

Investigation on auto-configuring Apache vhosts in Tomcat for use with Railo

Paul Klinkenberg, september 24, 2010 (paul@ongevraagdadvies.nl)

Contents

Investigation on auto-configuring Apache vhosts in Tomcat for use with Railo.....	1
Option 1: 1 tomcat Host with multiple contexts, using a RewriteRule.....	1
Pros:.....	2
Cons:.....	2
Option 2: Hosts added to tomcat's server.xml, using ProxyPassMatch.....	3
Pros.....	3
Cons.....	3
Remarks and questions.....	4
Same hostname, but different port and webroot.....	4
Can Hosts be added to tomcat without restarting?.....	5

To start off, I haven't found a perfect solution yet. All options mentioned here have their pros and cons. If you can think of a better way, then please let me know!

What I wanted to do, is create a cfml script which checks the Apache configuration files, extracts all virtualhosts and the default hosts, and adds them to the tomcat configuration, without any necessary user interaction. That way, a web admin only needs to add his virtual host in one place: his frontend webserver.

Background for trying to accomplish this, is the problematic installation/configuration that users (and myself in the beginning) often encounter when trying out Railo.

Reason for using tomcat is that Railo will be distributed with tomcat by default.
IIS 6 and 7 will be handled later on; this document purely focuses on Apache 2.2.

Environment:

- Apache 2.2.9+ webserver, used as the frontend webserver, with zero to many VirtualHosts.
- Tomcat 6.0 with Railo

Option 1: 1 tomcat Host with multiple contexts, using a RewriteRule

As Sean Corfield [pointed out](#), you can configure everything as follows.

Tomcat server.xml:

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
  [... some Listener directives ...]
  <GlobalNamingResources>[...]</GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" />
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>

      <Host name="localhost"/>
```

```

    </Engine>
  </Service>
</Server>

```

Apache's httpd.conf + optional extra included conf files:

```

[... lot of config here ...]
# necessary for proxying to Tomcat:
<IfModule !proxy_module>
    LoadModule proxy_module libexec/apache2/mod_proxy.so
</IfModule>
<IfModule !rewrite_module>
    LoadModule rewrite_module libexec/apache2/mod_rewrite.so
</IfModule>

ProxyPreserveHost On
RewriteEngine On
## All urls ending in .cfm/cfc, and urls containing '.cfm/' and '.cfc/', are
passed to Tomcat-Railo:
RewriteRule ^/(.*\.cf[cm])$ ajp://localhost:8009/{HTTP_HOST}/$1 [P]
RewriteRule ^/(.*\.cf[cm])/(.*)$ ajp://localhost:8009/{HTTP_HOST}/$1?
path_info=$2 [P,QSA]
## Redirect all flash/flex/amf calls to Tomcat-Railo: WILL NOT WORK
RewriteRule ^/((flashservices/gateway|messagebroker/|flex2gateway/|
openamf/gateway/).*) ajp://localhost:8009/{HTTP_HOST}/$1 [P,NC]

<VirtualHost *:80>
    DocumentRoot /developing/test.local/data/
    ServerName www.test.local
    # Vhosts by default do not inherit the main rewriteRules, so:
    RewriteEngine On
    RewriteOptions Inherit
</VirtualHost>
[... more Vhosts and/or settings ...]

```

For every hostname found in Apache's configuration, we need to add a file in tomcat's configuration, which will be picked up by tomcat almost instantaneous:

`{tomcat-install-dir}/conf/Catalina/localhost/{hostname}.xml`

These xml files will contain the following:

```

<?xml version="1.0" encoding="UTF-8"?>
<Context docBase="{VHost document-root}" />

```

So for the site `www.test.local`, we will add a file

`/Applications/tomcat/conf/Catalina/localhost/www.test.local.xml`, with the contents:

```

<?xml version="1.0" encoding="UTF-8"?>
<Context docBase="/developing/test.local/data/" />

```

Pros:

1. Tomcat does not need to be restarted for changes to take effect.
2. In Apache, cfm usage can be enabled/disabled per virtualhost. (note: if an `.htaccess` file was placed in the webroot of such a non-cfm site with the same 2 config lines, then cfm would still be enabled).

Cons:

1. In Apache, the web admin needs to add 2 lines within every `<VirtualHost>`
2. The `cgi.script_name` environment variable in Railo will be 'wrong'. When calling `/index.cfm` on `www.test.local`, the `cgi.script_name` will be `/www.test.local/index.cfm`.
3. Amf (Flash/flex remoting) will not work, because the servlet listens to `"/flashservices/..."`,

not “/www.test.local/flashservices/...”.

Option 2: Hosts added to tomcat's server.xml, using ProxyPassMatch

This is more or less a default configuration, and tomcat needs to be restarted to pick up the changes.

Tomcat server.xml:

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
  [... some Listener directives ...]
  <GlobalNamingResources>[...]</GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" />
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>

      <Host name="www.test.local">
        <Context docBase="/developing/test.local/data/" />
      </Host>
      <!-- ... more vhosts ... -->
    </Engine>
  </Service>
</Server>
```

