

Albrecht Weinert

Tutorial

JMX mit SSL — für Java auf Windows



Stand: 20.06.2008



Albrecht Weinert

Labor für Medien und verteilte Anwendungen (MEVA-Lab)

Fachbereich Informatik der Hochschule Bochum

JMX mit SSL — für Java auf Windows

V1.00 (18.06.2007 17:56) : neu, unvollständig

V1.01 (19.06.2008 19:52) : nur kleinere red. Änderungen

Version: V1.01

Zuletzt geändert von A. Weinert am 19.06.2008

Copyright © 2007 Albrecht Weinert. All rights reserved. a-weinert.de

Inhalt

| | |
|---|----|
| 1. Umfeld und Einordnung | 2 |
| 2. Zertifikate und Truststores als (SSL-) Infrastruktur | 3 |
| 3. Ein einheitlicher Satz von Steuerdateien | 7 |
| 3.1 Start und JMX-management.properties..... | 7 |
| 3. Rollen und Passworte | 9 |
| 3.2 Sun's Handhabung der Passwortdateisicherheit..... | 10 |
| 4. Eine mit JMX B&B-bare Java-Anwendung | 11 |
| 5. B&B (remote, SSL) einer Java-Anwendung mit JConsole | 12 |
| 6. Kopplung (remote, SSL) von Java-Anwendungen | 14 |
| 7. Was noch zu tun bleibt..... | 14 |
| 8. Résumé | 15 |
| A. Anhang | 16 |
| A1. JMX im Framework | 16 |
| A1.1 de.a_weinert.App als DynamicMBean — mit XML..... | 16 |
| A1.2 de.a_weinert.AppBase als DynamicMBean — mit XML..... | 19 |
| A2. Abkürzungen | 22 |
| A3. Literatur | 25 |

1. Umfeld und Einordnung

Es geht um Java, Java Beans und MBeans, die Instrumentierung und Bedienung und Beobachtung von Java-Anwendungen, JMX (Java Management Extensions), Sicherheit, SSL (Secure Socket Layer) sowie um den Einsatz des Werkzeugs JConsole.

Vorausgesetzt wird die Beherrschung und jeweils geeignete Installation von Java. Siehe dazu [3]..[6] im selben Verzeichnis wie dieses Dokument.

Die JMX (Java Management Extensions) spezifizieren Schnittstellen und Protokolle zur Bedienung und Beobachtung (B&B) von Java-Anwendungen. Seit Java 5 und insbesondere seit Java 6 ist JMX in die Standard-API integriert. Das nachträgliche Dranbasteln von Referenzimplementierungen etc. entfällt damit. Hinzu kommt zweierlei:

- eine Instrumentierung der JVM selbst, die u.A. Auskunft über Klassen, Threads, Garbage Collection etc. erteilt, und
- ein sehr praktischer, schön gemachter graphischer JMX-Client namens JConsole. Damit kann man mehrere JVMs und die darin laufenden Anwendungen bedienen und beobachten — und das lokal und entfernt (remote).

JMX ermöglicht es auch, dass sich mehrere Java-Anwendungen gegenseitig beeinflussen. Dies ist nicht weiter erstaunlich, da ja die erwähnte JConsole ja selbst genau so eine Java-Anwendung ist. Verteilte Java-Automatisierungsanwendungen können also JMX als ihren Kommunikationsbus (quasi als middleware) benutzen:

- JMX als Kommunikationsbus in einem mit Java-Systemen verteilt automatisierten unbemannten, kanadischen Arbeits-U-Boot.
- Ein bekannteres Beispiel ist der J2EE-Container Tomcat, der JMX für seine (remote-) Administrierungsschnittstelle benutzt.
- Alle Anwendungen, die auf dem Framework `de.a_weinert` beruhen (a-weinert.de/java/aweinertbib.html), sind optional JMX-bediene- und beobachtbar.

Bei einem Einsatz von JMX als Kommunikationsbus für verteilte Automatisierungsanwendungen muss mit Authentifizierungen, Zertifikaten und abgesicherter Kommunikation (SSL z.B.) gearbeitet werden. Zumindest muss dies in kritischen Anwendungsfällen und in einem (oft aus guten Gründen) nicht ganz abgeschotteten Netzwerk möglich sein. ^{*1)}

Die Beherrschung von JMX auch in diesem Sinne passt zum Anspruch des [MEVA-Lab](#) („[mission statement](#)“). Das vorliegende Tutorial führt durch den Einstieg in die JMX-Technologie — mit genau diesem Anspruch.

Erstes Ziel ist eine remote JMX-B&B einer Java-Anwendung mit SSL.

Am besten nachvollziehbar ist das Folgende mit einer geeigneten J(S)DK/JRE 1.6.0-Installation. Hier ist JMX am besten integriert und unterstützt. Und man kann JDK/JRE1.6.x als Plattform (incl. Browser plug in) nutzen, ohne Java5-features zu verwenden.

*1) Anders ausgedrückt: Falls man man die Technologie (JMX) nicht auch in diesem Aspekt beherrscht, kann man sie für ernsthafte verteilte Anwendungen (Automatisierung, Geschäftsprozesse) i.A. nicht nutzen.

2. Zertifikate und Truststores als (SSL-) Infrastruktur

Eine Gruppe von Automatisierungs- und B&B-Rechnern mit einem industriegängigen Betriebssystem, wie beispielsweise Windows Server 2003, tragen die betreffende verteilte Java-Anwendung. Deren Kommunikation untereinander soll mit JMX erfolgen.

Bild 1 zeigt eine solche Gruppe von Rechnern bzw. PCs als beispielhafte Konfiguration.

Die Art der Automatisierungs- oder (Geschäfts-) Prozessanwendung erfordert eine abgesicherte Kommunikation. Bezüglich der JMX-Kommunikation soll es keine Einschränkungen geben, welcher der beteiligten Rechner sich (prinzipiell) mit welchem verbinden kann. Diese Applikation-zu-Applikation- und im „remote-Fall“ damit Rechner-zu-Rechner-JMX-Kommunikationsbeziehungen sind im Bild 1 als drei gestrichelte Linien angedeutet. Die letzte Forderung bedeutet, dass es deren beliebige geben darf.

