Catalina.bat是tomcat所有脚本中最重要的脚本，完成几乎所有的tomcat操作。如启动，关闭等等,都是由catalina.bat脚本来完成的。接下来，我将对Tomcat catalina.bat脚本进行分析。   
  
    首先省去catalina.bat开头诸多注解，这些注解主要是讲解各个变量是干什么的。需要的话，自己看下英文就可以了。这里就不翻译了。   
  
rem Guess CATALINA\_HOME if not defined  查看是否在tomcat目录下，与startup.bat里相同，不解释了。需要的话可以看我的另一篇博客。   
set CURRENT\_DIR=%cd%   
if not "%CATALINA\_HOME%" == "" goto gotHome   
set CATALINA\_HOME=%CURRENT\_DIR%   
if exist "%CATALINA\_HOME%\bin\catalina.bat" goto okHome   
cd ..   
set CATALINA\_HOME=%cd%   
cd %CURRENT\_DIR%   
:gotHome   
if exist "%CATALINA\_HOME%\bin\catalina.bat" goto okHome   
echo The CATALINA\_HOME environment variable is not defined correctly   
echo This environment variable is needed to run this program   
goto end   
:okHome   
  
rem Get standard environment variables   
if exist "%CATALINA\_HOME%\bin\setenv.bat" call "%CATALINA\_HOME%\bin\setenv.bat" 如果存在setenv.bat脚本，调用它，我的tomcat 没有这个脚本   
  
rem Get standard Java environment variables   
if exist "%CATALINA\_HOME%\bin\setclasspath.bat" goto okSetclasspath 查看是否存在setclasspath.bat脚本，如果存在，转到okSetclasspath位置   
echo Cannot find %CATALINA\_HOME%\bin\setclasspath.bat 否则输出下面两行，并退出   
echo This file is needed to run this program   
goto end   
:okSetclasspath    okSetclasspath位置   
  
set BASEDIR=%CATALINA\_HOME%   设定BASEDIR变量与CATALINA\_HOME变量值相同   
call "%CATALINA\_HOME%\bin\setclasspath.bat" %1   调用setclasspath.bat脚本并加上参数   
if errorlevel 1 goto end      如果存在错误 退出   
  
rem Add on extra jar files to CLASSPATH   设定JSSE\_HOME变量，如果存在加入CLASSPATH，不存在跳过   
  
if "%JSSE\_HOME%" == "" goto noJsse    检查是否存在JSSE\_HOME变量   
set CLASSPATH=%CLASSPATH%;%JSSE\_HOME%\lib\jcert.jar;%JSSE\_HOME%\lib\jnet.jar;%JSSE\_HOME%\lib\jsse.jar 如果有加入到CLASSPATH变量后面   
:noJsse                                     
set CLASSPATH=%CLASSPATH%;%CATALINA\_HOME%\bin\bootstrap.jar  将bootstrap.jar加入到CLASSPATH里   
  
if not "%CATALINA\_BASE%" == "" goto gotBase  如果CATALINA\_BASE变量不为空，跳过，转到gotBase位置   
set CATALINA\_BASE=%CATALINA\_HOME%  如果为空，将CATALINA\_BASE设为CATALINA\_HOME变量的值   
:gotBase   
  
if not "%CATALINA\_TMPDIR%" == "" goto gotTmpdir   CATALINA\_TMPDIR不为空，跳过，转到gotTmpdir位置   
set CATALINA\_TMPDIR=%CATALINA\_BASE%\temp   如果为空，将 CATALINA\_TMPDIR设为%CATALINA\_BASE%\temp变量的值（即tomcat\temp）   
:gotTmpdir   
  
if not exist "%CATALINA\_HOME%\bin\tomcat-juli.jar" goto noJuli  如果不存在tomcat-juli.jar这个类，转到noJuli位置   
set JAVA\_OPTS=%JAVA\_OPTS% -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager - Djava.util.logging.config.file="%CATALINA\_BASE%\conf \logging.properties"  如果存在，将变量加入到JAVA\_OPTS里   
:noJuli   
  
  
set JAVA\_OPTS=%JAVA\_OPTS% -Xms128m -Xmx512m -Dfile .encoding=UTF8 -Duser.timezone=GMT -Djava.security.auth.login.config=%CATALINA\_HOME%/conf/jaas.config  设定JAVA\_OPTS变量   
  
