

In diesem Kursteil

- Kursübersicht
 - Worum geht's
- Einleitung
- JiniWorks
 - Die Jini Dienste
- Jini
- HelloWorld

Jini

-

Der Start

1.1. Kursübersicht

Dieses Skript zeigt Schritt für Schritt, wie Sie ein erstes Jini-Beispiel zum Laufen bringen können, hoffentlich!

Die Anleitung hier enthält Hintergrundinformationen. Auf dem Web finden Sie ein EXE, welches alle Software, ausser dem JDK (ab 1.4) enthält.

Sie brauchen kaum Anpassungen zu machen. Überprüfen Sie aber die vorgeschlagenen Verzeichnisse in der Datei SetEnvironment.bat.

Grundvoraussetzung für das Funktionieren ist ein TCP/IP Stack. Sie sollten also gar nicht erst versuchen, Jini Beispiele ohne Netzwerk-Anschluss zu testen.

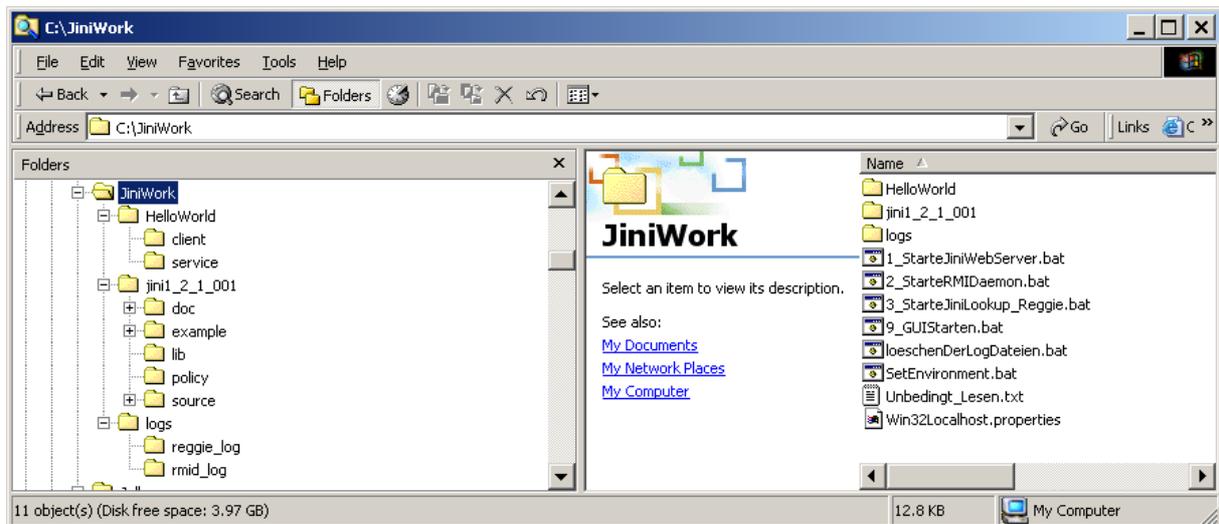
JINI - DER START

1.2. Einleitung

Als Beispiel verwenden wir eine Implementierung von „Hello World“. Sie brauchen aber die Details des Programms nicht zu verstehen. Ziel ist es, und dies sollten Sie unbedingt erreichen, ein erstes Jini Programm zum Laufen zu bringen. Details des Programms können Sie später immer noch nachschlagen. Aber zum Laufen sollten Sie das Programm bringen.

Falls dies nicht klappt, schreiben Sie mir unbedingt ein email!

1.3. Verzeichnisstruktur



1.3.1. JiniWork

Dieses Verzeichnis ist sozusagen Ihr Jini Root Verzeichnis. Wie Sie sehen, befindet sich darin

- Das eigentliche Jini System, wie man es von Sun herunterladen kann
- Skripts / Batch Dateien, für die Jini Dienste und eine Datei, in der die Umgebungsvariablen gesetzt werden.
Wir werden noch im Detail auf diese Dateien / Skripte eingehen.
- Das HelloWorld Verzeichnis. Darauf werden wir noch zurückkommen.

