

Jmeter Part 3: Script Recording and Crawling

Posted At : July 28, 2011 12:39 PM | Posted By : Mark Kruger

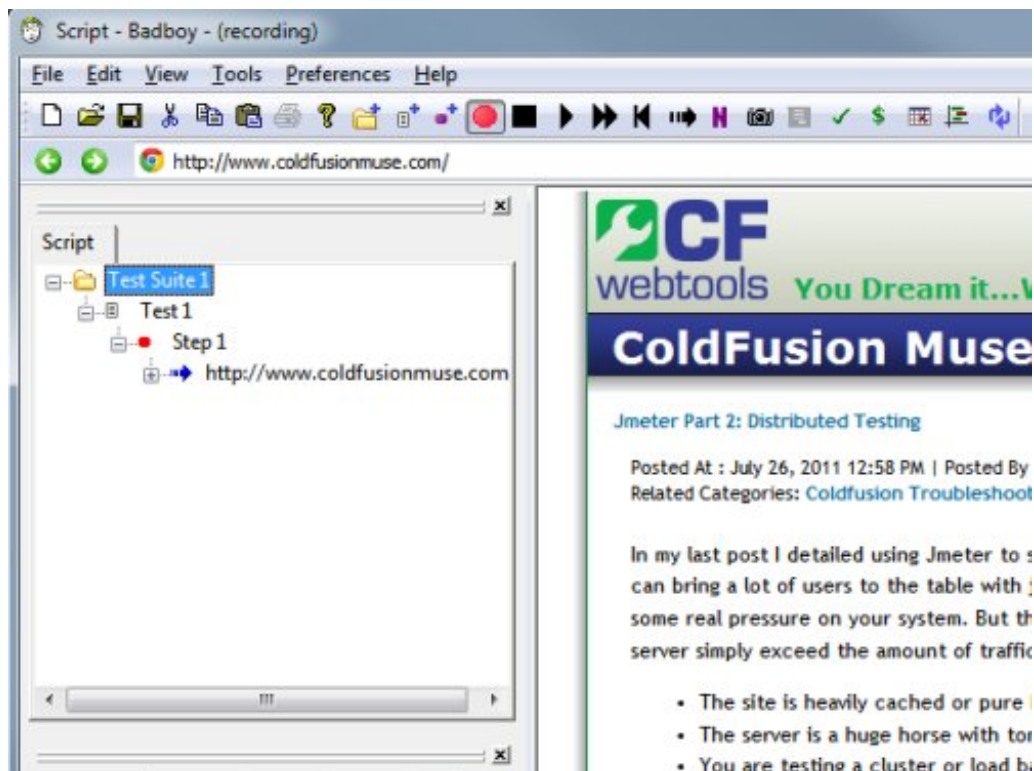
Related Categories: Coldfusion Troubleshooting

This is my last post on the topic of Jmeter testing. I wanted to finish up by showing you 2 things. First, CF Guru Larry Lyons pointed me to a different test tool called "bad boy". This test tool has a good deal of capability in it's own right but the main reason I bring it to your attention is that you can record a test and export it to Jmeter in a few simple steps. Second, as promised in my previous posts ([Jmeter testing](#) and [Jmeter Distributed testing](#)) I'm going to share a Jmeter script that is able to crawl your site using the magic of regular expressions. First, let's explore Bad Boy.

