

Augmenting Application.cfm with Filters

Charles Arehart
Founder/CTO, Systemmanage

carehart@systemmanage.com
http://www.systemmanage.com

1/10/2003

Topics

- What's a filter? What's it got to do with Application.cfm?
- Template pre-processing tricks
- Applying them in your environment
- Finding filters
- Template post-processing tricks
- Benefits over CF native approaches

What's a filter?

- Technically: "can inspect or transform the content or headers of an HTTP request or response"
 - Can record information about a request, redirect access to a different page, and more
- Simply: Filters can perform actions before a CF page is processed, or manipulate output after it's processed
 - Can augment traditional application.cfm operations
- They don't typically create output themselves

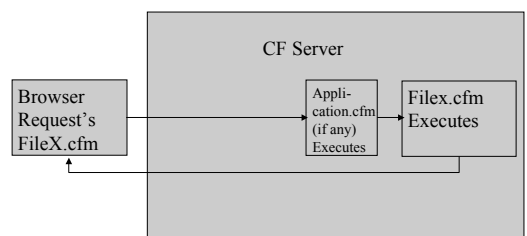
What a filter can do

- Examine request coming into page, then:
 - Invoke the resource (i.e., CF, servlet, JSP, HTML page) in the normal manner
 - Invoke the resource with modified request information
 - Invoke the resource but modify the response before sending it to the client
 - Prevent the resource from being invoked and instead redirect to a different resource, return a particular status code, or generate replacement output
- Separate hi-level access decisions from presentation code

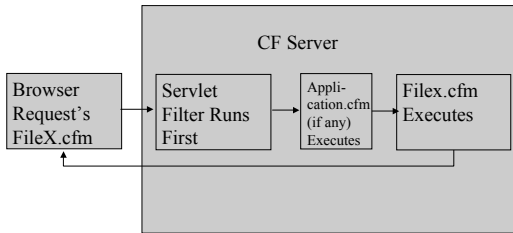
Isn't it a java thing?

- Filters are actually a feature of the Java Servlet API
 - But don't let that scare you!
 - CFMX is built upon a J2EE foundation and includes the full support of the Servlet API
- Don't need to know Java, or servlets, to use filters
 - Don't even need to write filters to get started

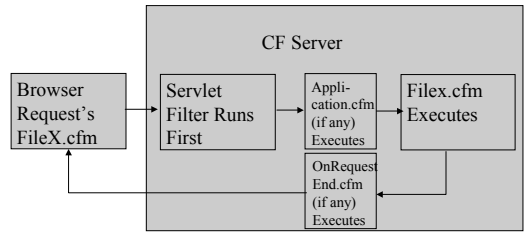
What's it got to do with Application.cfm?



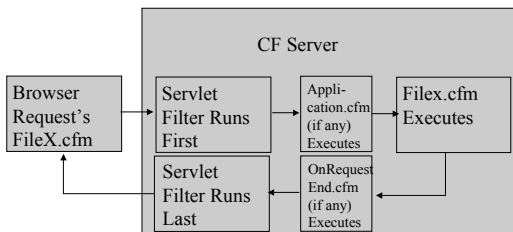
Input (request) filters run before application.cfm



What about OnRequestend.cfm?



Output (response) filters run after OnRequestEnd.cfm



Possible combinations

- Can have either, none, or both of application.cfm/onrequestend.cfm
 - But CF processes onrequestend.cfm only if application.cfm has been processed in same directory
- Can have either, none, or both of input (request) filter/output (response) filter
- Can have any combination of all of them
 - Can actually have multiple filters

Flexible Filters

- Filters are defined by mapping them to a URL pattern in a CFMX XML configuration file
- Can designate filter to be executed for a given URL pattern:
 - All under a given directory/virtual path
 - Implies all subdirs/paths under that
 - In other words, entire application or subset
 - All under root (/)
 - In other words, entire server
 - A specific file

Template pre-processing tricks

- What kinds of things could a request filter do?
 - Logging and auditing visits
 - Blocking execution of a page for a given time period (perhaps during a maintenance period)
 - Blocking visitors from given domains
 - Triggering events when a resource is accessed
 - Caching/reusing a page's results
 - Modifying the input request before processing by CF
 - Modifying request character encoding
 - To name a few



What other uses of filters?

