

IBM System z9 and zSeries

# Automatisierung für z/VSE und andere Plattformen über Open Source Ant

Jörg Schmidbauer / Ingo Franzki



# Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and / or other counties.

CICS*	IBM*	Virtual Image
DB2*	IBM logo*	Facility
DB2 Connect	IMS	VM/ESA*
DB2 Universal Database	Intelligent Miner	VSE/ESA
e-business logo*	Multiprise*	VisualAge*
Enterprise Storage Server	MQSeries*	VTAM*
HiperSockets	OS/390*	WebSphere*
	S/390*	xSeries
	SNAP/SHOT	z/Architecture
	*	z/VM
		z/VSE
		zSeries

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

LINUX is a registered trademark of Linus Torvalds

Tivoli is a trademark of Tivoli Systems Inc.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

Intel is a registered trademark of Intel Corporation.

# Automatisierung von Vorgängen im VSE

## § Welche Möglichkeiten gibt es heute?

- JCL
- REXX
- JConVSE
- OEM Software

## § Eine neue Möglichkeit:

- Apache Ant

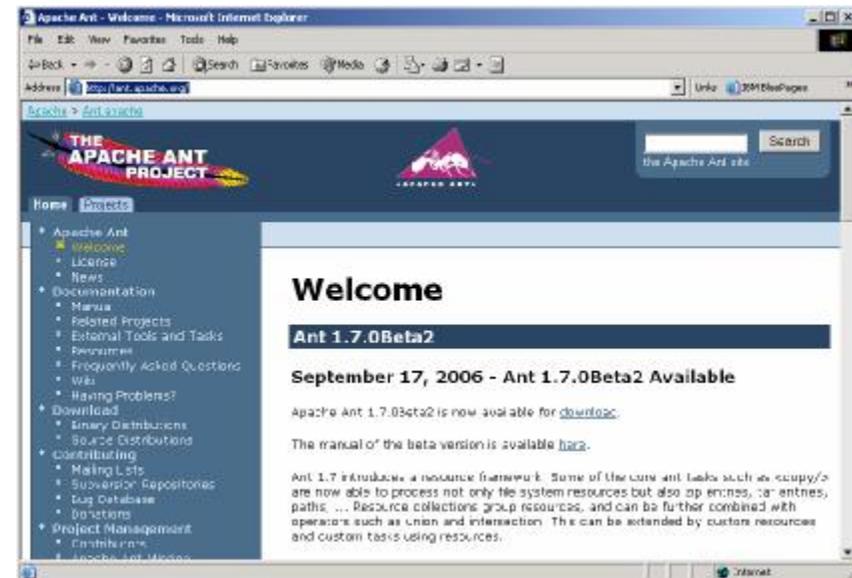


## Voraussetzungen

	Host-seitig	PC-seitig
<b>JCL</b>	Alles da	-
<b>REXX</b>	REXX (VSE oder VM)b	-
<b>JConVSE</b>	TCP/IP, VSE Connector Server	VSE Connector Client, JConVSE
<b>OEM Software</b>	Je nach dem ...	Je nach dem ...
<b>Apache Ant</b>	TCP/IP , VSE Connector Server	VSE Connector Client, Apache Ant
...	...	...

## Was ist Apache Ant

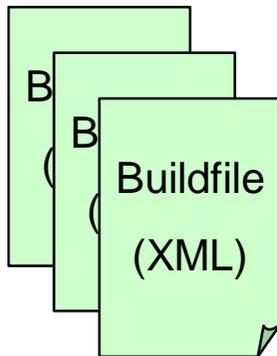
- § Apache Ant ist ein in Java geschriebenes Open-Source Build-Tool, ähnlich Make.
- § Ursprünglich zum automatisierten Build (Compile) von Java Programmen gedacht
- § Ant stellt Java-Klassen (Tasks) zur Verfügung um verschiedenste Dinge zu automatisieren.
- § Build-Scripte werden in XML geschrieben
- § Web Seite:
  - <http://ant.apache.org/>