echo Using CATALINA\_BASE:   %CATALINA\_BASE%     输出CATALINA\_BASE变量值   
echo Using CATALINA\_HOME:   %CATALINA\_HOME%     输出CATALINA\_HOME变量值   
echo Using CATALINA\_TMPDIR: %CATALINA\_TMPDIR% 输出CATALINA\_TMPDIR变量值   
if ""%1"" == ""debug"" goto use\_jdk        如果变量%1里存在debug ，转到use\_jdk位置   
echo Using JRE\_HOME:        %JRE\_HOME%     输出JRE\_HOME变量值   
goto java\_dir\_displayed      转到java\_dir\_displayed   
:use\_jdk   
echo Using JAVA\_HOME:       %JAVA\_HOME%    输出JAVA\_HOME变量值   
:java\_dir\_displayed   
                        下面几行设定相应变量   
set \_EXECJAVA=%\_RUNJAVA%          
set MAINCLASS=org.apache.catalina.startup.Bootstrap   
set ACTION=start   
set SECURITY\_POLICY\_FILE=   
set DEBUG\_OPTS=   
set JPDA=   
  
if not ""%1"" == ""jpda"" goto noJpda   
set JPDA=jpda   
if not "%JPDA\_TRANSPORT%" == "" goto gotJpdaTransport   
set JPDA\_TRANSPORT=dt\_shmem   
:gotJpdaTransport   
if not "%JPDA\_ADDRESS%" == "" goto gotJpdaAddress   
set JPDA\_ADDRESS=jdbconn   
:gotJpdaAddress   
if not "%JPDA\_SUSPEND%" == "" goto gotJpdaSuspend   
set JPDA\_SUSPEND=n   
:gotJpdaSuspend   
if not "%JPDA\_OPTS%" == "" goto gotJpdaOpts   
set JPDA\_OPTS=-Xdebug -Xrunjdwp:transport=%JPDA\_TRANSPORT%,address=%JPDA\_ADDRESS%,server=y,suspend=%JPDA\_SUSPEND%  
:gotJpdaOpts   
shift   
:noJpda   
  
if ""%1"" == ""debug"" goto doDebug    如果%1为debug，转到doDebug，运行debug模式   
if ""%1"" == ""run"" goto doRun        如果%1为run，转到doRun，运行正常模式   
if ""%1"" == ""start"" goto doStart    如果%1为start，转到doStart，启动tomcat   
if ""%1"" == ""stop"" goto doStop      如果%1为stop，转到doStop，关闭tocmat   
if ""%1"" == ""version"" goto doVersion 如果%1为version，转到doVersion，显示tomcat的版本号   
  
echo Usage:  catalina ( commands ... ) 如果%1没有上述内容，输出下面几行，并结束   
echo commands:   
echo   debug             Start Catalina in a debugger   
echo   debug -security   Debug Catalina with a security manager   
echo   jpda start        Start Catalina under JPDA debugger   
echo   run               Start Catalina in the current window   
  
echo   run -security     Start in the current window with security manager   
echo   start             Start Catalina in a separate window   
echo   start -security   Start in a separate window with security manager   
echo   stop              Stop Catalina   
echo   version           What version of tomcat are you running?   
goto end   
  
:doDebug   
shift                    将%2里的值转到%1   
set \_EXECJAVA=%\_RUNJDB%  将变量 \_EXECJAVA设为\_RUNJDB变量的值   
set DEBUG\_OPTS=-sourcepath "%CATALINA\_HOME%\..\..\jakarta-tomcat-catalina\catalina\src\share"    
设定DEBUG\_OPTS变量   
  
if not ""%1"" == ""-security"" goto execCmd      
如果%1不为-security，转到execCmd位置   
  
shift       将%2里的值转到%1   
echo Using Security Manager       输出该行   
set SECURITY\_POLICY\_FILE=%CATALINA\_BASE%\conf\catalina.policy   
设定SECURITY\_POLICY\_FILE变量的值   
  
goto execCmd     转到execCmd位置   
  
:doRun   
shift        将%2里的值转到%1   
if not ""%1"" == ""-security"" goto execCmd   如果%1不为-security，转到execCmd位置   
shift        将%2里的值转到%1   
echo Using Security Manager   输出该行   
set SECURITY\_POLICY\_FILE=%CATALINA\_BASE%\conf\catalina.policy   
设定SECURITY\_POLICY\_FILE变量的值   
  
