

TomC@ - HowTo's

by Peter Rossbach, Michael Kloss

1. Apache / Tomcat via JNI

Peter Roßbach und Michael Kloss

Um den Tomcat mit dem Apache zu kombinieren gibt es mehrere Möglichkeiten. Man kann dies über eine mod_proxy Konfiguration im Apache erreichen, oder über das AJP-Protokoll. Diese Möglichkeiten sind auch gut dokumentiert und häufig eingesetzt und getestet. Eine dritte Möglichkeit ist die Kommunikation über das Java Native Interface (JNI). Diese Möglichkeit ist leider nicht gut dokumentiert und wohl auch nicht gut getestet. Allerdings haben wir es inzwischen geschafft eine lauffähige Konstellation zu finden.

Peter Roßbach

Im Tomcat 5 ist eine JNI Unterstützung jetzt mit dem JK2 2.0.4 und Apache 2.0.48 auch möglich. Mit der Hilfe des Bug Reports 27167 wurde nun eine lauffähige Konfiguration mit dem aktuellen CVS HEAD zum Tomcat 5.0.20 (10.3.2004) hergestellt. Vermutlich wird diese dann auch mit neueren Versionen ab Tomcat 4.1.31 zusammenarbeiten.

1.1. Apache 2.0.48 / Tomcat 4.1.27 via JK2 2.0.4 JNI

Für die hier dargestellte Konfiguration benötigen Sie folgende Versionen:

- Apache Tomcat 4.1.27 (die Versionen **darunter** funktionieren auch)
- Apache HTTP-Server 2.0.44
- JK2 Connectoren 2.0.2

Nachdem Sie alle Versionen haben und den Apache und den Tomcat installiert haben, müssen Sie das .dll bzw. .so-Modul in das Apache-Modules-Verzeichnis kopieren und folgenden Eintrag in die httpd.conf hinzufügen.

```
LoadModule jk2_module modules\mod_jk2.dll
```

Auf der Seite des Tomcat benötigen Sie im conf-Verzeichnis eine Datei jk2.properties. In dieser Datei müssen folgende Einträge stehen:

```
## THIS FILE MAY BE OVERRIDEN AT RUNTIME. MAKE SURE TOMCAT IS STOPED
```

```
## WHEN YOU EDIT THE FILE.
## COMMENTS WILL BE _LOST_

# Just to check if the the config is working
shm.file=c:/Programme/Apache/Apache2.0.44/logs/jk2.shm

# In order to enable jni use any channelJni directive
channelJni.disabled = 0

# For Connecting to Tomcat outprocess
# apr.jniModeSo=/opt/apache2/modules/mod_jk2.so

# If set to inprocess the mod_jk2 will Register natives itself
# This will enable the starting of the Tomcat from mod_jk2
apr.jniModeSo=inprocess
```

Nun ist es fast geschafft. Allerdings fehlen noch die wichtigsten Konfigurationen auf der Apache-Seite. Hierzu ist eine workers2.properties Datei im conf-Verzeichnis des Apaches anzulegen mit folgendem Inhalt:

```
[logger]
level=DEBUG

[config:]
file=${serverRoot}/conf/workers2.properties
debug=0
debugEnv=0

[uriMap:]
info=Maps the requests. Options: debug
debug=0

[shm:]
info=Scoreboard. Required for reconfiguration and status with multiprocess servers
file=${serverRoot}/logs/jk2.shm
size=1000000
debug=0
disabled=0

[workerEnv:]
info=Global server options
timing=1
debug=0

[lb:lb]
info=Default load balancer.
debug=0

[channel.jni:jni]
info=The jni channel, used if tomcat is started inprocess

[status:]
```

