# 1 tomcat 文件配置详细说明

tomcat服务器需配置三个端口才能启动，安装时默认启用了这三个端口，当要运行多个tomcat服务时需要修改这三个端口，不能相同。端口配置路径为tomcat\ conf\service.xml

* 修改Shutdown端口（默认为8005端口）

<Server port="8005" shutdown="SHUTDOWN">

* 1
* 修改http访问端口（默认为8080端口）   
  负责建立HTTP连接。在通过浏览器访问Tomcat服务器的Web应用时，使用的就是这个连接器。

<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

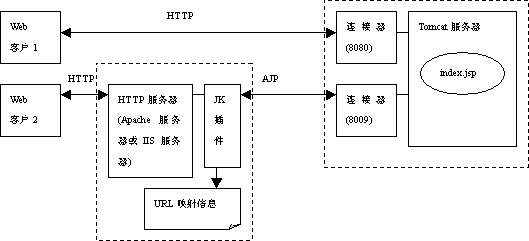
<!-- A "Connector" using the shared thread pool-->

<!-- <Connector executor="tomcatThreadPool" port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

* 1
* 2
* 3
* AJP协议访问端口（默认为8009端口）   
  负责和其他的HTTP服务器建立连接。在把Tomcat与其他HTTP服务器集成时，就需要用到这个连接器。

<!-- Define an AJP 1.3 Connector on port 8009 -->

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

* 1
* 2
* Web客户访问Tomcat服务器上JSP组件的两种方式如图所示。   
  

配置文件$CATALINA\_HOME/conf/server.xml的说明该文件描述了如何启动Tomcat Server

<!----------------------------------------------------------------------------------------------->

<!-- 启动Server

\*\*\*在端口8005处等待关闭命令\*\*\*

如果接受到"SHUTDOWN"字符串则关闭服务器

-->

<Server port="8005" shutdown="SHUTDOWN" debug="0">

<!-- Listener ???

目前没有看到这里

-->

<Listener className="org.apache.catalina.mbeans.ServerLifecycleListener" debug="0"/>

<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" debug="0"/>

<!-- Global JNDI resources ???

目前没有看到这里，先略去

-->

<GlobalNamingResources>

... ... ... ...

</GlobalNamingResources>

<!-- Tomcat的Standalone Service

Service是一组Connector的集合

它们共用一个Engine来处理所有Connector收到的请求

-->

<Service name="Tomcat-Standalone">

<!-- Coyote HTTP/1.1 Connector

className : 该Connector的实现类是org.apache.coyote.tomcat4.CoyoteConnector

\*\*\*port : 在端口号8080处侦听来自客户browser的HTTP1.1请求\*\*\*

minProcessors : 该Connector先创建5个线程等待客户请求，每个请求由一个线程负责

maxProcessors : 当现有的线程不够服务客户请求时，若线程总数不足75个，则创建新线程来处理请求

acceptCount : 当现有线程已经达到最大数75时，为客户请求排队

当队列中请求数超过100时，后来的请求返回Connection refused错误

redirectport : 当客户请求是https时，把该请求转发到端口8443去

其它属性略

-->

<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"

port="8080"

minProcessors="5" maxProcessors="75" acceptCount="100"

enableLookups="true"

redirectPort="8443"

debug="0"

connectionTimeout="20000"

useURIValidationHack="false"

disableUploadTimeout="true" />

<!-- Engine用来处理Connector收到的Http请求

它将匹配请求和自己的虚拟主机，并把请求转交给对应的Host来处理

默认虚拟主机是localhost

-->

<Engine name="Standalone" defaultHost="localhost" debug="0">

<!-- 日志类，目前没有看到，略去先 -->

<Logger className="org.apache.catalina.logger.FileLogger" .../>

<!-- Realm，目前没有看到，略去先 -->

<Realm className="org.apache.catalina.realm.UserDatabaseRealm" .../>

<!-- 虚拟主机localhost

appBase : 该虚拟主机的根目录是webapps/

它将匹配请求和自己的Context的路径，并把请求转交给对应的Context来处理

-->

<Host name="localhost" debug="0" appBase="webapps" unpackWARs="true" autoDeploy="true">

<!-- 日志类，目前没有看到，略去先 -->

<Logger className="org.apache.catalina.logger.FileLogger" .../>

<!-- Context，对应于一个Web App

path : 该Context的路径名是""，故该Context是该Host的默认Context

docBase : 该Context的根目录是webapps/mycontext/

-->

<Context path="" docBase="mycontext" debug="0"/>

<!-- 另外一个Context，路径名是/wsota -->

<Context path="/wsota" docBase="wsotaProject" debug="0"/>

</Host>

</Engine>

</Service>

</Server>

<!----------------------------------------------------------------------------------------------->

