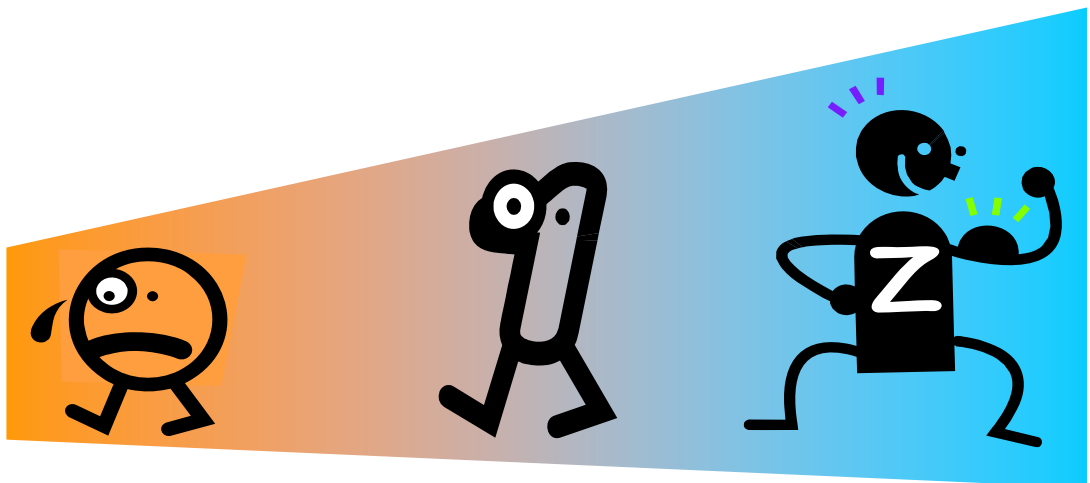


From zero to zHero: Java Batch development for IBM System z



Christian Strauer
Field Technical Sales Support IBM System z9
Mail: Christian.Strauer@de.ibm.com

Table of contents

Table of contents	2
Disclaimer	3
1 Introduction and general hints	4
2 Getting started	5
2.1 Setup PCOMM and start TSO	5
2.2 Basic information	8
3 HelloWorld (terminal based)	9
3.1 Verify Java installation	9
3.2 Your first Java HelloWorld	9
4 JZOS Labs	11
4.1 Installation of JZOS	11
4.2 HelloWorld with JZOS	13
4.2.1 Optional JZOS Lab 1.1 - diagnose problems	13
4.3 More MVS Java Programs	14
4.3.1 Write to operator console with Java	14
4.3.2 Handling condition codes with Java	16
4.4 Tomcat with JZOS	18
4.4.1 Install JZOS Samples	20
Appendix	22
A1 Basic FTP Tutorial	22
A2 Java Syntax	22
A3 z/OS Basics	22

Disclaimer

This document was created as an instruction guide for the Java Batch workshop in Germany and is no official IBM reference guide.

1 Introduction and general hints

This lab guide shell is intended as an introduction for people to become familiar with Java on z/OS.

You should not need any Java skill to run through these labs, but basic knowledge of z/OS and Mainframe technologies is required to understand them.

Here are some general hints that you should read before you begin with this lab:

- Java is case sensitive. So always be careful when you type Java source code!
- This lab guide is also available as PDF-document on the target machine under `/u/prak021/jbatch/docs/howto/lab_guide.pdf`.
- When you copy content from the lab guide and paste it into development tools, always be aware that some line breaks in this document might cause failures. In that case, remove the line break.
- Here are some general hints for ISPF:
 - When you are requested to press `<enter>`, please press the `<right CTRL>` key!
 - If you see stars (three stars!) `***` please press `<enter>`
 - You can only enter data in special screen areas. Use the `<TAB>` key to go to the next typo field
 - If you try to enter data in a non-typo area, your keyboard will be locked (see red sign, last line, left side `← ☺ →`)