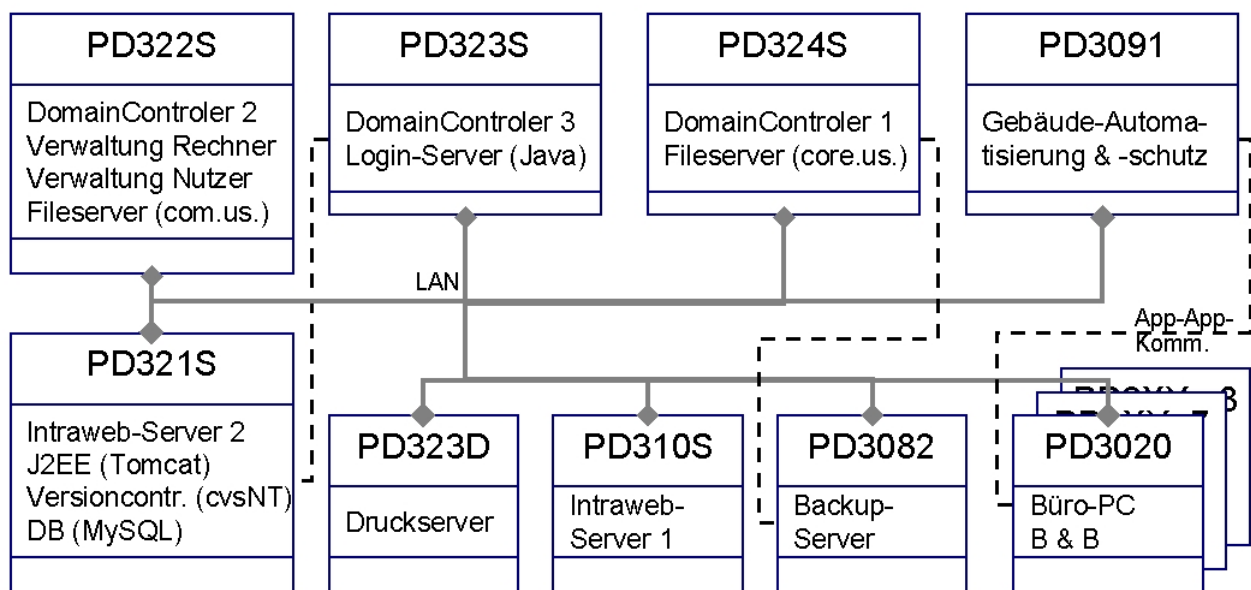


Bild 1: Eine beispielhafte Rechnerkonfiguration.

Dies ist also die Anforderungslage.

Hinweis: In den folgenden Beispielen werden die Rechnernamen der Beispielkonfiguration (Bild 1) benutzt. Ferner werden drei (Domain-) Nutzernamen, nämlich „weinert“, „wichtel“ und „garcon“, verwendet. Außerdem wird davon ausgegangen, dass auf jedem Rechner ein JDK (1.6.x) im Verzeichnis C:\programme\jdk\ installiert ist. Sinngemäße Änderungen bei anderen Namen oder Installationsgewohnheiten sind so einfach, dass nicht jeweils darauf hingewiesen wird.

Damit Java-Anwendungen von jedem zu jedem Rechner der betreffenden Gruppe (vgl. Bild 1) SSL-Verbindungen aufnehmen können, muss es für jeden Rechner ein (Server-) Zertifikat geben. Das zugehörige Schlüsselpaar muss (nur) im „keystore“ des jeweiligen Rechners hinterlegt sein. Und alle Zertifikate müssen in die „truststores“ jedes beteiligten Rechners eingetragen sein. Vergleiche dazu auch das betreffende Kapitel (zu https mit Tomcat) in [13].

Der einfachste (und mit JDK-Bordmitteln gangbare) Weg zu einer solchen (gruppen-) vollständigen Schlüsselpaar-/Zertifikat-Ausstattung führt über selbstunterschiedene Serverzertifikate.

Hinweis: Natürlich sind („fremd-“), am Besten von einer weltweit vertrauenswürdigen CA, unterschriebene Zertifikate der bessere, weiter tragende und allgemeiner verwendbare Weg. Für erste Installationen (und Gehversuche) ist der genannte einfache Weg zu empfehlen. Für eine abgeschlossene „in-house-“ Automatisierungsrechnergruppe sind selbstunterschriebenen Zertifikate eh ausreichend.

Selbst wenn man über die o.g. „besseren“ Zertifikate verfügen kann, sollte man vielleicht doch den einfachen Weg nehmen, bis der von A. Weinert gemeldete Bug Id 6557093 (keystore-Passwort betreffend) bereinigt ist.

Hinweis: Den Inhalt dieses Bug Id 6557093 können Sie nicht einsehen, da er von SUN als (passend zum Thema) sicherheitskritisch eingestuft wurde. Man kann nur aufpassen, ob er verschwindet.

Um eine (gruppen-) vollständige, selbst unterschriebene Zertifikatsausstattung zu erzeugen, bedient man sich am Besten des folgenden Scripts (Listing 1) MakeServerCert.bat :

```
@echo.
@echo MakeServerCert.bat V01.00 28.02.2007
@echo (C) 2007 Albrecht Weinert a-weinert.de
@echo.
@if %1X==X          goto :aufruf
@if /I %1==--help goto :aufruf

keytool -genkey -v -alias %1 -keyalg RSA -keypass changeit
  -storepass changeit -dname
  "CN=%1, OU=FB3-MEVA, O=fh-bochum.de, L=Bochum, S=NRW, C=DE"
  -keystore %1-keystore -validity 9999

@if not exist %1.cer goto :export
@echo Die Datei %1.cer existiert bereits
@echo Abort
@goto :end

:export
keytool -export -v -alias %1 -storepass changeit
  -keystore %1-keystore -file %1.cer

keytool -import -v -alias %1 -keypass changeit -storepass
  changeit -keystore truststore -file %1.cer -noprompt

@goto :ende

:aufruf
@echo Wechseln ins certificateFactory-Verzeichnis, wo alle
@echo (Kopien von allen) truststore und keystores sind.
@Echo.
@echo Aufruf dann dort: MakeServerCert servername
@echo.
```

```

@echo Erzeugt wird servername-keystore mit neuem Key-pair und
    Zertifikat,
@echo exportiert wird daraus als servername.cer (nur) das
    Zertifikat und
@echo importiert wird dieses in truststore
@echo sonstige Einstellungen:
@echo     passw=changeit; alias=servername; validity=9999

:ende
@Echo.

```

Listing 1: MakeServerCert.bat.

Dieses Script lässt man auf einem geeigneten Server in einem speziellen Verzeichnis

```
E:\certificateFactory
```

(zum Beispiel) für jeden Rechner der Gruppe (vgl. Bild 1) laufen:

```
E:\certificateFactory> MakeServerCert pd310s
```

```
E:\certificateFactory> MakeServerCert pd313d
```

```
E:\certificateFactory> ... u.s.w ...
```

Das Ergebnis muss danach etwa so (Listing 2) aussehen:

```

Verzeichnis von E:\certificateFactory          (PD323S)
letzte Modifikation      Größe      Dateiname
28.02.2007  18:47          1.200    makeServerCert.bat
13.04.2007  13:25          1.351    pd3021-keystore
13.04.2007  13:25           580    pd3021.cer
13.04.2007  13:25          1.350    pd3082-keystore
13.04.2007  13:25           580    pd3082.cer
13.04.2007  13:25          1.350    pd3091-keystore
13.04.2007  13:25           580    pd3091.cer
13.04.2007  13:25          1.348    pd310s-keystore
13.04.2007  13:25           580    pd310s.cer
13.04.2007  13:26          1.349    pd313d-keystore
13.04.2007  13:26           580    pd313d.cer
13.04.2007  13:24          1.349    pd321s-keystore
13.04.2007  13:24           580    pd321s.cer
28.02.2007  18:56          1.350    pd322s-keystore
28.02.2007  18:56           580    pd322s.cer
28.02.2007  18:41          1.350    pd323s-keystore
28.02.2007  18:41           580    pd323s.cer
28.02.2007  18:48          1.348    pd324s-keystore
28.02.2007  18:48           580    pd324s.cer
13.04.2007  13:26          5.531    truststore

```

Listing 2: Alle Zertifikate etc. der Gruppe erzeugt

Diese so (Listing 2) erzeugten Dateien werden wie folgt verteilt:

```
E:\certificateFactory\<<rechnername>-keystore
```

und (nur) der Vollständigkeit halber

```
E:\certificateFactory\<<rechnername>.cer
```

also beispielsweise

```
E:\certificateFactory\pd3021-keystore
```

```
E:\certificateFactory\pd3021.cer
```

kommen im jeweiligen Rechner (PD3021 im Beispiel) als Kopie in das Verzeichnis

```
C:\Programme\jdk\jre\lib\security
```

