. For the life of me I could not understand why I did not get this same behavior in ColdFusion 6.1. So, with the discovery of this little bit of information I again turned to FusionDebug to try and help me solve the problem.

There are many different CGI variables which read the same thing. FusionDebug showed me which ones were available during the cflocation and did not change. I decided to use `CGI.script_name` and it is working flawlessly in ColdFusion 6.1, ColdFusion 7, and ColdFusion 8. My point is that I was able to see and correct this problem in 15 minutes. Without FusionDebug I would have chosen another method to correct the problem but not known for sure why I was having this issue. So, I would like to take my hats off to the folks who developed FusionDebug. It is great to have another great application in my tool chest to help me build better and more secure ColdFusion applications. Since CFUnited FusionDebug has become an indispensable development tool for me and TMS is looking at additional purchases for other team members. If you value saving time and working more efficiently then this is the tool for you.

To purchase or learn more about this application see:

<http://www.fusion-reactor.com/fusiondebug/index.html>