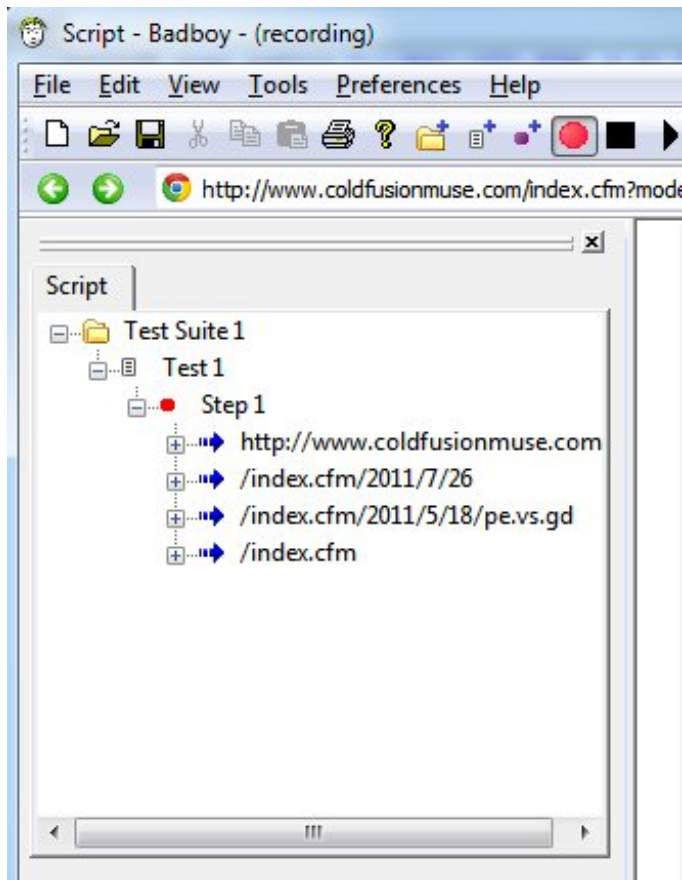
Bad Boy Script Recorder

As far as I can tell, **Bad Boy** is a windows only application. There's a license fee associated with it so if you intend on using it for hard core recording and testing in a professional capacity make sure you read the agreement and pay the fee (\$45 dollars is pretty nominal). Download and install it and then fire it up. One of the first things to note is that BB is not a "proxy" type capture tool. Other macro records are basically web proxies that intercept requests and record the steps you take. That can make them more of a pain to configure. BB on the other hand has its own browser built in. That's handy because it's going to work right out of the box. I'm not sure what code is "under the hood" so perhaps your site won't always function as planned - but you should be able to record basic to fairly complex scripts.

To record, make sure the "record button" (the big red toolbar button) is "on" and type something into the address bar and hit enter. You should see the contents in the view pane and the URL in the test plan like so:

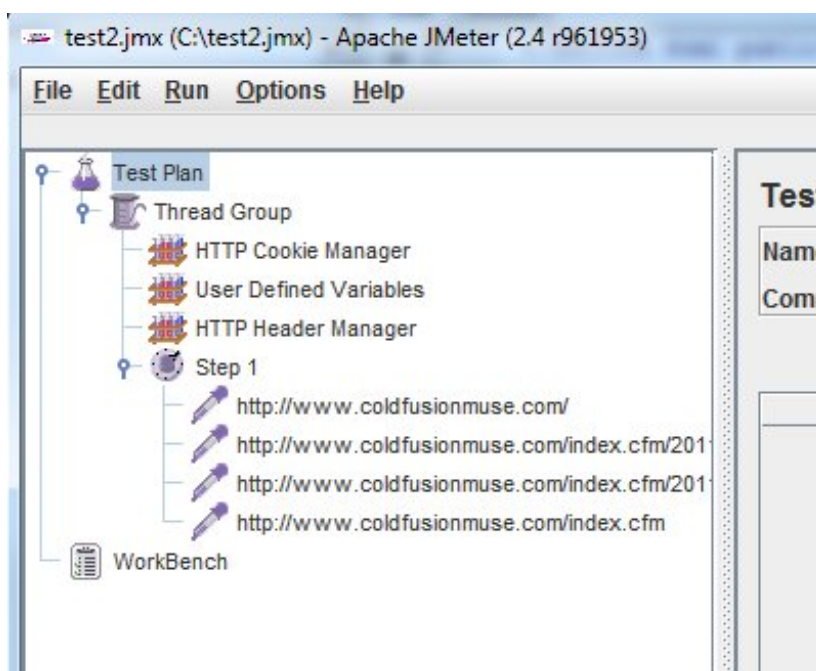


Click around a bit more and you should see additional entries show up in the test plan.



Now simply choose File->Export to Jmeter, and save your test plan as a .JMX file (it's a standard XML file that test plans for Jmeter use).