- Evaluate request to detect
 - Whether referer is expected one
 - Whether expected form fields are present
 - Whether expected cookies, authentication, etc. are present
 - Whether incoming form data has HTML (to filter it out to prevent cross-browser scripting)
 - So many more possibilities



The catch? They're written in Java

- Filters are a servlets feature
 - Servlets are written in Java
- No point looking at them since most here won't know java
 - Will point you to resources for learning more later if you want to write your own
- But you can find many examples already written
 - Only need to install and configure them



Finding filters

- www.servletsuite.com/servlets.htm
- JSPTags.com
- See resources for learning more at end of presentation



Some examples

- Cache filter
 - <http://www.servletsuite.com/servlets/cacheft.htm>
- Billing Filter
 - <http://www.servletsuite.com/servlets/billflt.htm>
- Profiler Filter (track execution times)
 - <http://www.servletsuite.com/servlets/proffit.htm>
- IP Filter (access restriction, load balancing)
 - <http://www.servletsuite.com/servlets/ipflt.htm>
- Will see zip filter later



Applying them in your environment

- Just 4 simple steps
 1. Find or build a filter
 2. Place java file for filter in CFMX java file dir
 3. Modify CFusionMX\wwwroot\WEB-INF\web.xml to define filter and its mapping
 4. Restart CFMX server



Finding one sample filter

- Let's use the timerfilter provided in a java article explaining filters (as listed at the end in the resources section)
 - http://javaboutique.internet.com/tutorials/Servlet_Filters/Filters-Demo-Source.zip
- We want the timerfilter in that zip file.



Placing the java file into CFMX

- Place java file for filter in CFMX java file dir
 - For .java or .class file, CFusionMX\wwwroot\WEB-INF\classes
 - For .jar file, CFusionMX\wwwroot\WEB-INF\lib
- Note: if downloaded file is a "War" (web archive) file, that's just a zip file that can be opened with WinZip, etc.



Understanding Java Packages

- May need to understand issues of Java packages, to know where to place java files
 - Can typically just place jar in lib directory
 - It holds files in an appropriate internal directory structure
 - For source (.java) files, need to pay attention to possible "packaging"
 - If filter source code shows use of package as in:

```
package com.kieff.FilterDemo;
```

Then it expects to be stored in com\kieff\FilterDemo directory under some directory in classpath
 - Create that under that classes directory on prev slide



Hidden Gem: No need to compile the filters

- CF can auto compile servlets (and therefore filters)
 - Not suitable for production, as it's a slight performance detriment to watch for changes
- In [cfmx_home]\wwwroot\WEB-INF\jrun-web.xml
 - add within the existing <jrun-web-app> element:

```
<reload>true</reload>
<compile>true</compile>
```
 - Restart the server
- CFMX will now reload classes from [cfmx_home]\wwwroot\WEB-INF\classes and lib
- See blog entry:
 - http://cfmxplus.blogspot.com/2002_08_04_cfmplus_archive.html#85325044



Modifying web.xml to map filter

- Register the filter for the desired pages
 - Use filter and filter-mapping in web.xml
- File already exists in CFusionmx\wwwroot\WEB-INF
 - Add needed xml elements within <web-app> root element