JINI - DER START



SetEnvironment.bat

```
@echo off
@echo off
Rem -----
-----
Rem Setzen aller Umgebungsvariablen für Jini Beispiele
Rem -----
set JAVA_HOME=c:\j2sdk1.4.1_01
set JINI_WORK=c:\JiniWork
set JINI_VERSION=jini1_2_1_001
Rem
set JINI_HOME=%JINI_WORK%\%JINI_VERSION%
set JINI_LIB=%JINI_HOME%\lib
set JINI_LOGS=%JINI_WORK%\logs
Rem
set REGGIE_LOG=%JINI_LOGS%\reggie_log
set RMID_LOG=%JINI_LOGS%\rmid_log
Rem alles auf einer Zeile
set CLASSPATH=%JINI_LIB%\jini-core.jar;%JINI_LIB%\jini-
ext.jar;%JINI_LIB%\sun-util.jar;%JINI_LIB%\sun-util.jar;
set SECURITY_POLICY=%JINI_HOME%\policy\policy.all
Rem
Rem Web / File Server für die Jini JAR Dateien
Rem
set JINI_HTTP_HOST=localhost
set JINI_HTTP_PORT=8081
Rem
Rem Reggie
Rem
set JINI_GROUP=public
Rem
Rem JavaSpaces
Rem
set JAVA_SPACE=JavaSpace
Rem
Rem Web / File Server für Anwendungs-Klassen
Rem (Proxies, Interfaces u.a.)
Rem Dieser Server kann auch mit dem Jini Web / File Server
Rem kombiniert werden...
Rem ich spiele aber gerne mit unterschiedlichen Verzeichnissen
Rem
set APP_HTTP_HOST=localhost
set APP_HTTP_PORT=8085
```

JINI - DER START

- 1) J2SDK
Ich verwende in der Regel die neuste Version. Jetzt gerade habe ich 1.4.1_01.
Verwenden Sie ruhig eine neuere oder nicht viel ältere (wegen den Security Problemen mit RMID).
- 2) JINI_...
Dies sind die grundsätzlichen Variablen für die Jini Konfiguration.
- 3) CLASSPATH
sollten Sie so stehen lassen oder einem neueren Release von Jini anpassen.
- 4) ..._HTTP_...
ich verwende in diesem HelloWorld ZWEI Web Server
Der Erste dient Jini als File-Server. Sein Root ist JINI_LIB.
Der Zweite dient der Applikation als File-Server. Sein Root ist
%JINI_WORK%\HelloWorld\service, wie wir noch sehen werden (in den Skripts im
HelloWorld verzeichnis).



loeschenDerLogDateien.bat

Dieses Hilfsskript soll jeweils vor dem Starten der Jini Dienste alle Logs löschen.
Dieser Schritt ist sehr wichtig! Wenn er nicht ausgeführt wird, kann Reggie, der Jini Lookup Service, nicht gestartet werden.



1_StarteJiniWebServer.bat

```
@echo off
@echo Starten des Web Servers / Datei Servers...
@echo Testen:
@echo beispielsweise im Browser http://localhost:8080/Test.txt
@echo liefert die Textdatei aus dem %JINI_HOME%\lib
@echo Verzeichnis
@echo Oder:
@echo telnet localhost 8080 GET /Test.txt
@echo (der '/' vor Text.txt ist ein Muss)
Rem -----
Rem Jini Umgebung definieren
Rem -----
call SetEnvironment.bat
Rem -----
Rem die nächsten drei Zeilen sind in der Datei eine einzige
Rem Zeile!!
%JAVA_HOME%\bin\java -jar -classpath %CLASSPATH%
%JINI_LIB%\tools.jar -port %JINI_HTTP_PORT% -dir %JINI_LIB% -
verbose
pause
```

Mit dem Jini System werden verschiedene Hilfsprogramme mitgeliefert, unter anderem ein Web / Datei Server. Dieser steht in %JINI_LIB%\tools.jar.

