

Apache 与 Tomcat 整合手册

（无集群版）

作者：Steel.Ma

Mail:	steel.ma@139.com
	steel.ma@tom.com
MSN:	Motorola_8088@hotmail.com
Blog:	http://hi.csdn.net/iamsteelma
	http://blog.sina.com.cn/steelma

一、环境准备说明

首先需下载相关的软件：

- 1、JDK 1.5 以上（笔者用的是 1.5.04）
- 2、Apache 2.2 以上（笔者用的是 2.2.11）
- 3、Tomcat 6.0 以上（笔者用的是 6.0.14）

相关的安装配置比较简单，全部采用默认即可，本处略过。

二、Apache 配置

- 1、打开相关的转发代理 so 的配置

在 2.2 版本以前，如果需要整合 Apache 与 Tomcat 的话，比如自己下载代理的 so 文件，并进行复杂的配置，2.2 版本以后，Apache 中已经内置了，所以只需将在 conf/httpd.conf 文件中，将相关被注释掉的配置语句释放即可，需要释放的配置如下：

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

修改完毕后应如下：

```
118 #LoadModule spelling_module modules/mod_spelling.so
119 #LoadModule ssl_module modules/mod_ssl.so
120 #LoadModule status_module modules/mod_status.so
121 #LoadModule substitute_module modules/mod_substitute.so
122 #LoadModule unique_id_module modules/mod_unique_id.so
123 #LoadModule userdir_module modules/mod_userdir.so
124 #LoadModule usertrack_module modules/mod_usertrack.so
125 #LoadModule version_module modules/mod_version.so
126 #LoadModule vhost_alias_module modules/mod_vhost_alias.so
127
128 LoadModule proxy_module modules/mod_proxy.so
129 LoadModule proxy_connect_module modules/mod_proxy_connect.so
130 LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
131 LoadModule proxy_http_module modules/mod_proxy_http.so
132 LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
133 LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
134
135
136 <IfModule !mpm_netware_module>
137 <IfModule !mpm_winnt_module>
138 #
139 # If you wish httpd to run as a different user or group, you must run
```

2、加入转发代理地址配置

在 conf/httpd.conf 文件最下面加入如下配置：

ProxyRequests Off

<proxy balancer://cluster>

BalancerMember ajp://127.0.0.1:8009 loadfactor=1 route=jvml

</proxy>

需要说明的是，Apache 在与 Tomcat 整合时允许进行集群配置（cluster），但本处不进行集群配置，因此只需按照如上所述进行配置，效果如下：

```
485 # but a statically compiled-in mod_ssl.
486 #
487
488
489 <IfModule ssl_module>
490 SSLRandomSeed startup builtin
491 SSLRandomSeed connect builtin
492 </IfModule>
493
494 ProxyRequests Off
495 <proxy balancer://cluster>
496 BalancerMember ajp://127.0.0.1:8009 loadfactor=1 route=jvml
497 </proxy>
498 |
```

3、打开虚拟服务器的配置

在 conf/httpd.conf 文件找到这句语句：

```
#Include conf/extra/httpd-vhosts.conf
```

将其注释释放，效果如下：

```
462 # User home directories
463 #Include conf/extra/httpd-userdir.conf
464
465 # Real-time info on requests and configuration
466 #Include conf/extra/httpd-info.conf
467
468 # Virtual hosts
469 Include conf/extra/httpd-vhosts.conf
470
471 # Local access to the Apache HTTP Server Manual
472 #Include conf/extra/httpd-manual.conf
473
474 # Distributed authoring and versioning (mod_dav)
```

相关的虚拟服务器的配置将在 conf/extra/httpd-vhosts.conf 文件中进行设置。

4、加入 jsp 文件支持

在 conf/httpd.conf 文件找到这句语句

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

将其改为如下：

```
<IfModule dir_module>
    DirectoryIndex index.html index.jsp
</IfModule>
```

效果如下：

```
241
242 #
243 # DirectoryIndex: sets the file that Apache will serve
244 # is requested.
245 #
246 <IfModule dir_module>
247     DirectoryIndex index.html index.jsp
248 </IfModule>
249
250 #
251 # The following lines prevent .htaccess and .htpasswd
252 # viewed by Web clients.
253 #
254 <FilesMatch "\.ht">
```