Example

```
<filter>
  <filter-name>Timer</filter-name>
  <filter-class>com.kieff.FilterDemo.TimerFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>Timer</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```



Filter mapping options

- Several possibilities
 - /* means all templates served by server
 - Can do a subpath such as "/secure/*"
 - Or all files of a type, like "*.cfm"
 - Can name a specific file, like "/filtered_page.html"
 - Or map to a named servlet as well
 - Same as for <servlet-mapping> block, for those familiar with it



Ordering XML elements in web.xml

- Have read that:
 - filter, filter-mapping, servlet, and servlet-mapping elements must appear in the web application deployment descriptor in that order
- But have seen it work in CFMX even if not in that strict order



JRun's order of processing

- JRun orders the execution of filters for each request depending on the URL patterns and servlets that the request URI matches in the web.xml file
- JRun considers url-pattern matches first
 - in the order that they appear in the web.xml file
- followed by servlet-name matches
 - in the order that they appear in the web.xml file



More flexibility in mapping

- Can designate multiple URL patterns for a filter
- Can designate multiple filters to a given URL pattern
- Filters can be added without application or filter knowing about each other
- Can modify filter without modifying application
 - And can apply that change across several apps



Restart and run

- Restart server
- Execute page that's mapped to filter to test
- This filter was set to run for all templates
 - Also writes its output "System.out.println"
 - commonly used in java apps to write to "console"
 - No such "console"
 - So where does CF write such output? ...



Where System.out.println writes in CFMX

- If filter (or servlet) writes to System.out.println, where does it go in CFMX?
 - CFusionMX\runtime\logs\default-out.log
- Writes in strange order
 - Descending by date
 - (latest date's output at top of file)
 - But within that date, ascending by time



Doesn't work?

- Are you sure you saved all files?
- Correct mappings in web.xml?
- Correct location of java files in classes directory?
 - And in subdirectories if filter uses a package
- Set the option for CF to auto-compile java files?
- Restarted the server?

Template post-processing tricks

- What kinds of things can a response filter do?
 - Logging page execution time
 - Localization (targeting output to a locale)
 - Image conversion
 - XML transformations of XML output
 - Encryption
 - Data compression

Examples

- Again, see previous page listing sites with examples
- Let's focus on compression filter
 - <http://www.servletsuite.com/servlets/gzipflt.htm>
 - Place the gzipflt.jar in the cfusionmx\wwwroot\WEB-INF\lib directory

Compression Filter

- Basically, it zips output of pages to compress them
 - CF pages typically have lots of white space
 - All web pages benefit from some compression
 - Most browser support compression
- A good compression filter:
 - Only compresses if browser will support it
 - Browsers send accept-encoding header indicating if they support compression
 - Can view that in CF with `cgi.http_accept_encoding`
 - Doesn't bother if the request output isn't large enough to benefit from compression
- Haven't explored this one to see how it works

Web.xml configuration