goto execCmd  转到execCmd位置   
  
:doStart   
shift        将%2里的值转到%1   
if not "%OS%" == "Windows\_NT" goto noTitle  如果OS变量不为Windows\_NT，转到noTitle   
set \_EXECJAVA=start "Tomcat" %\_RUNJAVA%       设定\_EXECJAVA变量的值   
goto gotTitle      转到gotTitle位置   
:noTitle   
set \_EXECJAVA=start %\_RUNJAVA%    设定 \_EXECJAVA 变量的值   
:gotTitle                 
if not ""%1"" == ""-security"" goto execCmd   如果%1不为-security，转到execCmd位置   
shift                    将%2里的值转到%1   
echo Using Security Manager        输出该行   
set SECURITY\_POLICY\_FILE=%CATALINA\_BASE%\conf\catalina.policy     
设定SECURITY\_POLICY\_FILE变量的值   
  
goto execCmd       转到execCmd位置   
  
:doStop   
shift                 将%2里的值转到%1   
set ACTION=stop       将ACTION的变量设为stop   
set CATALINA\_OPTS=    设CATALINA\_OPTS为空   
  
goto execCmd          转到execCmd位置   
  
:doVersion            显示tomcat版本号   
%\_EXECJAVA% -classpath "%CATALINA\_HOME%\server\lib\catalina.jar" org.apache.catalina.util.ServerInfo   执行该命令   
goto end              结束该程序   
  
:execCmd             
rem Get remaining unshifted command line arguments and save them in the   
以下几行将命令参数存入CMD\_LINE\_ARGS变量中   
  
set CMD\_LINE\_ARGS=   
:setArgs   
if ""%1""=="""" goto doneSetArgs   
set CMD\_LINE\_ARGS=%CMD\_LINE\_ARGS% %1   
shift   
goto setArgs   
:doneSetArgs   
  
rem Execute Java with the applicable properties   
if not "%JPDA%" == "" goto doJpda        如果JPDA变量不为空，转到doJpda位置   
if not "%SECURITY\_POLICY\_FILE%" == "" goto doSecurity   
如果SECURITY\_POLICY\_FILE变量不为空，转到doSecurity位置   
  
如果都没有执行下面命令，并结束该程序   
%\_EXECJAVA% %JAVA\_OPTS% %CATALINA\_OPTS% %DEBUG\_OPTS% -Djava.endorsed.dirs="%JAVA\_ENDORSED\_DIRS%" -classpath "%CLASSPATH%" -Dcatalina.base="%CATALINA\_BASE%" -Dcatalina.home="%CATALINA\_HOME%" -Djava.io.tmpdir="%CATALINA\_TMPDIR%" %MAINCLASS% %CMD\_LINE\_ARGS% %ACTION%   
goto end   
:doSecurity        执行下面命令，并结束该程序   
%\_EXECJAVA% %JAVA\_OPTS% %CATALINA\_OPTS% %DEBUG\_OPTS% -Djava.endorsed.dirs="%JAVA\_ENDORSED\_DIRS%" -classpath "%CLASSPATH%" -Djava.security.manager -Djava.security.policy=="%SECURITY\_POLICY\_FILE%" -Dcatalina.base="%CATALINA\_BASE%" -Dcatalina.home="%CATALINA\_HOME%" -Djava.io.tmpdir="%CATALINA\_TMPDIR%" %MAINCLASS% %CMD\_LINE\_ARGS% %ACTION%   
goto end   
:doJpda     
如果SECURITY\_POLICY\_FILE变量不为空，转到doSecurityJpda位置，为空执行下面命令，并结束该程序     
  
if not "%SECURITY\_POLICY\_FILE%" == "" goto doSecurityJpda   
%\_EXECJAVA% %JAVA\_OPTS% %CATALINA\_OPTS% %JPDA\_OPTS% %DEBUG\_OPTS% -Djava.endorsed.dirs="%JAVA\_ENDORSED\_DIRS%" -classpath "%CLASSPATH%" -Dcatalina.base="%CATALINA\_BASE%" -Dcatalina.home="%CATALINA\_HOME%" -Djava.io.tmpdir="%CATALINA\_TMPDIR%" %MAINCLASS% %CMD\_LINE\_ARGS% %ACTION%   
goto end   
:doSecurityJpda  执行下面命令，并结束该程序   
%\_EXECJAVA% %JAVA\_OPTS% %CATALINA\_OPTS% %JPDA\_OPTS% %DEBUG\_OPTS% -Djava.endorsed.dirs="%JAVA\_ENDORSED\_DIRS%" -classpath "%CLASSPATH%" -Djava.security.manager -Djava.security.policy=="%SECURITY\_POLICY\_FILE%" -Dcatalina.base="%CATALINA\_BASE%" -Dcatalina.home="%CATALINA\_HOME%" -Djava.io.tmpdir="%CATALINA\_TMPDIR%" %MAINCLASS% %CMD\_LINE\_ARGS% %ACTION%   
goto end   
  