Open your test plan in Jmeter and you will see something like this:



Jmeter adds a couple items by default that we haven't covered. First it adds the "User Defined Variables" config element. A quick look shows "viewstate" and "JsessionID" as user defined variables that come along with this recorded script. I'm not sure where "viewstate" comes from. Viewstate as a cookie or form variable is a .NET convention so maybe BB uses it and it becomes part of the trapped variables somehow. JsessionID makes sense however since ColdFusion uses it to tie a user to a session. You could specify a value for JsessionID but I suspect that would be more trouble than it's worth since the cookie manager is going to allow you to establish and maintain a session for each thread. Of course you could tie all requests in your test to a single session by adding a value here - and perhaps you might have a good reason for doing that.

The Header Manager config element is useful in that allows you to specify headers that are defaulted with each request. This can be very useful if you want to test code that teases out the browser type and version and takes action based on the browser's capability. This is more common these days with mobile content. By default BB has exported 3 headers for me:

- *User-Agent* - Looks like it is set to IE 8.
- *Accept* - looks like it's set to a long list of possibilities.
- *Accept-Language* set to US English.

HTTP Header Manager

Name: HTTP Header Manager

Comments:

Headers Stored in the Header Manager

Name:	
User-Agent	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64;
Accept	application/x-ms-application, image/jpeg, application/xaml+xml
Accept-Language	en-us

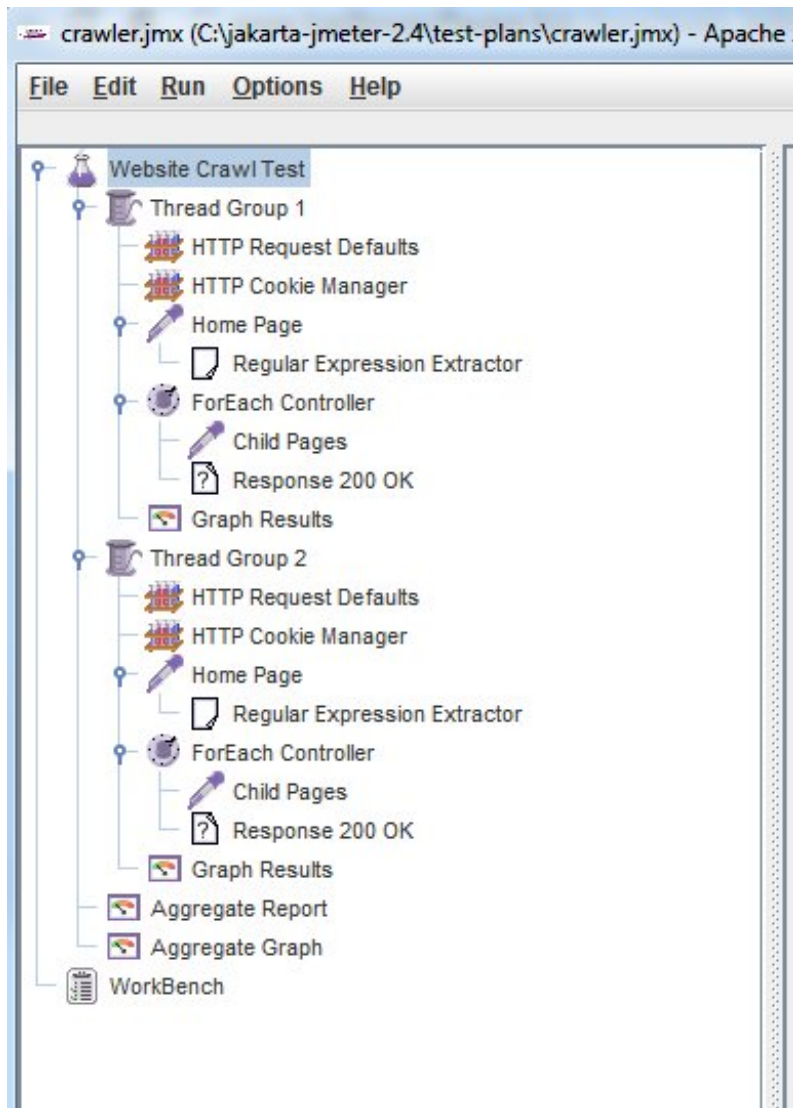
Editing these values is not intuitive. You have to click on the space to edit (either the variable or the value) and the type or hit a key. It doesn't really change to highlight or focus. But once you figure it out it's pretty simple to make a change - like switching to a FF user agent. You can of course add your own headers as you wish.