```
<filter>
  <filter-name>GZIPFilter</filter-name>
  <filter-
class>com.cj.gzipflt.GzipFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>GZIPFilter</filter-name>
  <url-pattern>*.cfm</url-pattern>
</filter-mapping>
```

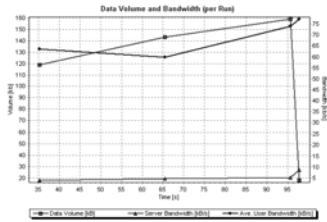
Implementing and testing

- Follow previous steps to implement
- Use load testing tool to demonstrate effect
- Several exist
 - One I like for local testing is paessler.com
 - MS has free tool, not as effective
 - OpenDemand.com has cool web-based tool, but not suitable without net connection
 - Companies like Empirix, Segue, and Mercury Interactive have very expensive (and powerful) alternatives

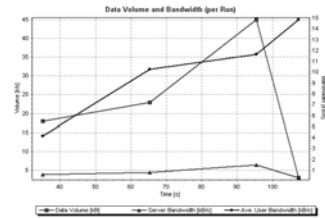
Paessler WebServer Stress Tool

- Using their release 6 beta
 - <http://www.paessler.com/products/webstress/wbstressbeta.htm>
- Pretty cheap too prices range from \$50 to 999
 - See <http://www.paessler.com/order>

Results without compression



Results with compression: 4:1 reduction in bytes sent to browser!



Running tests of compression

- Be aware, when testing this way, or with CFHTTP, that you need to send a header to the server (as a browser would) indicating that you accept Gzip compression
 - See `cgi.http_accept_encoding`
 - Set with `accept-encoding:gzip,deflate` in load testing tool
 - Or `<cfhttpparam type="CGI" name="Accept-Encoding" value="gzip,deflate">` on CFHTTP
- See my blog: http://cfmxplus.blogspot.com/2002_11_24_cfmplus_archive.html#85712370

Key difference between filters and OnRequestend

- OnRequestEnd processing does not occur if CFABORT is executed
 - Makes it unreliable for some end of request processing
- Response filters, if designated, will always be executed regardless of CF processing
- Response filters apply to ALL files matching the given URL: CF, JSP (enterprise), servlets, and even static files (HTML) and graphics!
 - Flash as well, if there's some benefit to doing so

Benefits of filters over CF native approaches

- Can apply to more than just application
 - A file, or entire server
- Don't need to modify multiple `application.cfm`'s to effect behavior across several applications
- Can change filter without changing CF template(s)

More about filters

- There are more topics worth exploring
 - Accessing the servlet context in filter
 - Using initialization parameters (to modify filter behavior at deployment time)
 - Modifying the request scope to pass to template
 - Chaining of multiple filters
 - Disabling the invoker servlet
- Equally compelling are event listeners
 - Can track/respond to session creation/modification/expiration

Learning more about filters

- Beware: tend to be oriented toward java developers
- Sun article, "The Essentials of Filters"
 - <http://java.sun.com/products/servlet/Filters.html>
 - Samples at <http://java.sun.com/products/servlet/filters.zip>
- O'reilly article, "Writing Servlet 2.3 Filters"
 - <http://www.onjava.com/lpt/a/826>

Learning more about filters

- Java Boutique's "Writing Servlet Filters"
 - http://javaboutique.internet.com/tutorials/Servlet_Filters/
 - Code: http://javaboutique.internet.com/tutorials/Servlet_Filters/Servlet-Filters-Demo-Source.zip
- JRun 4 Programmer's Guide, Chapter 7
 - http://livedocs.macromedia.com/jrun4docs/Programmers_Guide/filters.jsp

Learning more about filters

- Marty Hall's More Servlets and JSP
 - Chapter 9 "Servlet and JSP Filters"
 - Available online in PDF form
 - http://developer.java.sun.com/developer/Books/javaserverpages/servlets_javaserver/servlets_java_server09.pdf
- My blog (cfmxplus.blogspot.com)
 - Fun with Filters and Listeners
 - http://cfmxplus.blogspot.com/2002_09_01_cfmplus_archive.html#85411326

Summary

- Filters are powerful
 - And relatively easy to use
- Don't need to know java
 - If you can find existing filters
- Just need to know about where to put filters, how to setup web.xml file
- They offer far more power than application.cfm and onrequestend.cfm
- Hope you enjoyed this introduction!

Excerpted from One Day Seminar

- This is one section of a day-long seminar on Java/CFMX integration
 - "Understanding Java/J2EE Integration in CFMX, For Those With No Java Experience"
 - More at <http://www.systemmanage.com/seminars/>
- Offered to public at \$249 per attendees
 - Available onsite at \$2499, flat fee for all attendees
- Seminars are lecture format, like this talk
 - No need for computers
 - Hundreds of slides like these details provided

Contact Information

- Happy to answer questions related to talk
 - Email: carehart@systemmanage.com
 - Phone: 301 604 8399
 - Web: www.systemmanage.com
- Also available for
 - Training (custom or pre-written)
 - CF, DB, JRun/J2EE, Javascript, wireless, and more
 - Consulting (very short-term)
 - best practices, architecture, setup
 - Developer Group Mentoring