:end   
  
终结：   
  
   catalina.bat咋一看还以为是什么windows程序，可以双击运行，实际上就是调用java命令运行Bootstrap类。从上面代码可以看 出tomcat确实是一个纯java的程序，脚本最后都变成直接使用java命令执行程序，与我们普通写的java程序，没有什么不同。只不过由于 tomcat可以使用各种众多的模式（如debug,Security等），以及各种需要各种参数所以不得不使用脚本来执行。   
  
如果你想看看你到底使用了什么命令可以在“if not "%SECURITY\_POLICY\_FILE%" == "" goto doSecurity”这行下面添加两行。   
       echo  %\_EXECJAVA% %JAVA\_OPTS% %CATALINA\_OPTS% %DEBUG\_OPTS% -Djava.endorsed.dir ....（下面那段，是一行但太长，折行了）   
       pause   
   第一行的命令是打印这具话，系统会将% %里面的变量提换成找到的值并输出。第二行是暂停程序，你可以通过任意键来恢复运行。   
   下面是我的程序打印的结果：   
  
start "Tomcat" "C:\Program Files\Java\jdk1.6.0\_01\bin\java"  -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.util.logging.config.file="C:\work\tomcat\conf\logging.properties" -Xms128m -Xmx512m -Dfile.encoding=UTF8 -Duser.timezone=GMT -Djava.security.auth.login.config=C:\work\tomcat/conf/jaas.config   -Djava.endorsed.dirs="C:\work\tomcat\common\endorsed" -classpath "C:\Program Files\Java\jdk1.6.0\_01\lib\tools.jar;C:\work\tomcat\bin\bootstrap.jar" -Dcatalina.base="C:\work\tomcat" -Dcatalina.home="C:\work\tomcat" -Djava.io.tmpdir="C:\work\tomcat\temp" org.apache.catalina.startup.Bootstrap  start   
  
    start "tomcat"是另开一个窗口，窗口名是tomcat的意思，你可以去掉这部分，完将你程序打印的拷贝到命令行下，点回车，看是不是一样运行 tomcat了。Catalina.bat startup实际时就将各种系统变量加以总结，输出成这个命令。   呵呵!

=========================================================================

在安装版的Tomcat中没有catalina.bat，实际上这些catalina.bat的设置都是对注册表的修改，对应的注册表键值如下

HKEY\_LOCAL\_MACHINE\SOFTWARE\Apache Software Foundation\Procrun 2.0\Tomcat6\Parameters\Java\Options

=========================================================================

在使用MyEclipse+Tomcat发布项目的时候通常会给如下提示信息：[org.apache.catalina.core.AprLifecycleListener]-[INFO] The Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path: （...你的系统环境PATH路径...）

在停止时发现出现如下信息   
[org.apache.catalina.core.AprLifecycleListener]-[INFO] Failed shutdown of Apache Portable Runtime

解决：Windows下的安装配置与使用   
1、先去 http://tomcat.heanet.ie/native/ 下载编译好的 tcnative-1.dll 文件，也可以下载tomcat-native.zip自己进行编译   
2、修改一下 catalina.bat，加上   
　　set CATALINA\_OPTS="-Djava.library.path=../../APR/lib"   
因为我把这个DLL放在与Tomcat目录同级的APR/lib目录下，所以采用了相对路径。或者把dll加到你的系统变量PATH里之类的，最简单的方法应当就是把tcnative-1.dll文件放在Tomcat的bin目录下：）   
  
这样设置后，启动可以看到信息如下：   
[org.apache.coyote.http11.Http11AprProtocol]-[INFO] Initializing Coyote HTTP/1.1 on http-9091   
在停止时发现出现如下信息   
[org.apache.coyote.http11.Http11AprProtocol]-[INFO] Pausing Coyote HTTP/1.1 on http-9091   
......   
[org.apache.coyote.http11.Http11AprProtocol]-[INFO] Stopping Coyote HTTP/1.1 on http-9091   
  
这样就配置成功了。