One of the effective ways to use BB is to "shell out" your test. Record all the steps you want to test - or record multiple tests for multiple thread groups. Once you have the basic JMX file created you can modify it and fine tune it to get exactly the sort of testing protocol you want. It saves you the effort of having to manually add config elements and HTTP samples to your test. I've just started experimenting with it, but I can certainly see it's a huge time saver.

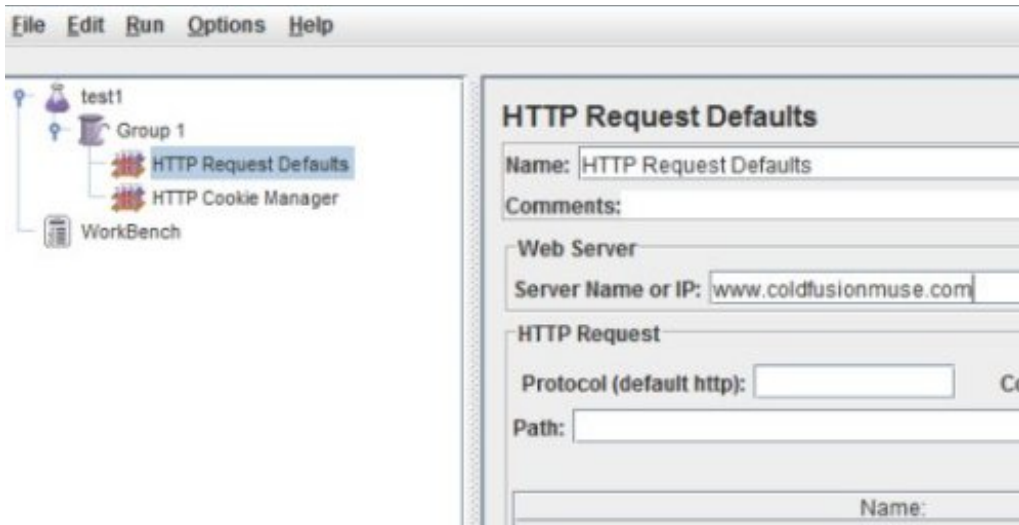
The Crawler Script

Finally, in keeping with my claims on the power of Jmeter I'd like to share a script written by fellow Guru [Wil Genovese](#), who works with me and who I rely on as wealth

of knowledge, inspiration and occasional ribbing (he's also a cigar aficionado and biker). He's one of 3 CFWT developers living and working in Minnesota (we sort of think of the twin cities as our northeast branch). Anyway, Wil shared this **JMX crawler Script**. Go ahead and save it to a .jmx file and open it in the Jmeter Gui. Once open you will see a view like this in the test plan pane:



To use it make sure you do the following *before* you try to run it. First, open the "request defaults" and set the "server name/IP" to the site you want to test. You might remember this from the first time we set up a test:



Make sure and do this for *both thread groups*. Secondly (again for *both* thread groups), go to each "Regular Expression Extractor" and modify the "regular expression" to specify the domain in question as well.

Regular Expression Extractor

Name: Regular Expression Extractor

Comments:

Apply to:

☒ Main sample only ☐ Sub-samples only ☐ Main sample and sub-samples

Response Field to check

☒ Body

Reference Name:	inputVar
Regular Expression:	<a href="(http://www.coldfusionmuse.com[^"]+)"
Template:	\$1\$
Match No. (0 for Random):	-1
Default Value:	fout

These 2 steps allow the script to start with your home page and test each page subsequently linked to on the same site.

Hopefully this illustrates that Jmeter is capable of very complex variablized testing.