

Debugging Flash Remoting Using Service Capture

Posted At : August 31, 2005 10:31 AM | Posted By : Mark Kruger

Related Categories: Flash Remoting

I was introduced to a debugging tool called **Service Capture** by it's author Kevin Langdon. In fact he was reading a previous post of mine on my frustration with the **netconnection debugger** and he added a comment that he had created his "Service Capture" tool to deal with some of that frustration. I tried it out and it has some *excellent* features and advantages.

For one thing it does more than capture "just" your flash remoting calls. Service Capture is a proxy service that captures all of your http traffic and displays the headers for you. The first thing I found was that my QuickBooks 2002 has a TSR running that checks for updates to quickbooks every 10 minutes (ha). Once I started playing with the remoting features it got interesting. I loaded up a small application that serves as a simple search and drill down for sales leads. The interface loads 2 flash movies and makes 3 remoting calls to populate 2 grids and a tabbed interface. Here's what the interface looks like:

The first thing I noticed was that the remoting calls for queries come back into a "recordset" object in the debugging window.

Wow! That sure beats the Netconnection Debuggers raw "*array object*" display. I hate having to expand all those objects to see what the data contains. Now, the columns and rows are all lined up nice and neat in a record set display. Unfortunately, even though all my queries return record sets, not all of the record sets showed up in the SC debugger. I sent a note to Kevin and asked him to clarify how SC determines what to format this way. I have a feeling there will be a fix or explanation for that. In any case during development it is really helpful to be able to examine the data easily.

The second thing that stood out was the left hand pane showing the parameters that were passed as a part of the call. In this case it shows that "parameter 1" was an INT and parameters 2 and 3 where blank.

Again, this is an organized and meaningful way of determining what was sent as a part of the function call. The netconnection Debugger features a raw "array" syntax and a string showing the call to the object. It's hosted in one of Macromedia's tiny little interfaces with tiny text. MM's flash interfaces using text - like the hideous help system - are all made by elves with better than perfect vision. This is much better. It's human readable and shows the information in the same way a programmer might think about it.

SC also traps the raw header information for you to see - both request and response.

Request

```
POST /flashservices/gateway/ HTTP/1.1
Accept: */*
X-Flash-Version: 7,0,19,0
Content-Type: application/x-amf
Content-Length: 235
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Host: scrubber....
Proxy-Connection: Keep-Alive
Pragma: no-cache
Cookie: CFID=106761; CFTOKEN=83514360; JSESSIONID=02303f81870aC$3Fg$B2
```

Response

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 31 Aug 2005 12:33:42 GMT
Content-Type: application/x-amf
Content-Encoding: gzip
Transfer-Encoding: chunked
Expires: Wed, 01 Jan 1997 12:00:00 GMT
Cache-Control: max-age=86400
Vary: Accept-Encoding
```

SC allows you to throttle the bandwidth of a call as well. You can set the speed to Dial up, DSL or Cable speed. The flash IDE has this capability - but this is useful for production. Sometimes, in development, you don't have enough data being passed around to get a real take on how your app performs. This gives you another way of examining that issue.

Some Gotchas

This is a proxy service that intercepts and forwards http calls and handles the results before passing them to the calling object (browser or movie). As such you have to configure your browser to handle proxy services. One important check box in IE is the one under the advanced tab for "Use HTTP 1.1 Through Proxy Connections".

If you do not have this checked, Service Capture will make your remoting calls correctly, but it will be unable to return the results to the calling flash file. Mozilla is even a bit trickier. You will need to add a pointer to the service capture configuration file. The instructions are on the [Service Capture](#) download page.

This is a Java Application and you need to install the Java Runtime (1.4.2+) in order to use it. Developers often already have the latest Java version installed for other reasons - but it's worth mentioning. Resource-wise it will use around 10 megs of physical memory and 30 to 60 of virtual memory. That's not ground-breaking but it is worth keeping in mind if you have resource issues to contend with.

Cost

The cost is 35.00. If you do a lot of remoting work in flash I would wager it is well worth it. If you only do the occasional XML file call or loadVariable call - you will probably not be able to leverage it's power. It does have a free trial so it's worth checking out.