Die Angaben dazu (-port %JINI_HTTP_PORT% -dir %JINI_LIB%) wurden in der Datei SetEnvironment.bat definiert.

JINI - DER START

Der Port wird so gewählt, dass möglichst kein Konflikt mit der Anwendung oder einem bereits verwendeten Port entsteht.

Das Verzeichnis (`-dir %JINI_LIB%`) legt das Web / File Server Root fest: ab dort werden Klassen und Archive heruntergeladen, bzw. dort werden sie gesucht.

Im Kommentar finden Sie Hinweise, wie Sie das Funktionieren des Servers überprüfen können (`@echo Testen: ...`).



2_StarteRMIDaemon.bat

```
@echo off
call SetEnvironment.bat
call loeschenDerLogDateien.bat
Rem
@echo Starten des RMID Daemon...
@echo Der Daemon zeigt keine Aktivitaet an (Daemon)
%JAVA_HOME%\bin\rmid -J-Dsun.rmi.activation.execPolicy=none -
log %RMID_LOG%
```

- 1) Als erstes werden, wie in fast allen Skripts, die Umgebungsvariablen gesetzt.
- 2) Dann müssen bereits vorhandene Log Dateien gelöscht werden.
- 3) Da Jini RMI als Kommunikations-Infrastruktur einsetzt, müssen wir den RMI Daemon starten. Der Startvorgang ist oft etwas langsam. Sie müssen also nach dem Starten etwas warten. Zudem kann `rmid` auch aus unerklärlichen Gründen aufhören zu existieren.
Also: hier liegt Potential für Ärger.
- 4) Der Ärger mit `rmid` geht gleich weiter:
Ab Version 1.3 von Java wurden die Sicherheitseinstellungen geändert. Damit Sie nicht dauernd die Meldung erhalten, dass die Security Ihnen Zugriffe verwehrt, sollten Sie an `-J-Dsun.rmi.activation.execPolicy=none` nichts ändern. Später können Sie sich immer noch darum kümmern. Aber bitte nicht gleich beim ersten Jini Beispiel.

JINI - DER START



3_StarteJiniLookup_Reggie.bat

```
@echo off
@echo Starten des Jini Lookup Service (Reggie)
call SetEnvironment.bat
call loeschenDerLogDateien.bat
Rem
Rem policy aus dem Jini Installationskit verwenden
Rem
%JAVA_HOME%\bin\java -jar %JINI_LIB%\reggie.jar
http://%JINI_HTTP_HOST%:%JINI_HTTP_PORT%/reggie-dl.jar
%SECURITY_POLICY% %REGGIE_LOG% %JINI_GROUP%
pause
```

- 1) Als erstes werden, wie in fast allen Skripts, die Umgebungsvariablen gesetzt.
- 2) Dann müssen bereits vorhandene Log Dateien gelöscht werden.
- 3) Jetzt können wir, sofern der RMI Daemon erfolgreich gestartet wurde, den Lookup Dienst von Jini starten. Das gelingt aber nicht immer problemlos.
Also: hier liegt Potential für Ärger.
Ein typischer Fehler wäre: der TCP/IP Stack ist nicht vorhanden (keine Netzwerk-Verbindung)

Im glücklichsten Fall sind jetzt alle Jini Dienste am Laufen!

Man kann auch alle Dienste aus dem „Launcher“ heraus starten:



9_GUIStarten.bat

```
@echo off
Rem -----
Rem Setzen aller Umgebungsvariablen für Jini Beispiele
Rem -----
call SetEnvironment.bat
call loeschenDerLogDateien.bat

Rem -----
%JAVA_HOME%\bin\java -cp %JINI_LIB%\jini-examples.jar
com.sun.jini.example.launcher.StartService
Win32Localhost.properties
```

Für GUI Freaks: diese Dienste kann man aus dem Launcher heraus starten:

Das obige Skript startet ein einfaches Swing GUI.