5、修改 conf/extra/httpd-vhosts.conf 文件，配置虚拟服务器

在文件最后加入如下配置：

```
<VirtualHost *:80>

ServerAdmin webmaster@dummy-host3.steel.com

ServerName localhost

ServerAlias localhost

ProxyPass / balancer://cluster/ stickysession=jsessionid nofailover=On
ProxyPassReverse / balancer://cluster/

</VirtualHost>
```

效果如下：

```
32     ErrorLog logs/dummy-host.steel.com-error.log
33     CustomLog "logs/dummy-host.steel.com-access.log" common
34 </VirtualHost>
35
36 <VirtualHost *:80>
37     ServerAdmin webmaster@dummy-host2.steel.com
38     DocumentRoot "D:/apache/docs/dummy-host2.steel.com"
39     ServerName dummy-host2.steel.com
40     ErrorLog "logs/dummy-host2.steel.com-error.log"
41     CustomLog "logs/dummy-host2.steel.com-access.log" common
42 </VirtualHost>
43
44 <VirtualHost *:80>
45     ServerAdmin webmaster@dummy-host3.steel.com
46     ServerName localhost
47     ServerAlias localhost
48     ProxyPass / balancer://cluster/ stickysession=jsessionid nofailover=On
49     ProxyPassReverse / balancer://cluster/
50 </VirtualHost>
51
52
53
```

以上，Apache 端的配置全部完成

三、Tomcat 配置

Tomcat 端只需修改 tomcat/conf/server.conf 即可

1、修改 Engine 配置

找到下面这句话：

```
<Engine name="Catalina" defaultHost="localhost">
```

将其修改为如下：

```
<Engine name="Standalone" defaultHost="localhost" jvmRoute="jvm1">
```

效果如下：

```

90
91     <!-- An Engine represents the entry point (within Catalina) that processes
92          every request.  The Engine implementation for Tomcat stand alone
93          analyzes the HTTP headers included with the request, and passes them
94          on to the appropriate Host (virtual host).
95          Documentation at /docs/config/engine.html -->
96
97     <Engine name="Standalone" defaultHost="localhost" jvmRoute="jvm1">
98     <!--
99     <Engine name="Catalina" defaultHost="localhost">
100 -->
101
102     <!--For clustering, please take a look at documentation at:
103          /docs/cluster-howto.html  (simple how to)
104          /docs/config/cluster.html (reference documentation) -->
105     <!--
106     <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster">

```

2、修改/确认 Connector 配置

检查文件中对 Connector 端口 8009 配置是否如下，若不是，请修改之。

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

```

83         maxThreads=150 scheme=https secure=true
84         clientAuth="false" sslProtocol="TLS" />
85     -->
86
87     <!-- Define an AJP 1.3 Connector on port 8009 -->
88     <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
89
90
91     <!-- An Engine represents the entry point (within Catalina) that processes

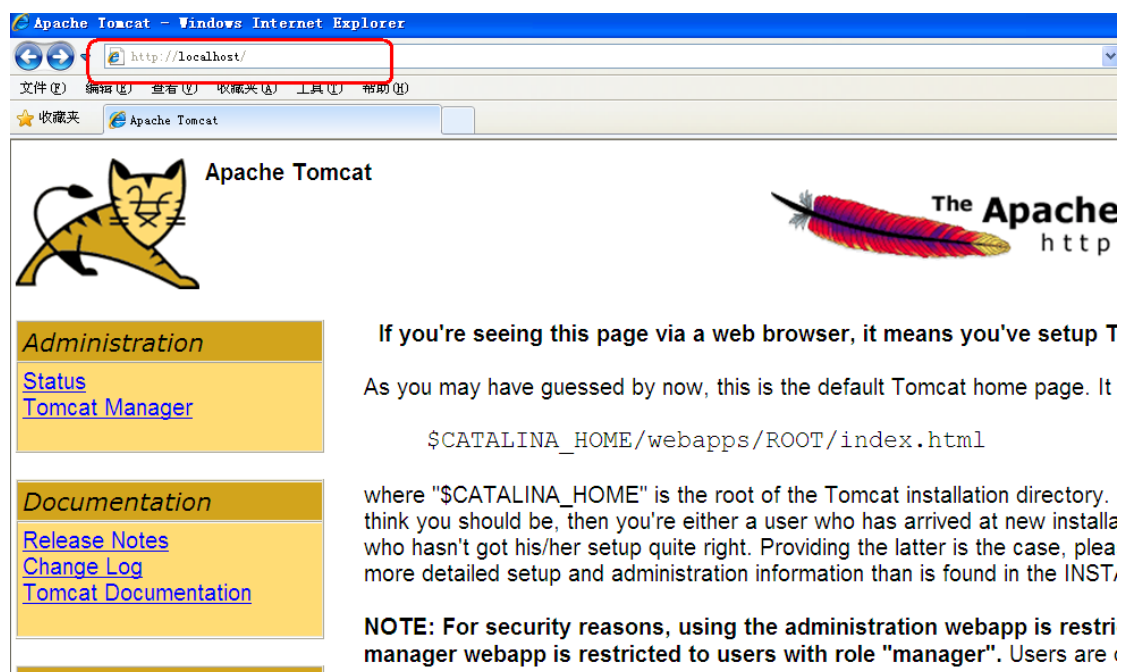
```