In dasselbe Verzeichnis jedes Rechners der Gruppe wird auch die Datei

```
E:\certificateFactory\truststore
```

kopiert. Dies Verzeichnis hat dann auf jedem Rechner der Gruppe (in etwa, Listing 3) folgenden Inhalt:

```
Verzeichnis von C:\Programme\jdk\jre\lib\security (PD3021)
letzte Modifikation      Größe      Dateiname
12.04.2007  12:14      40.624      cacerts
12.04.2007  12:14      2.221      java.policy
12.04.2007  12:14      9.910      java.security
12.04.2007  12:14      132      javaws.policy
12.04.2007  12:14      2.940      local_policy.jar
13.04.2007  13:25      1.351      pd3021-keystore
13.04.2007  13:25      580      pd3021.cer
13.04.2007  13:26      5.531      truststore
12.04.2007  12:14      2.469      US_export_policy.jar
```

Listing 3: Zertifikat, truststore etc. in jre\lib\security

Nun kann jeder Rechner der (Automatisierungs-) Gruppe zu jedem (mit Java-Mitteln) eine SSL-Verbindung aufnehmen.

Soll ein weiterer Rechner in die Gruppe aufgenommen werden, ist für ihn mit dem Script (Listing 1) MakeServerCert.bat in der „certificate factory“ beginnend sinngemäß zu verfahren. Der dann dort ergänzte, sprich um einen Eintrag vergrößerte, truststore ist natürlich in allen anderen Rechnern der Gruppe ebenfalls auszuwechseln.

Hinweis: Ein versionsneutrales, und damit immer gleiches, Installationsverzeichnis für das aktuelle JDK ist (vom Installer-Vorschlag abweichende) bewährte Praxis.

Hinweis: Ob der truststore im jeweiligen Rechner alle benötigten Einträge enthält, prüft man mit:

```
D:\>keytool -list -keystore
      C:\Programme\jdk\jre\lib\security\truststore
Geben Sie das Keystore-Passwort ein:*****
Keystore-Typ: JKS .....
```

Hinweis: Nach Einstieg / Erprobung etc. darf man den Zeitpunkt, zu dem man von den default-Passworten „changeit“ (!) weg muss, nicht verpassen.-

3. Ein einheitlicher Satz von Steuerdateien

3.1 Start und JMX-management.properties

Um die JVM (java.exe) und die in ihr gestartete Anwendung via JMX un SSL bedien- und beobachtbar (B&B) zu machen, sind beim Start mindestens die system properties

- com.sun.management.config.file
- javax.net.ssl.keyStore
- javax.net.ssl.keyStorePassword

geeignet zu setzen. Aus

```
java WinAppDemo -verbose -en
```

wird also dann etwas wie

```
java -Djavax..... WinAppDemo -verbose -en
```

Diese Komplikation versteckt man am besten in einem Script jmx.bat, welches in

```
C:\bat\
```

auf jedem Rechner bereitgestellt wird.

Wenn C:\bat\ vorne im Pfad (path) ist, ersetzt man man nun einfach

```
java WinAppDemo -verbose -en
```

durch

```
jmx WinAppDemo -verbose -en
```

und man hat die Anwendung sicher JMX-bedienbar gestartet; Listing 4.

```
@Echo.
@Echo jmx.bat V01.04 (21.06.2007) (c) A. Weinert
@Echo Starte java.exe mit JMX
@Echo.
java.exe -Dcom.sun.management.config.file="C:\Programme\jdk\
jre\lib\management\%username%-management.properties"
-Djavax.net.ssl.keyStore=
"C:\Programme\jdk\jre\lib\security\%computename%-keystore"
-Djavax.net.ssl.keyStorePassword=changeit %*

@REM w/o SSL use:
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false

@Echo.
```

Listing 4: JVM- und Anwendungsstart

und man hat die Anwendung sicher JMX-bedienbar gestartet; Listing 4.

Nun sind noch einige properties der Form

```
com.sun.management.dieseUndJeneEigenschaft
```

zu bestimmen. Dies könnte mit weiteren Optionen der Form „D-com.sun.manag.....“ im Startkommando von java.exe bzw. javaw.exe geschehen. Dies würde dadurch „ellenlang“.

Glücklicherweise kann das Setzen dieser Eigenschaften auch in eine properties-Datei auslagern. Das Start-Script jmx.bat (Listing 4) bestimmt diese Datei benutzerspezifisch als

```
C:\Programme\jdk\jre\lib\
management\%username%-management.properties
```

Ist der Benutzer „wichtel“ angemeldet (Beispiel), werden die Einstellungen also aus einer Datei

```
C:\..\jre\lib\management\wichtel-management.properties
```

geholt (Listing 5).

```
#####
# Wichtel Configuration File for Java Platform Management
#####

# Management Config. file for user Wichtel 16.05.2007 10:43
# V01.01 18.06.2007: aufs Wesentliche (JMX via SSL) reduziert
#Einstellungen können überfahren werden mit:
#      java[w] -Dproperty=andererWert ...

com.sun.management.jmxremote.port=9886

com.sun.management.snmp.acl=true
#      Default for this property is true.

com.sun.management.jmxremote.ssl=true
#      Default for this property is true.

com.sun.management.jmxremote.ssl.need.client.auth=false
#      Default for this property is false.

com.sun.management.jmxremote.registry.ssl=false
#      Default for this property is false.

com.sun.management.jmxremote.authenticate=true
#      Default for this property is true.

com.sun.management.jmxremote.password.file=C:\\programme\\jdk\
\jre\\lib\\management\\wichtel-jmxremote.password

com.sun.management.jmxremote.access.file=C:\\programme\\jdk\\j
re\\lib\\management\\jmxremote.access

#-----
```

Listing 5: Nutzerspezifische JMX-Eigenschaften (wichtel als Beispiel)

3. Rollen und Passworte

Das Bedienen und Beobachten einer so gestarteten Anwendungen und JVM via JMX und SSL ermöglicht einige auch ggf. sicherheits- und verfügbarkeitskritische Abfragen und Eingriffe. Die B&B-Anwendung bzw. der Anwender (auf dem Client-Rechner) sollten sich zusätzlich authentifizieren.

Defaultmäßig sieht SUN hierfür Namen bzw. Rollen und Passwortverfahren vor. Die Namen bzw. Rollen werden in einer Datei definiert, welche Listing 5 als

```
C:\programme\jdk\jre\lib\management\jmxremote.access
```

bestimmt. Der Inhalt einer solchen Datei kann so einfach sein, wie Listing 6 zeigt:

```
# Default access control entries:
# o The "monitorRole" role has readonly access.
# o The "controlRole" role has readwrite access.

monitorRole    readonly
controlRole    readwrite
```

Listing 6: Rollendefinition; jmxremote.access (SUNs default-Werte)

In dieser Datei werden beliebig viele Rollen (=JMX-Log-in-Nutzernamen) bestimmt; im Beispiel sind es „monitorRole“ und „controlRole“. In dieser Datei werden diesen Nutzernamen lediglich ihre Fähigkeiten zugewiesen. Man kann hier nur grob zwei Fälle unterscheiden:

- readonly (nur beobachten; „B“) und
- readwrite (bedienen und beobachten: „B&B“)

Welche der (möglicherweise vielen) Namen = Rollen mit welchem Passwort nutzbar sind, wird in einer Passwortdatei bestimmt, welche (SUN will es so) zu dem Nutzer gehört, der die zu bedienende / zu beobachtende JVM und Anwendung gestartet hat.