## Wie funktioniert Ant?

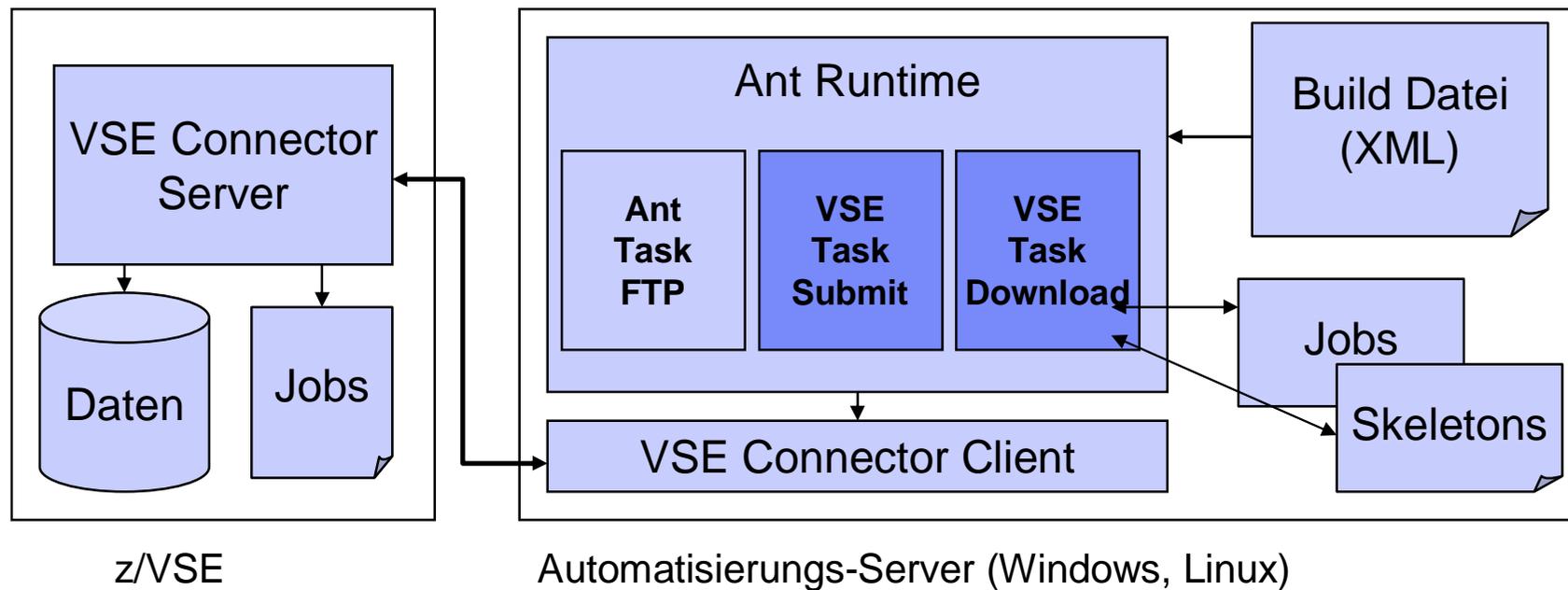


- § Ant stellt Funktionen als sog. Ant-Tasks zur Verfügung
- § Abläufe werden in Form von XML Build-Scripts angegeben, in denen Ant-Tasks verwendet werden.
- § Ant-Tasks sind jeweils Java-Klassen
- § Es gibt die Möglichkeit, eigene Ant-Tasks zu schreiben.



## Idee: Anbindung von Ant an VSE

- § Verwendung des Java-basierten VSE Connectors zur netzwerkmäßigen Anbindung eines PCs an VSE
- § Implementierung einer VSE-spezifischen Klassenbibliothek von Ant-Tasks um VSE Abläufe über Ant steuern zu können.



## Beispiel eines Ant Build-Scripts

```

<?xml version="1.0"?>
<project name="Project1" default="main" basedir=".">
  ...
  <taskdef name="submit"
           classname="com.ibm.vse.ant.Submit"
           classpath="vseant.jar"/>
  ...
  <target name="startmsg">
    <echo>Starting build of ${Name}</echo>
  </target>
  ...
  <target name="dowork" depends="startmsg">
    <submit jobfile="${tempdir}/test.job"
            waitforoutput="true"
            vsesystem="${vsesystem}"/>
  </target>
  ...
  <target name="endmsg" depends="dowork">
    <echo>Finished build of ${Name}</echo>
  </target>
  ...
  <target name="main" description="runs this script">
    <tstamp/>
    <antcall target="endmsg"/>
  </target>
  ...
</project>