- starte den Launcher : 9_GUIStarten.bat
- laden der Properties : "File->OpenPropertyFile->'Win32Localhost.properties'"
- anpassen der Properties!
- Starten der Dienste : "Run-Tab->Start RMID ->Start WebServer ->Start Reggie"

JINI - DER START

1.3.1.1. jini_2_1_001

In diesem Verzeichnis steht Jini, entzippt, einfach heruntergeladen vom Jini / Sun Web Site. Sie brauchen im Moment nichts weiter darüber zu wissen. Unser Ziel ist es, möglichst ungestört ein Beispiel zum Laufen zu bringen.

1.3.1.2. HelloWorld



01_CompileService.bat

```
@echo off
call ..\SetEnvironment.bat
mkdir service
%JAVA_HOME%\bin\javac -classpath %CLASSPATH%;.;.. -d
%JINI_WORK%\HelloWorld\service HelloWorldServiceInterface.java
HelloWorldService.java
pause
```

Im Skript werden die Umgebungsvariablen gesetzt, das Service Verzeichnisses kreiert und die Class-Datei ins Verzeichnis `%JINI_WORK%\HelloWorld\service` verschoben, welches vorgängig angelegt wird (`mkdir service`).



02_CompileClient.bat

```
@echo off
call ..\SetEnvironment.bat
mkdir client
%JAVA_HOME%\bin\javac -classpath %CLASSPATH%;.;.. -d
%JINI_WORK%\HelloWorld\client HelloWorldServiceInterface.java
HelloWorldClient.java
pause
```

Im Skript werden die Umgebungsvariablen gesetzt, das Service Verzeichnisses kreiert und die Class-Datei ins Verzeichnis `%JINI_WORK%\HelloWorld\client` verschoben, welches vorgängig angelegt wird (`mkdir client`).

JINI - DER START



03_StarHelloWorldtHTTP.bat

```
@echo off
Rem -----
Rem JINI Umgebungsvariablen setzen
Rem -----
call ..\SetEnvironment.bat

Rem -----
@echo HTTP Server fuer den Transport von Anwendungsklassen
(Pproxies, Interfaces,...)
@echo Arbeitsverzeichnis : %JINI_WORK%\HelloWorld\service\
@echo Arbeits-Port : %APP_HTTP_PORT%
@echo Sie koennen diesen Server testen mit
http://%APP_HTTP_HOST%:%APP_HTTP_PORT%
@echo indem Sie beispielsweise eine Textdatei ins Verzeichnis
%JINI_WORK%\HelloWorld\service\
@echo kopieren und wiefolgt im Browser anzeigen lassen:
http://%APP_HTTP_HOST%:%APP_HTTP_PORT%/Test.txt
@echo Nicht vergessen: das Server Root ist
%JINI_WORK%\HelloWorld\service\
@echo -----
java -jar %JINI_LIB%\tools.jar -dir
%JINI_WORK%\HelloWorld\service -verbose -port %APP_HTTP_PORT%
pause
```

Dieser Web / File-Server wird benötigt, um Proxy-Objekte zu verschieben, sofern Sie über den Lookup-Service verlangt werden:

Falls das Beispiel erfolgreich ausgeführt wurde, steht nach dem Client-Aufruf folgende Message im DOS-Fenster des obigen Skripts:

```
HelloWorldServiceProxy.class requested from 127.0.0.1:2544
```

JINI - DER START



04_StartService.bat

```
@echo off
Rem -----
Rem JINI Umgebungsvariablen setzen
Rem -----
call ..\SetEnvironment.bat

Rem -----
java -cp %CLASSPATH%;service;.;.. -
Djava.rmi.server.codebase=http://%APP_HTTP_HOST%:%APP_HTTP_POR
T% -Djava.security.policy=%SECURITY_POLICY% HelloWorldService
pause
```

Der Service muss sich beim Lookup-Service anmelden. Unser Service besteht aus mehreren Klassen und inneren Klassen.