Ist dies (in unserem Beispiel) der Nutzer „wichtel“, so hat die einheitliche Startdatei (jmx.bat; Listing 4) diesem Nutzer die nutzerspezifische Eigenschaftsdatei (wichtel-management.properties; Listing 5) zugewiesen. Diese nun verweist ihrerseits auf die nutzerspezifische Passwortdatei wichtel-jmxremote.password; Listing 7:

```
#-----
#           jmxremote.password (wichtel) 16.05.2007 09:56
#
# 23.02.2007 16:40: v00.00 für Test
# 15.06.2007 14:49 v00.01: umbenannt - wird _
#
monitorRole    QED
controlRole    R&D
```

Listing 7: Nutzerspezifische Passwortdatei (für „wichtel“ als Beispiel)

Diese (also wichtels) Passwortdatei lässt für alle unter diesem Nutzer gestarteten JVMs bzw. Applikationen JMX-B&B in beiden Rollen (vgl. Listing 6) und weist diesen die Passworte „QED“ bzw. „R&D“ (SUN-Beispiel; i.A. nicht so lassen).

3.2 Sun's Handhabung der Passwortdateisicherheit

Der Ansatz B&B-Nutzernamen bzw. -Rollen in beliebiger Anzahl (aber nur zwei Ausprägungen readonly und readwrite) zu definieren und diesen dann abhängig vom Konto, das die JVM / Anwendung startete, Passworte zuzuweisen erscheint nicht direkt sinnfälliger oder sehr gewöhnungsbedürftig (gelinde ausgedrückt). Außerdem ist er nicht an eh vorhandene Nutzer und Rollen (=Gruppen) beispielsweise in einer Domain (active directory) anschließbar.

Dieser (seltsame) SUN-Ansatz heißt auch, dass dieselbe Anwendung auf demselben Server andere Passworte für die remote JMX-B&B erfordern kann, nur wenn sie von einem anderen Nutzer-account gestartet wurde. (Für kritische Anwendungen werden die startenden Nutzer „system“ oder solche mit besonderen administrativen Rechten sein.) Da unterschiedliche JMX-Passworte je nach startendem Nutzer-account i.A. zu Verwirrung und schwer lokalisierbaren Fehlfunktionen führen würde, wird man i.A. für alle Nutzerkonten, die B&B-bare (Automatisierungs-) Anwendungen starten, dasselbe Passwort-File verwenden wollen.

Letzteres aber wird durch SUNs Ansatz zum Schutz des Passwort-Files verhindert: Dieser passt leider schlecht zu den meisten (Windows-Administrator-) Gewohnheiten bezüglich der Vergabe von ACLs und erwies sich als sehr mühsam in der Praxis.

SUNs merkwürdige Bedingungen für das Passwort-File sind:

1. Read access für den JVM-Benutzer (den account der die JVM / Anwendung auf dem Server startet). read access reicht, Vollzugriff für diesen Nutzer geht aber auch.
2. Keine (null, 0, nada) irgendwelchen anderen Zugriffsrechte (access) für irgendwelche anderen Nutzer. Und dies schließt geerbte Rechte und auch Rechte für Administratoren hart aus.
3. Die Passwortdatei muss im Besitz des o.g. JVM-Benutzers sein.

Insbesondere der Punkt 2 macht alle möglichen Wartungsarbeiten, die dieses auch nur indirekt betreffen, zu einer „Freude“ für Administratoren. Ist aber nur eine der Bedingungen verletzt, wird das Passwortfile ignoriert, und eine JMX-remote-B&B verhindert.

Bedingung 1. und 2. lässt sich mit etwas zwischen

```
cacls
"C:\Programme\jdk\jre\lib\management\garcon-jmxremote.password"
/P garcon:F
Sind Sie sicher (J/N)?j
```

für Vollzugriff bzw. mit

```
cacls
"C:\Programme\jdk\jre\lib\management\garcon-jmxremote.password"
/P garcon:R
Sind Sie sicher (J/N)?j
```

für nur Lesen erreichen.

Für das Besitzersetzen gibt es leider kein Standard-Kommandozeilen-Tool, hierzu muss durch den Explorer geklickt werden (Sicherheit, erweitert, Besitzer). Desgleichen kann man bei cacls die lästige Abfrage nicht (oder vielleicht nur mit dem Eingabeumleitungs-

trick) umgehen. Also keine guten Voraussetzungen für die Automatisierung / Unterstützung der administrativen Arbeiten mit Scripts oder Tools.

Bei installierten Windows-Support-Tools wird die ganze Geschichte mit

```
"c:\programme\Support Tools\XcAcls.exe"  
"C:\Programme\jdk\jre\lib\management\garcon-jmxremote.password"  
/P garcon:FO /Y  
processed file: C:\Programme\jdk\jre\lib\management\garcon-  
jmxremote.password
```

automatisierbar.

Die unschöne lange Pfadangabe bei XcAcls wird man durch Aufnahme der Support-Tools in den Programmsuchpfad (PATH) los oder (viel besser) durch einfaches Kopieren von

```
24.03.2003 22:15          63.488      xcaccls.exe
```

in ein PATH-Verzeichnis wie

```
C:\programme\util\ .
```

Hinweis am Rande: Der Programmsuchpfad einer (vernünftigen) Windows-Installation — für auch Java-Entwicklung und -Anwendungen — sieht etwa so aus:

```
C:\bat;C:\Programme\util;C:\programme\jdk\bin;C:\WINDOWS\system32;  
C:\WINDOWS;C:\WINDOWS\System32\Wbem
```

Insbesondere ist er nicht wesentlich länger.

Der Passwortdateiansatz von SUN erfordert also nutzerspezifische Passwort- und indirekt dadurch auch nutzerspezifische Steuerdateien. In der Praxis wird man den Inhalt dieser Dateien für alle Nutzer-accounts, die JMX-B&B-bare (Automatisierungs-) Anwendungen starten dürfen i.A. gleich halten.

Anmerkung: Dass bei Windows Server 2003 enterprise edition diese „Support-Tools“ nicht standardmäßig dazugehören, war ja schon immer ärgerlich.

Hinweis: Letztlich wird man unter Windows / AD diese — vom Ansatz her immer unschöne — zusätzliche Nutzer-/Passwortdateiverwaltung durch systemgerechte Ansätze ersetzen wollen.

4. Eine mit JMX B&B-bare Java-Anwendung

Alle Anwendungen, die auf dem Framework `de.a_weinert` beruhen, sind optional JMX-bediener- und beobachtbar (B&B). Siehe dazu Einzelheiten im Anhang A1 und unter a-weinert.de/java/aweinertbib.html.

Wenn Sie eine kleine Server-Anwendung „MyServerApp“ als Erbe von [de.a_weinert.App](#) bzw. von [WinApp](#) geschrieben haben, nutzen Sie diese für das folgende Experiment.

Hinweis: Mit „Server“ ist hier lediglich gemeint, dass diese Anwendung im Dauerbetrieb oder zumindest einige Stunden läuft. Eine Anwendung, die nach 1 s fertig ist von einem anderen Rechner aus remote beobachten zu wollen, hat wenig Erfolgsaussichten.

Falls Sie keine eigene solche Anwendung haben nutzen Sie den Dateiaufdat- /aufräum- und Synchronisationsdienst `de.a_weinert.apps.FileServices`, dem Sie etwas Hin und Her zwischen ein paar Dummy-/temp- Verzeichnissen zu tun geben. Siehe Bild 2.