Zabbix密码重置（通过数据库update直接重置）

mysql> use zabbix;

mysql> [update](https://www.centos.bz/tag/update/) users set passwd='5fce1b3e34b520afeffb37ce08c7cd66' where userid='1';

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

解释：5fce1b3e34b520afeffb37ce08c7cd66 = zabbix

因为zabbix默认密码就是zabbix

# Apache和Apache Tomcat的区别和联系

Apache是web服务器（静态解析，如HTML），tomcat是java应用服务器（动态解析，如JSP）　　   
Tomcat只是一个servlet(jsp也翻译成servlet)容器，可以认为是apache的扩展，但是可以独立于apache运行   
同为服务器，Tomcat的特别之处在于它还是Java的Web容器，能够处理JSP；而Apache一般是与PHP结合，并不能处理JSP。

#### 两者从以下几点可以比较的：

1、两者都是apache组织开发的 　　   
2、两者都有HTTP服务的功能 　　   
3、两者都是开源免费的

### 联系

1）Apache是普通服务器，本身只支持html即普通网页，可以通过插件支持php，还可以与Tomcat连通(Apache单向连接Tomcat，就是说通过Apache可以访问Tomcat资源，反之不然)。　　　　   
2）Apache只支持静态网页，但像Jsp动态网页就需要Tomcat来处理。　　   
3）Apache和Tomcat整合使用：　　　　   
如果客户端请求的是静态页面，则只需要Apache服务器响应请求；　　　　   
如果客户端请求动态页面，则是Tomcat服务器响应请求，将解析的JSP等网页代码解析后回传给Apache服务器，再经Apache返回给浏览器端。　　　　   
这是因为jsp是服务器端解释代码的，Tomcat只做动态代码解析，Apache回传解析好的静态代码，Apache+Tomcat这样整合就可以减少Tomcat的服务开销。　　   
4）Apache和Tomcat是独立的，在同一台服务器上可以集成。

### 区别

Apache是有C语言实现的，支持各种特性和模块从而来扩展核心功能；Tomcat是Java编写的，更好的支持Servlet和JSP。　　   
1、Apache是Web服务器，Web服务器传送(serves)页面使浏览器可以浏览，Web服务器专门处理HTTP请求(request)，但是应用程序服务器是通过很多协议来为应用程序提供 (serves)商业逻辑(business logic)。　　   
Tomcat是运行在Apache上的应用服务器，应用程序服务器提供的是客户端应用程序可以调用(call)的方法 (methods)。它只是一个servlet(jsp也翻译成servlet)容器，可以认为是Apache的扩展，但是可以独立于apache运行。　　   
2、Apache是普通服务器，本身只支持html静态普通网页。不过可以通过插件支持PHP，还可以与Tomcat连通(单向Apache连接Tomcat,就是说通过Apache可以访问Tomcat资源，反之不然)，Tomcat是jsp/servlet容器，同时也支持HTML、JSP、ASP、PHP、CGI等，其中CGI需要一些手动调试，不过很容易的。　　   
3、Apache侧重于http server，Tomcat侧重于servlet引擎，如果以standalone方式运行，功能上Tomcat与apache等效支持JSP，但对静态网页不太理想。　　   
4、Apache可以运行一年不重启，稳定性非常好，而Tomcat则不见得。　　   
5、首选web服务器是Apache，但Apache解析不了的jsp、servlet才用tomcat。　　6、Apache是很最开始的页面解析服务，tomcat是后研发出来的，从本质上来说tomcat的功能完全可以替代Apache，但Apache毕竟是tomcat的前辈级人物，并且市场上也有不少人还在用Apache，所以Apache还会继续存在，不会被取代，apache不能解析java的东西，但解析html速度快。

### 两者例子：

Apache是一辆车，上面可以装一些东西如html等，但是不能装水，要装水必须要有容器（桶），而这个桶也可以不放在卡车上，那这个桶就是TOMCAT。

### 两者整合：

Apache是一个web服务器环境程序，启用他可以作为web服务器使用不过只支持静态网页，不支持动态网页，如asp、jsp、php、cgi　　   
如果要在Apache环境下运行jsp就需要一个解释器来执行jsp网页，而这个jsp解释器就是Tomcat　　   
那为什么还要JDK呢？因为jsp需要连接数据库的话就要jdk来提供连接数据库的驱程，所以要运行jsp的web服务器平台就需要APACHE+TOMCAT+JDK

### 整合的好处：

如果客户端请求的是静态页面，则只需要Apache服务器响应请求　　如果客户端请求动态页面，则是Tomcat服务器响应请求　　   
因为jsp是服务器端解释代码的，这样整合就可以减少Tomcat的服务开销