TomC@ - HowTo's

```
info=Status worker, displays runtime informations

[vm:]
info=Parameters used to load a JVM in the server process
OPT=-Djava.class.path=PATH-TO-tomcat-jni.jar; \
    PATH-TO-commons-logging.jar; \
    PATH-TO-log4j-1.2.8.jar;PATH-TO-tools.jar
OPT=-Dtomcat.home=PATH-TO-tomcat.home
OPT=-Dcatalina.home=PATH-TO-catalina.home
#Example
#OPT=-Djava.class.path= \
    C:/server/jakarta/jakarta-tomcat-4.1.29/bin/tomcat-jni.jar; \
    C:/server/jakarta/jakarta-tomcat-4.1.29/server/lib/commons-logging.jar;\
    C:/server/jakarta/jakarta-tomcat-4.1.29/server/lib/log4j-1.2.8.jar; \
    C:/java/j2sdk1.4.1_01/lib/tools.jar
#OPT=-Dtomcat.home=C:/server/jakarta/jakarta-tomcat-4.1.29
#OPT=-Dcatalina.home=C:/server/jakarta/jakarta-tomcat-4.1.29
OPT=-Xmx128M
disabled=0

[worker.jni:onStartup]
info=Command to be executed by the VM on startup. This one will start tomcat.
class=org/apache/jk/apr/TomcatStarter
ARG=start
disabled=0
stdout=${serverRoot}/logs/stdout.log
stderr=${serverRoot}/logs/stderr.log

[worker.jni:onShutdown]
info=Command to be executed by the VM on shutdown. This one will stop tomcat.
class=org/apache/jk/apr/TomcatStarter
ARG=stop
disabled=0

[uri:/jkstatus/*]
info=Display status information and checks the config file for changes.
group=status:

[uri:/examples/*]
info=Map the whole webapp
```

Wenn Sie nun den Apache starten und den Browser auf <http://mydomain/examples> zeigen lassen, sollten Sie die Beispiel-Seiten des Tomcat sehen und dort sowohl die Servlets als auch die JSPs ansehen können. Zur Kontrolle, ob der Tomcat wirklich gestartet ist, schauen Sie in den Log-Dateien `stderr.log` und `stdout.log` im Apache-conf-Verzeichnis.

1.2. Apache 2.0.48 / Tomcat 5.0.20 via JK2 2.0.4 JNI

Mit dem Tomcat 5 haben sich die start und Stopp Parameter und der notwendige Klassenpfad geändert. Einen aktuelle `workers2.properties` Datei finden Sie hier:

Für die hier dargestellte Konfiguration benötigen Sie folgende Versionen:

- Apache Tomcat 5.0.20 (die Versionen **darunter** funktionieren nicht)
- Apache HTTP-Server 2.0.48
- JK2 Connectoren 2.0.4

```
[logger]
level=DEBUG

[config:]
file=${serverRoot}/conf/workers2.properties
debug=0
debugEnv=0

[uriMap:]
info=Maps the requests. Options: debug
debug=0

[shm:]
info=Scoreboard. Required for reconfiguration and status with multiprocess servers
file=${serverRoot}/logs/jk2.shm
size=1000000
debug=0
disabled=0

[workerEnv:]
info=Global server options
timing=1
debug=0

[lb:lb]
info=Default load balancer.
debug=0

[channel.jni:jni]
info=The jni channel, used if tomcat is started inprocess

[status:]
info=Status worker, displays runtime informations

[vm:]
info=Parameters used to load a JVM in the server process
#OPT=-Xdebug -Xnoagent -Xrunjwdp:transport=dt_socket,address=8000,server=y,suspend=n
OPT=-Xmx32m
OPT=-Xms32m
OPT=-Djava.home=D:/java/j2sdk1.4.2_03
OPT=-Dcatalina.config=file:../conf/catalina.properties
OPT=-Dcatalina.home=C:/tomcat5/jakarta-tomcat-5/build
OPT=-Dcatalina.base=C:/server/jk2jvm-server
OPT=-Djava.class.path=D:/java/j2sdk1.4.2_03/lib/tools.jar;C:/tomcat5/jakarta-tomcat-5/b

disabled=0
```

TomC@ - HowTo's

```
[worker.jni:onStartup]
info=Command to be executed by the VM on startup. This one will start tomcat.
class=org/apache/jk/apr/TomcatStarter
ARG=startd
disabled=0
stdout=${serverRoot}/logs/stdout.log
stderr=${serverRoot}/logs/stderr.log

[worker.jni:onShutdown]
info=Command to be executed by the VM on shutdown. This one will stop tomcat.
class=org/apache/jk/apr/TomcatStarter
ARG=stopd
disabled=0

[uri:/jkstatus/*]
info=Display status information and checks the config file for changes.
group=status:

[uri:/myapps/*]
info=Map the whole webapp

[uri:/manager/*]
info=Map the whole webapp
```