Für die JMX-Bedien- und -Beobachtbarkeit ist es entscheidend, die Serveranwendung nicht einfach mit

```
java MyApp -optionen parameter
```

zu starten, sondern mit

```
jmx MyApp -optionen parameter
```

also mit `jmx.bat` etwa gemäß Listing 4 (siehe Seite 7) nehmen können.

```
FileServices - Aufdat- und Nachführdienste (Java)
Datei Dienste Protokoll Ansicht
backup D:\FB3WWW\WWWBack\miro-lab\showit\styles\menues
... \ blackmetal.css          1063 16.01.2006 07:48:00 rw : deleted + backup
... \ bluedream.css           1229 16.01.2006 07:48:00 rw : deleted + backup
... \ bluesea.css             1062 16.01.2006 07:48:00 rw : deleted + backup
... \ bonanza.css             914 22.02.2006 11:33:20 rw : deleted + backup
... \ chiquita.css            1061 16.01.2006 07:48:00 rw : deleted + backup
... \ greyhound.css           927 16.01.2006 07:48:00 rw : deleted + backup
... \ heaven.css              1229 16.01.2006 07:48:00 rw : deleted + backup
... \ oldenglish.css          700 16.01.2006 07:48:00 rw : deleted + backup
... \ RedLobster.css          1063 16.01.2006 07:48:00 rw : deleted + backup
... \ sahara.css              1229 16.01.2006 07:48:00 rw : deleted + backup
... \ showit_standart.css     927 16.01.2006 07:48:00 rw : deleted + backup
... \ turkey.css              616 16.01.2006 07:48:00 rw : deleted + backup

Clean W:\studieninfo_\automation
backup D:\FB3WWW\WWWBack\studieninfo_\automation
... \ index.html              407 21.02.2006 13:47:49 rw : deleted + backup

*** 129371 / 1452 Dateien geändert ( Mo, 25.06.2007, 10:08:48 ... 11:06 ).
```

Bild 2: FileServices läuft im Dauerbetrieb als Serveranwendung.

5. B&B (remote, SSL) einer Java-Anwendung mit JConsole

Wir nehmen an, dass gemäß den vorangehend geschilderten Einstellungen und Festlegungen eine Serveranwendung „MyApp“ oder „Fileservices“ (zum Beispiel) auf dem Server PD310S mit dem Script `jmx.bat` (Listing 8 im letzten Kapitel) gestartet wurde.

Damit ist diese Anwendung via JMX und SSL (sicher) remote bedien- und beobachtbar. Mit dem Standard-JDK-Werkzeug JConsole kann dies vergleichsweise einfach getestet werden. Bei entsprechen geschickter Gestaltung der Anwendung und ihrer Objekte als DynamicMBean liefert JConsole auch eine für manche Fälle hinreichende B&B-Station.

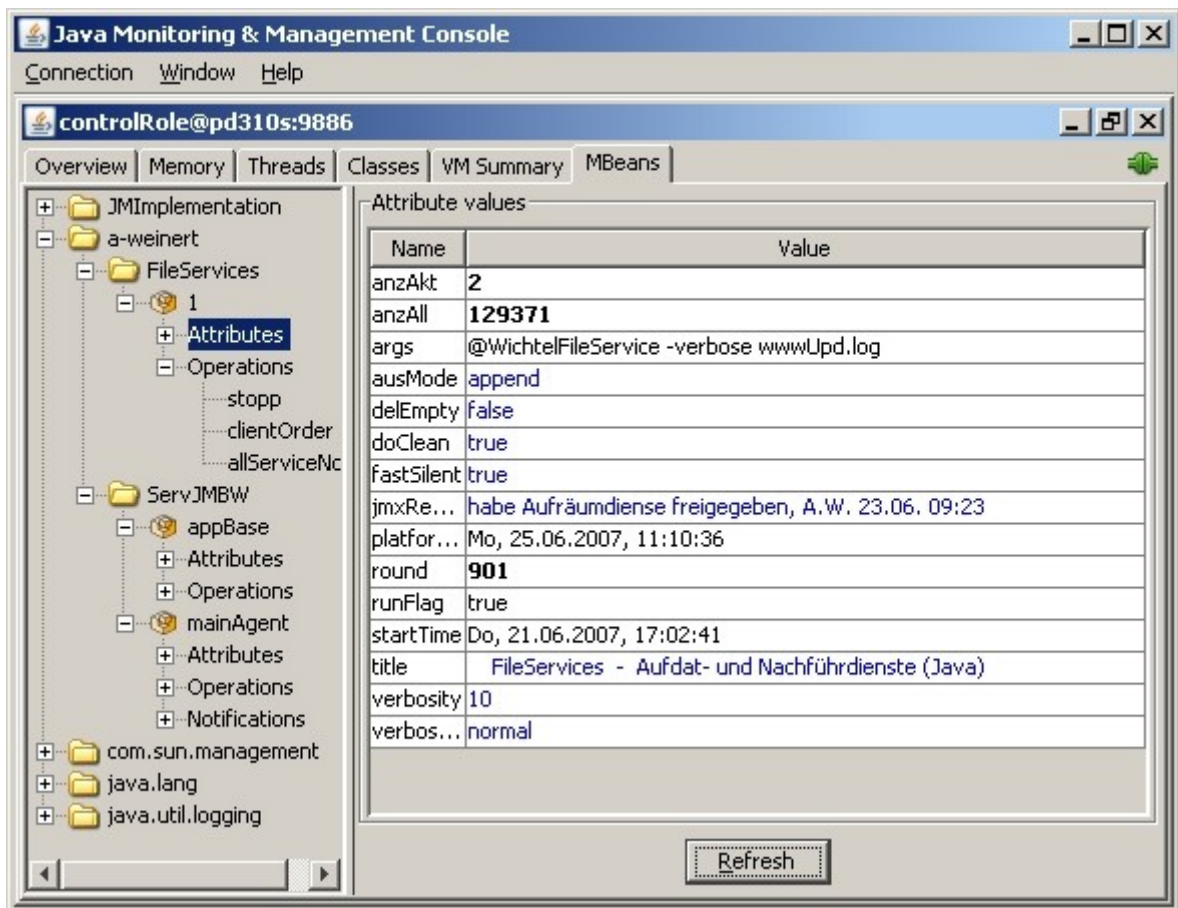


Bild 3: FileServices remote B&B mit JConsole.

Nach dem entsprechenden Start der JConsole auf einem anderen Beobachtungs-Rechner der Gruppe kann man unter anderem einige Eigenschaften (Attribute) beobachten und teilweise (die blauen) auch direkt bedienen; vgl. Bild 3. Dies geschieht erst nach entsprechender Authentifizierung und via SSL.

Damit das alles gemäß den im Vorausgehenden genannten Ansprüchen und Einstellungen funktioniert, darf man die JConsole nicht direkt starten. Am Besten verwendet man ein Script, wie `jmxConsole.bat`; Listing 9. Bei direktem Start ohne die entsprechenden Parameter würde die (SSL-) Verbindung nicht funktionieren.

```
@Echo.
@Echo jmxConsole.bat Starte jConsole.exe %* mit JMX
@Echo V01.04 (01.03.2007) (c) A. Weinert
@Echo.

@Echo @jconsole -J-
  Djavax.net.ssl.trustStore=C:\...,trustStorePassword=... %*
@jconsole.exe -J-
  Djavax.net.ssl.trustStore="C:\Programme\jdk\jre\lib\security
  \truststore" -J-Djavax.net.ssl.trustStorePassword=changeit
  %*
@Echo.
```

Listing 8: SSL-fähiger Start der JConsole; `jmxConsole.bat`

Nach Start der JConsole (mit Listing 8) erscheint ein Anmeldefenster; vgl. Bild 4. Dort trägt man die (remote-) Verbindungsdaten (PD310S:9886 im Beispiel) und die Authentifizierung (controlRole, R&D im Beispiel) ein. Dies führt im Erfolgsfalle und nach etwas Navigation durch die „MBeans“ zur B&B der Anwendung; vgl. Bild 3.