Falls alles gut abgelaufen ist, sollte im DOS Fenster des obigen Skripts etwa folgende Message stehen:

```
[HelloWorldService]main()
[HelloWorldService.main()]HelloWorldService()
[HelloWorldService]HelloWorldService()
[HelloWorldService]createProxy()
[HelloWorldServiceProxy]HelloWorldProxy()
[HelloWorldService.HelloWorldService()]setSecurityManager()
[HelloWorldService.HelloWorldService()]LookupDiscovery()
[HelloWorldService.HelloWorldService()]addListener()
[HelloWorldService.main()]Thread.start()
[HelloWorldService]run()
[HelloWorldService.run()]sleep() 31 Jan 2003 20:13:02 GMT
[HelloWorldService$Listener.discovered()]Es wurde ein neuer
Lookup Service entdeckt!
[HelloWorldService]registerWithLookup()
[HelloWorldService.registerWithLookup()]getServiceID()
ServiceID : 7154483b-4e3b-405a-990c-
f235acd8af82
[HelloWorldService.run()]sleep() 31 Jan 2003 20:13:12 GMT
```

JINI - DER START



05_StartClient.bat

```
@echo off
Rem -----
Rem JINI Umgebungsvariablen setzen
Rem -----
call ..\SetEnvironment.bat

Rem -----
java -cp %CLASSPATH%;client;.;.. -
Djava.rmi.server.codebase=http://%APP_HTTP_HOST%:%APP_HTTP_PORT%
-Djava.security.policy=%SECURITY_POLICY% HelloWorldClient
pause
```

Der Client sucht den Dienst mithilfe des Lookup-Dienstes. Falls er Dateien / Classes benötigt, lädt er diese dynamisch über die Codebase, die wir mithilfe des zweiten Web / File-Servers realisieren.

Falls alles gut abgelaufen ist, sollte im DOS Fenster des obigen Skripts etwa folgende Message stehen:

```
[HelloWorldClient]main()
[HelloWorldClient.main()]HelloWorldClient()
[HelloWorldClient.HelloWorldClient()]ServiceTemplate()
[HelloWorldClient.HelloWorldClient()]setSecurityManager()
[HelloWorldClient.HelloWorldClient()]LookupDiscovery()
[HelloWorldClient.HelloWorldClient()]LookupDiscovery() :
net.jini.discovery.LookupDiscovery@a2005e46
[HelloWorldClient.HelloWorldClient()]LookupDiscovery.addDiscoveryListener()
[HelloWorldClient.main()]Thread.start()
[HelloWorldClient]run()
[HelloWorldClient.run]sleep()
[HelloWorldClient$Listener]discovered()
[HelloWorldClient$Listener.discovered()]Es wurde 1 Service
gefunden
[HelloWorldClient$Listener.discovered()]Service :
com.sun.jini.reggie.RegistrarProxy@cf2bc390
[HelloWorldClient]lookForService()
=>Es wurde ein passender Service
gefunden.
[HelloWorldServiceProxy.HelloWorldProxy()]getMessage()
=>Und hier seine Message an die Welt:
Hello, world!
```

JINI - DER START

Das war's für's erste!

Und nicht vergessen: falls irgend etwas nicht geklappt hat – ein email würde mich freuen. Ich werde damit nur klüger!!

Ich habe das Ganze getestet. Bei einer Neuinstallation habe ich lediglich im Skript SetEnvironment.Bat das Jini-Root angepasst (**set JINI_WORK=c:\temp\JiniWork**). Und schon lief alles wie am Schnürchen! Aber eben: Sie finden mehr Fehler als ich, da mein PC vorbelastet ist.

JINI - DER START	1
1.1. KURSÜBERSICHT	1
1.2. EINLEITUNG	2
1.3. VERZEICHNISSTRUKTUR.....	2
1.3.1. <i>JiniWork</i>	2
1.3.1.1. jini1_2_1_001.....	7
1.3.1.2. HelloWorld.....	7