```

1

§ Ein **Project** enthält mehrere **Targets**

2

§ Ein **Target** kann von anderen **Targets** abhängen

3

§ Innerhalb eines **Targets** können **Tasks** aufgerufen werden

§ Selbstgeschriebene **Tasks** müssen per **taskdef** definiert werden

## Ant Properties (Variablen)

- § Im Ant Script können Properties spezifiziert werden
- § Variablen Werte können über `${variable}` abgefragt werden, z.B.

```
<property name="tcpip.partition" value="F7"/>  
.  
.  
.  
partition="${tcpip.partition}"
```

- § Tasks können Variablen setzen (als Ergebnis) oder lesen
- § Ablauf kann abhängig vom Wert einer Variablen verzweigen oder beendet werden,
  - Bsp. Waitfor, Fail, If, While, Loop

## Bsp. für Verzweigung aufgrund Job-RC

```
<submit jobfile="c:\vsejobs\define_vtape.job"
    waitforoutput="true"
    outfile="c:\vsejobs\output\definetape.txt"
    propertyprefix="definetape.job"
    vsesystem="TESTVSE1" />

<condition property="definetape.failed">
    <not>
        <equals arg1="{definetape.job.maxrc}"
            arg2="0000" />
    </not>
</condition>

<fail if="definetape.failed"
    message="Define VTAPE on SCSI disk failed with
        MAXRC={definetape.job.maxrc}." />
```

## Standard ANT Tasks (Core & Optional Tasks)

- § **Dateioperationen** - Copy, Move, Delete, Rename, Mkdir
- § **Zip/Unzip** - Komprimieren von Dateien
- § **SQL** – Datenbankabfragen (benötigt JDBC Driver)
- § **FTP** – Dateitransfer zu anderen Systemen
- § **Mail** – Versenden von e-Mails (auch mit Anhang)
- § **Telnet** – Kann eine Telnet-Session steuern
- § **Exec** – Führt ein lokales System Kommando aus
- § **Echo** – Gibt Messages aus
- § ...

## VSE Tasks

### § VSEConnectionManager

- Verwaltet TCP/IP Verbindungen zum VSE

### § Upload / Download (LIBR Members)

- Hochladen und herunterladen von LIBR Members, z.B Konfigurations-Dateien

### § SubmitJob

- Submitten von Jobs in den POWER Reader und abholen des Job-Outputs

### § Console

- Absetzen von Console-Kommandos und holen von Console-Messages

### § VSAM

- Download von VSAM Daten als Text, Comma Separated File (CSV) oder HTML

## Beispiele für den Einsatz von Ant

### § Dateitransfer Automatisierung

- Upload von Dateien ins VSE (z.B. VSAM)
- Anschließend starten eines Batch-Jobs zur Verarbeitung der hochgeladenen Daten

### § Laden einer Datenbank

- Download von VSAM Daten (per FTP oder VSAM-Task)
- Laden der Datenbank per Aufruf eines SQL-Loaders oder SQL Task
- z.B. Data Warehouse füttern

## Beispiele für den Einsatz von Ant (2)

### § Steuern von Jobabläufen

- Submitten von VSE Jobs
- Reagieren auf Return Codes von Jobs oder Fehlermeldungen im Listing
- Reagieren auf Console Messages
- Bei Fehler Mail/SMS an Operator

### § Switch zwischen Online und Batch

- Herunterfahren von CICS (per Console Task)
- Wenn CICS unten ist, Batch Jobs submitten oder starten
- Wenn Batch fertig ist, CICS wieder hochfahren

## Vorteile

- § Eine zentrale Steuerung für mehrere VSE Systeme
- § Einbinden von anderen Plattformen und Systemen in den Automatisierungs-Prozess
- § Zentrale Speicherung aller Automatisierungs-Scripte, Dateien, Job Templates, etc.
- § Zentrale Weiterverarbeitung, Aufbereitung und ggf. Archivierung von Logs

## Zusammenfassung

- § Apache Ant ist ein sehr mächtiges Werkzeug zur Automatisierung in heterogenen Umgebungen
- § Aufbauend auf dem Java-based Connector ist es mit kleinem Aufwand möglich ein Open Source Tool für VSE nutzbar zu machen



## Downloads

### § VSE Connector Client

- <http://www.ibm.com/servers/eserver/zseries/zvse/downloads/>

### § Apache Ant

- <http://ant.apache.org/>

### § VSE Tasks

- Die VSE Tasks sind momentan nicht auf der VSE Homepage, schreiben Sie eine e-mail an [zvse@de.ibm.com](mailto:zvse@de.ibm.com) und wir schicken Ihnen das Paket zu!

Fragen ?



Weiter zur Live Demo