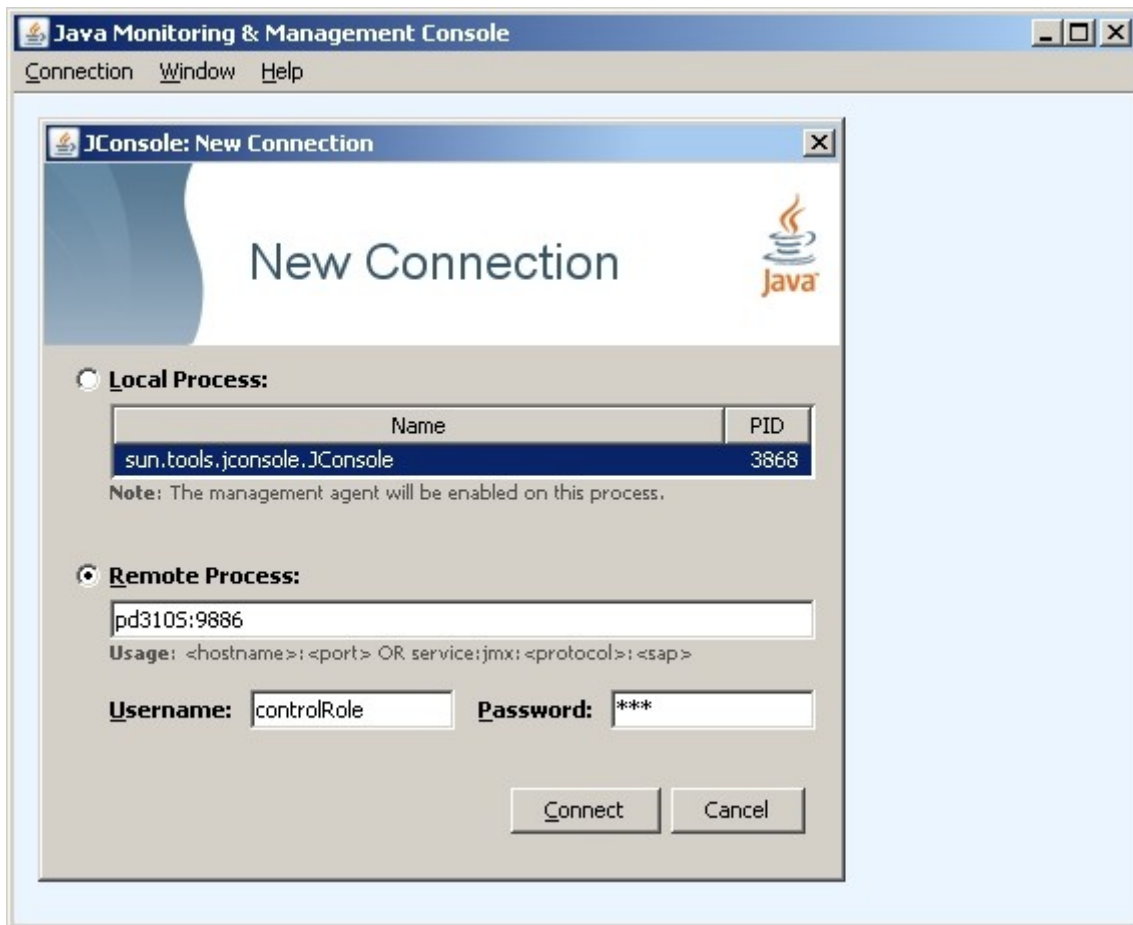


Bild 4: JConsole, remote-Anmeldung an PD310S

6. Kopplung (remote, SSL) von Java-Anwendungen

Was mit der JConsole geht (B&B), geht auch mit einer entsprechenden JMX- (Client-) Anwendung.

todo

7. Was noch zu tun bleibt

todo; Stichworte:

JMX-Client.

Client-Auth: # com.sun.management.jmxremote.ssl.need.client.auth=true|false

8. Résumé

JMX ist der SUN-Standard für die Kopplung und die Bedienung und Beobachtung von verteilten Java-Anwendungen und Objekten, die als manageable Beans gestaltet werden müssen.

Die Hürde dieser Beans-Programmierung fällt dem mit OO, Pattern und Komponentenmodellen vertrauten Java-Programmierer nicht so schwer, insbesondere mit Unterstützung durch das Framework de.a_weinert (a-weinert.de/java/aweinertbib.html).

Die sehr komfortable Erzeugung von MBeanInfo-Objekten aus einer XML-Beschreibung (de.a_weinert.net.JmxHelper.makeMBeanInfo() u.A.) ist da nur eine der vielen Hilfen.

Die andere (oft eigentliche) Hürde ist die Handhabung der sicheren Kommunikation für JMX unter allen an der verteilten Anwendung beteiligten (Windows-) Rechnern. Diese Anforderung ist nur abgeschwächt, wenn alle beteiligten Rechner der Automatisierungsanwendung an einem nur (exklusiv) dafür verwendeten (privaten) LAN hängen und es dazu keinen Zugang von einem allgemein verwendeten (Büro- / Firmen-) LAN aus gibt.

Aus vielen guten Gründen wird man dies aber genau so nicht machen; die häufigsten Gründe sind Kosten, unkompliziertes Nutzen vorhandener Netzwerke als Ressourcen und vor Allem frei verteilte (remote) B&B.

Wenn dies so ist, braucht man die Verbindung mittels JMX einer Gruppe von Automatisierungs- und B&B-Rechnern in einem recht offenen LAN eine sichere Kommunikation. Wenn Sie das vorliegende Tutorial erfolgreich nachvollzogen haben, haben Sie den Einstieg dazu geschafft — nicht mehr, aber auch nicht weniger.

A. Anhang

A1. JMX im Framework

Das Framework `de.a_weinert` (a-weinert.de/java/aweinertbib.html) unterstützt unter vielem Anderen mehr auch JMX:

1. Alle Anwendungen, die auf dem Framework `de.a_weinert` beruhen, sind optional JMX-bedien- und beobachtbar.
2. Die etwas umständliche Erstellung (des „Objektgebieges“) von `MBeanInfo` wird auf die Erstellung einer sinnfälligen XML-Beschreibung reduziert.
3. `de.a_weinert.App`-Erben suchen und finden diese xml-Beschreibung unter dem Namen `<Anwendung>MBean.xml` als Datei oder Ressource (aufwärts) entlang der Vererbungslinie. Gefunden wird dabei „spätestens“ die im nächsten Abschnitt gelistete Beschreibung von `App` als `DynamicMBean`.

Diese XML-MBean-Beschreibung (Funktion 2) steht allen Anwendungen über die Klassen `de.a_weinert.net.JmxHelper` bzw. `JmxHelper.MBeanInfoFactory` zur Verfügung.