Apache's httpd.conf + optional extra included conf files:

```
[... lot of config here ...]
# necessary for proxying to Tomcat:
<IfModule !proxy_module>
  LoadModule proxy_module libexec/apache2/mod_proxy.so
</IfModule>
<IfModule !rewrite_module>
  LoadModule rewrite_module libexec/apache2/mod_rewrite.so
</IfModule>

ProxyPreserveHost On
ProxyPassMatch ^/(.+\.cf[cm]($|/.*))?$ ajp://localhost:8009/$1?path_info=$2
ProxyPassMatch ^/((flashservices/gateway|messagebroker/|flex2gateway/|
openamf/gateway/).*) ajp://localhost:8009/$1

<VirtualHost *:80>
  DocumentRoot /developing/test.local/data/
  ServerName www.test.local
</VirtualHost>
[... more Vhosts and/or settings ...]
```

As you can see, there is no extra config necessary in the <VirtualHost> tags of Apache, which is great. The only really big downturn is, that tomcat needs to be restarted for the new <Host> settings to take effect.

Pros

1. No per-host settings necessary in Apache
2. Everything (except the SES urls/cgi.path_info) works like expected

Cons

1. Tomcat needs to be restarted on every config change. Since I am trying to create a cfml

- script which handles the config changes for tomcat, while it is running on tomcat, I am pessimistic about the option of restarting tomcat.
2. Also, this restarting would mean that a live server would become unresponsive for a few seconds at least,
 3. and sessions in Railo would be cleared while restarting.

Remarks and questions

Same hostname, but different port and webroot

In Apache, it is possible to have the same hostname listening to different ports, where it will serve from different webroots. For example:

```
<VirtualHost *:80>
    DocumentRoot /developing/test.local/data/
    ServerName www.test.local
</VirtualHost>
<VirtualHost *:81>
    DocumentRoot /developing/oldsite.test.local/data/
    ServerName www.test.local
</VirtualHost>
```

How can we distinguish between these 2 websites in tomcat? The <Host> tag does not have a *port* parameter, because it is by default only listening to one port.

With Option 1 described above, we could change the RewriteRule to something like:

```
RewriteRule ^/(.*\.cf[cm])$ ajp://localhost:8009/{HTTP_HOST}%{SERVER_PORT}/$1
[P]
```

Then we would need to change the name of the tomcat xml files accordingly: from “www.test.local.xml” to “[www.test.local80.xml](#)”.

If a port is not explicitly defined in Apache, by using

```
<VirtualHost *:*>
    DocumentRoot /developing/test.local/data/
    ServerName www.test.local
</VirtualHost>
```

then I still have an escape. Since I gather all listening ports from the httpd.conf (defined as “Listen 80” and “Listen 443” etc.), I would just add tomcat xml files for all those ports, for this host.

With option 2, it is a bit more complicated. Since tomcat is then only looking at the host name, I would need to start extra instances of tomcat for every “same hostname, but different port and different webroot”. Also, I would need to change the ProxyPassMatch directives to use the right ajp listener port (i.e. ajp://localhost:8009/).

Now, because there are limits to what an automated script can and should do, I would just limit the amount of tomcat instances to 2, where all sites on port 80 will go to instance 1, and all other requests to instance 2. That way, we have the most obvious Apache config covered, where an ssl site has the same (or no defined) hostname, but another webroot:

```
<VirtualHost *:80>
    DocumentRoot /developing/test.local/data/
    ServerName www.test.local
</VirtualHost>
<VirtualHost *:443>
    DocumentRoot /developing/members.test.local/data/
    ServerName www.test.local
</VirtualHost>
```

Can Hosts be added to tomcat without restarting?

I have seen the tomcat documentation ([link](#): paragraph 6) which says that you can also add contexts (not Hosts) by adding new directories and an xml file like this: `$CATALINA_BASE/conf/[enginename]/[hostname]/context.xml.default`. I have tried to create new Hosts that way, but to no avail. Even after a restart, tomcat did not use the directory as a new Host. This [article about tomcat reloading/restarting](#) gives a lot of reload options, but none on how to load new Hosts.

And as a last remark: only now I am starting to see the beauty of Resin ;-)