以上，Tomcat 端配置全部完成。

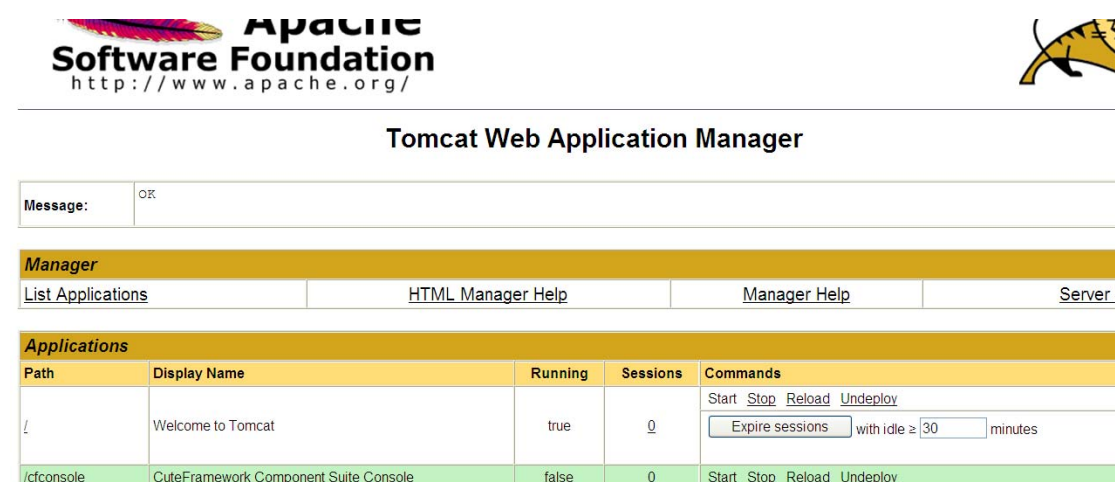
四、测试

1、启动 Apache 和 Tomcat

2、直接访问 <http://localhost>，由于之前将 `index.jsp` 也配置为默认文件名，因此浏览器将出现熟悉的 Tomcat 画面，里面的功能全部能运行，表明配置成功。



例如我们进入 Tomcat Manager，登陆后，即可出现管理界面：



3、进一步测试

我们做一个最简单的 jsp 文件，只在浏览器上显示 1 行文字，代码如下：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> New Document </TITLE>
```

```
<META NAME="Generator" CONTENT="EditPlus">
```

```
<META NAME="Author" CONTENT="">
```

```
<META NAME="Keywords" CONTENT="">
<META NAME="Description" CONTENT="">
</HEAD>

<BODY>
test !

</BODY>
</HTML>
```

将其保存到 tomcat/webapps/ROOT，文件名为 ttt.jsp

然后我们在浏览器中输入：<http://localhost/ttt.jsp>，整合后的系统便将这个新的 jsp 文件编译并运行，效果如下：