Für die B&B an sich (Funktion 1) müssen `de.a_weinert.App`-Erben im Rahmen ihrer Initialisierung lediglich die Methode `registerAsMBean(name, erg)` aufrufen. Falls gewünscht, sprich falls die für `App` vorgegebene Beschreibung nicht ganz passt, ist noch eine ergänzte/ individualisierte Datei `<Anwendung>MBean.xml` im selben Verzeichnis / jar-File wie die Anwendung bereitzustellen. Mehr ist nicht zu tun.

A1.1 `de.a_weinert.App` als `DynamicMBean` — mit XML

```
@Echo.  
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!-- MBeanInfo Beschreibung / MBeanInfo description  
      Auch als Beispiel      / is also an example  
  
      Copyright 2007 Albrecht Weinert (a-weinert.de)  
  
      Beschreibung der Elternklasse aller Framework-Anwendungen  
      als DynamicMBean. Bei Verwendung der Framework-Methoden  
      zur MBean-Registrierung und MBeanInfo-Generierung wird  
      diese Beschreibung vererbt.  
-->  
<MBeanInfo className="de.a_weinert.App"  
  version="$Revision: 1.4 $"  
  date="$Date: 2007/05/23 14:12:29 $"  
  application="App (super class)" >  
  <!-- dtd: className ist erforderlich -->  
  <!-- dtd: text is description
```

```

    Text ist Beschreibung. Mehrere Textstücke werden
    concateniert.
    Der ermittelte Text wird von umfassenden white
    space befreit.
    Gilt für alle Tags / applies also to all (sub-) tags.
-->
de.a_weinert.App - based Java-Application

<MBeanOperationInfo name="stop" impact="ACTION"
                    returnType="java.lang.String">
    Stoppt die Anwendung.
    <!-- dtd: 0..n MBeanParameterInfo , same as for
Constructors -->
</MBeanOperationInfo>

<MBeanOperationInfo name="clientOrder" impact="ACTION"
                    returnType="java.lang.String">
    clientOrder(command): command to the App-Object.
    <MBeanParameterInfo name="command"
                        type="java.lang.String" >
        the (specific) command
    </MBeanParameterInfo>
    <!-- dtd: 0..n MBeanParameterInfo , same as for
Constructors -->
</MBeanOperationInfo>

<MBeanConstructorInfo name="App">
    parameterloser Konstruktor;
    Mit go(...) wird die Anwendung gestartet.
    App is abstract; do not use; Test-Example only.
    <!-- dtd: 0..n MBeanParameterInfo , same as for
Operations -->
</MBeanConstructorInfo>

<MBeanAttributeInfo name="platformTime"
attributeType="java.lang.String"
                    isReadable="true" >
    aktuelle Zeit der Plattform
</MBeanAttributeInfo>

<MBeanAttributeInfo name="title"
attributeType="java.lang.String"

```

```

    isWritable="true" isReadable="true" >
      (Fenster-) Titel der Anwendung
    </MBeanAttributeInfo>

    <MBeanAttributeInfo name="startTime"
attributeType="java.lang.String"
    isReadable="true">
      Startzeit der Anwendung
    </MBeanAttributeInfo>

    <MBeanAttributeInfo name="args"
attributeType="java.lang.String"
    isReadable="true">
      Startparameter
    </MBeanAttributeInfo>

    <MBeanAttributeInfo name="verbosity" attributeType="int"
    isWritable="true" isReadable="true">
      Ausführlichkeits-Level
    </MBeanAttributeInfo>

    <MBeanAttributeInfo name="verbosityString"
    attributeType="java.lang.String" isWritable="true"
isReadable="true">
      Ausführlichkeits-Level
    </MBeanAttributeInfo>

    <MBeanAttributeInfo name="ausMode"
attributeType="java.lang.String"
    isWritable="true" isReadable="true">
      Öffnungsmodus für vorhandene Dateien
    </MBeanAttributeInfo>

    <MBeanAttributeInfo name="runFlag"
attributeType="boolean"
    isReadable="true">
      Anwendung läuft / soll laufen
    </MBeanAttributeInfo>

    <MBeanNotificationInfo name="javax.management.Notification"
    >
      Gewichtige Änderungen des Applications-Zustands

```

```

    <!-- dtd: 0..n notifTypes -->
    <notifType className="jmx.app.change" />
</MBeanNotificationInfo>

<MBeanAttributeInfo    name="jmxRemoteNote"
    attributeType="java.lang.String" isWritable="true"
isReadable="true">
    Kurznotiz eines JMX Bedieners / Beobachters (50 Zeichen)
</MBeanAttributeInfo>

</MBeanInfo>

```

Listing A1: XML-Beschreibung von de.a_weinert.App als DynamicMBean

A1.2 de.a_weinert.AppBase als DynamicMBean — mit XML

```

@Echo.
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- MBeanInfo Beschreibung / MBeanInfo description
AppBaseMBean.xml
Copyright 2007 Albrecht Weinert (a-weinert.de)

Beschreibung der Basisdienste für Framework- und andere
Anwendungen als DynamicMBean.
-->
<MBeanInfo className="de.a_weinert.AppBase"
    version=" $Revision: 1.1 $"
    date=" $Date: 2007/04/26 07:08:21 $" >
    de.a_weinert.AppBase - based Java-Application

    <MBeanOperationInfo name="stoppSig" impact="ACTION"
        returnType="java.lang.String">
        Stopp-Signal für alle Anwendungen (und für die
        JVM Achtung!).
    </MBeanOperationInfo>

    <MBeanOperationInfo name="stoppBaseApp" impact="ACTION"
        returnType="java.lang.String">
        Stoppt die Basisanwendung (und die JVM Achtung!).
    </MBeanOperationInfo>

```

```

<MBeanOperationInfo name="exec" impact="ACTION"
    returnType="int">
    <MBeanParameterInfo name="command"
        type="java.lang.String" >
        Anwendung [Parameter ...
    </MBeanParameterInfo>
    Startet eine Anwendung im Thread des Aufrufers
</MBeanOperationInfo>

<MBeanOperationInfo name="start" impact="ACTION"
    returnType="java.lang.String">
    <MBeanParameterInfo name="command"
        type="java.lang.String" >
        Anwendung [Parameter ...
    </MBeanParameterInfo>
    Startet eine Anwendung in einem eigenen Thread
</MBeanOperationInfo>

<MBeanOperationInfo name="testExec" impact="ACTION"
    returnType="int">
    <MBeanParameterInfo name="command"
type="java.lang.String" >
        Anwendung [Parameter ...
    </MBeanParameterInfo>
    Testet die Startbarkeit einer Anwendung
</MBeanOperationInfo>

<MBeanAttributeInfo name="platformTime"
attributeType="java.lang.String"
    isReadable="true" >
    aktuelle Zeit der Plattform
</MBeanAttributeInfo>

<MBeanAttributeInfo name="title"
attributeType="java.lang.String"
    isReadable="true" >
    Titel von AppBase
</MBeanAttributeInfo>

<MBeanAttributeInfo name="commonRun"
attributeType="java.lang.String"
    isReadable="true">

```

```

    Alle Anwendungen dürfen laufen
</MBeanAttributeInfo>

<MBeanAttributeInfo name="killJVMDelay" attributeType="int"
    isReadable="true">
    Zeitverzögerung Ende letzte app und JVM / ms
</MBeanAttributeInfo>

<MBeanAttributeInfo name="isBaseWinApp"
attributeType="java.lang.String"
    isReadable="true">
    Basis-Anwendung ist de.a_weinert.WinApp-Erbe
</MBeanAttributeInfo>

<MBeanAttributeInfo name="isBaseApp"
attributeType="java.lang.String"
    isReadable="true">
    Basis-Anwendung ist de.a_weinert.App-Erbe
</MBeanAttributeInfo>

<MBeanAttributeInfo name="baseApp"
attributeType="java.lang.String"
    isReadable="true">
    Basis-Anwendung (toString)
</MBeanAttributeInfo>

</MBeanInfo>

```

Listing A2: XML-Beschreibung von de.a_weinert.AppBase als DynamicMBean

A2. Abkürzungen

| | |
|-------|---|
| ACL | access control list (Liste mit Zugriffsrechten auf ein Objekt) |
| AD | Active Directory (Microsofts Interpretation von LDAP) |
| AJAX | Asynchronous JavaScript + XML |
| API | Application Programme Interface |
| BuB | Bedienen und Beobachten (von Prozessen) |
| C/S | Client-Server |
| CA | Certification authority |
| FAQ | Frequently Asked Questions (Hilfetexte in Frage-Antwort-Form) |
| FB | Fachbereich; insbesondere ... |
| FB3 | Fachbereich Elektrotechnik und Informatik der FH Bochum |
| FH | Fachhochschule |
| GSS | Generic Security Service |
| GUID | Globally Unique Identifier |
| GWT | Google Webtoolkit, AJAX mit nur Java |
| HTML | Hypertext Markup Language [RFC 1866] |
| HTTP | Hypertext Transfer Protokoll. Internet-Protokoll zur Übertragung von Seiten. |
| HTTPS | HTTP über SSL. Abgesicherte Übertragung. |
| HW | Hardware |
| IIOp | Internet Inter-ORB Protocol |
| IP | Internet Protocol |
| J2EE | Java 2 Enterprise Edition |
| J2ME | Java 2 Micro Edition |
| J2SE | Java 2 Standard Edition |
| JAAS | Java Authentication and Authorization Service |
| JAF | JavaBeans Activation Framework |
| JAR | Java Archive. (.zip + Semantik) |
| JAXP | Java API for XML Parsing |
| JCA | Java Cryptography Architecture (der Java Security API) |

| | |
|-------|--|
| JCE | Java Cryptography Extensions (zur JCA, Exportrestriktion wegen DAS, DES) |
| JDBC | Java Database Connectivity (Java Datenbankanschluss) |
| JDC | Java Developer Connection (Ein WWW-Service) |
| JDK | Java Development Kit; der Werkzeugsatz für die Entwicklung mit Java |
| JEB | Enterprise JavaBeans (ungleich JavaBeans) |
| JMX | Java Management Extensions |
| JNDI | Java Naming and Directory services Interface |
| JNI | Java Native Interface |
| JRE | Java Runtime Environment; JDK-Subset ohne Entwicklungswerkzeuge. |
| JSDK | Java Servlet Development Kit |
| JSF | Java Server Faces |
| JSP | Java Server Pages |
| JSSE | Java Secure Socket Extension (seit JDK1.4.x integriert) |
| JSTL | JavaServer Pages Standard Tag Library |
| JVM | Java virtual machine; der eigens für Java erfundene Prozessor. Er wird im Allgemeinen auf dem jeweiligen Zielsystem emuliert. |
| LAN | Local area network; Datennetz für mittlere Entfernungen |
| LDAP | Lightweight Directory Access Protocol |
| LGPL | Lesser GNU Public License |
| MBean | Managed Bean (JMX) |
| MEVA | Labor für Medien und verteilte Anwendungen |
| MS | Microsoft |
| NT | Betriebssystem Windows NT (MS) |
| OMG | Object Management Group |
| OS | Operating System |
| PAM | Pluggable Authentication Module |
| PC | Personal Computer |
| R&D | Research and Development |
| RAID | Redundant Array of inexpensive Disks |
| RDF | Resource Description Framework (W3C) |
| RMI | Remote Method Invocation |
| RPC | Remote Procedure Call |

| | |
|------|---|
| SAAJ | SOAP with Attachments API for Java |
| SMTP | Simple Mail Transfer Protocol |
| SOAP | Simple Object Access Protocol |
| SQL | Structured query language, Datenbankbearbeitungssprache |
| SSL | Secure Socket Layer. Protokollschicht zur Absicherung. |
| SSO | Single Sign on; Authentifizierung vieler (n) Anwendungen gegen eine (1) "security realm". |
| TCP | Transmission Control Protocol |
| TM | Trade Mark (Warenzeichen) |
| UML | Unified Modelling Language |
| URI | Uniform Resource Locator |
| W2K | Betriebssystem Windows 2000 (MS) |
| W2K3 | Betriebssystem Windows Server 2003 (MS) |
| W3 | Amerikanische Kurzform für WWW |
| W3C | World Wide Web Consortium |
| WS | Workstation |
| WSDL | Web Services Description Language |
| XML | eXtensible Markup Language |

A3. Literatur

- [1] Ed Ort and Mark Basler, AJAX Design Strategies, SUN 2006
<http://java.sun.com/developer/technicalArticles/J2EE/AJAX/.../design-strategies.pdf>
- [2] Brett McLaughlin, Mastering Ajax, Part 1..4, IBM, 2005
<http://www-128.ibm.com/developerworks/web/library/wa-ajaxintro.html>
- [3] Albrecht Weinert, Zur Installation des JDK (Java Development Kit)
<http://a-weinert.de/weinert/pub/java-install.txt>
- [4] Albrecht Weinert, Java — Tipps und Tricks
<http://a-weinert.de/weinert/pub/java-tips.txt>
- [5] Albrecht Weinert, AJAX mit GWT — Tipps und Tricks
<http://a-weinert.de/weinert/pub/gwt-tips.pdf>
- [6] Albrecht Weinert, Tipps zu CVS für Windows — cvsNT
<http://a-weinert.de/weinert/pub/cvsnt-tipp.txt>
- [7] Google, Web-Toolkit, online-Dokumentation (nicht am Stück verfügbar)
<http://code.google.com/webtoolkit/documentation/>.
- <8> Albrecht Weinert, Tipps zu JMX mit SSL
<http://a-weinert.de/weinert/pub/jmx-ssl-tips.pdf>
- [9] Albrecht Weinert, Windows 2003 Domain Migration von NT4 mit Fremd-DNS
<http://www.a-weinert.de/weinert/pub/w2k3domain.pdf>
- [10] Albrecht Weinert, Windows Server 2003 — Domain FB3-MEVA Schulungsräume und Infrastruktur — Renovierung 2007
<http://www.a-weinert.de/weinert/pub/fb3-meva-domain2007.pdf>
- [11] Albrecht Weinert, Tipps zu Tomcat (5.x für Windows) ersetzt durch [13] ([13] stattdessen für Tomcat >= 6) <http://a-weinert.de/weinert/pub/tomcat-tips.pdf>
- [12] Albrecht Weinert, Windows Server 2003 — Domain FB3-MEVA Workstations und Server — Renovierung 2007
<http://www.a-weinert.de/weinert/pub/fb3-meva-workst2007.pdf>
- [13] Albrecht Weinert, Tomcat — mit Windows und Active Directory (ersetzt [11] als Nachfolger) <http://www.a-weinert.de/weinert/pub/tomcat-win-ad.pdf>
- [14] Albrecht Weinert, Tipps zu MySQL (mit Java, für Windows) (2006) <http://a-weinert.de/weinert/pub/mysqjawi-tipp.txt>

Hinweis: Aus Dateien „.../docu/*.txt“ könnten inzwischen teilweise „.../weinert/pub/*.pdf“ geworden sein.