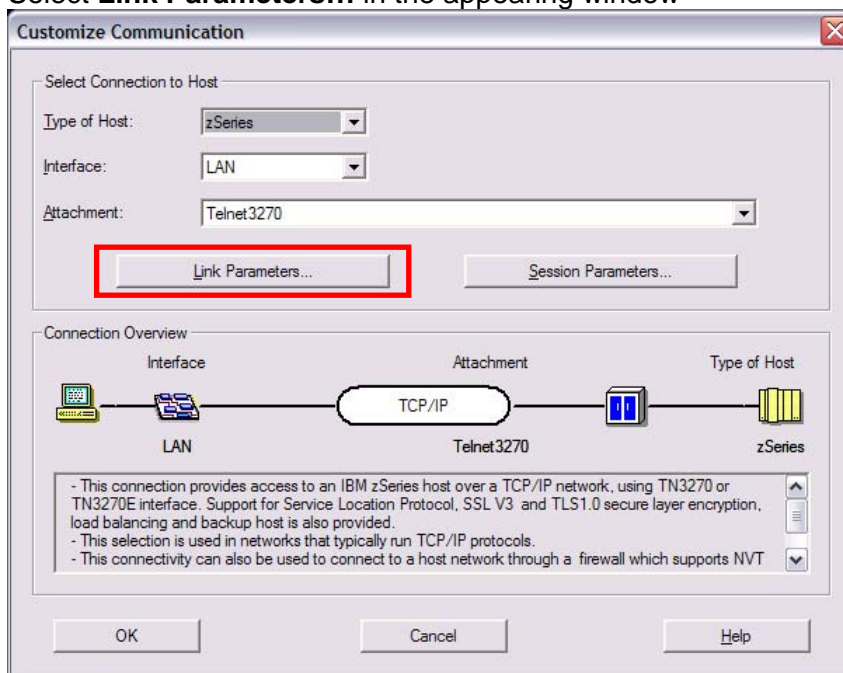
2 Getting started

This lab explains how to connect to the workshop host system and lists general information on the system structure.

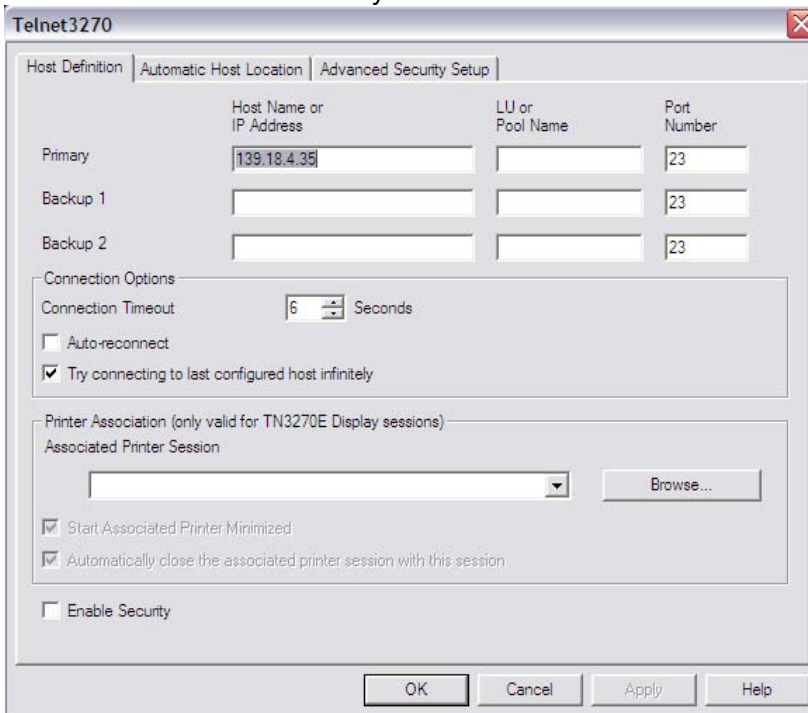
2.1 Setup PCOMM and start TSO

In this lab, we will establish a new TSO connection to the workshop host system.

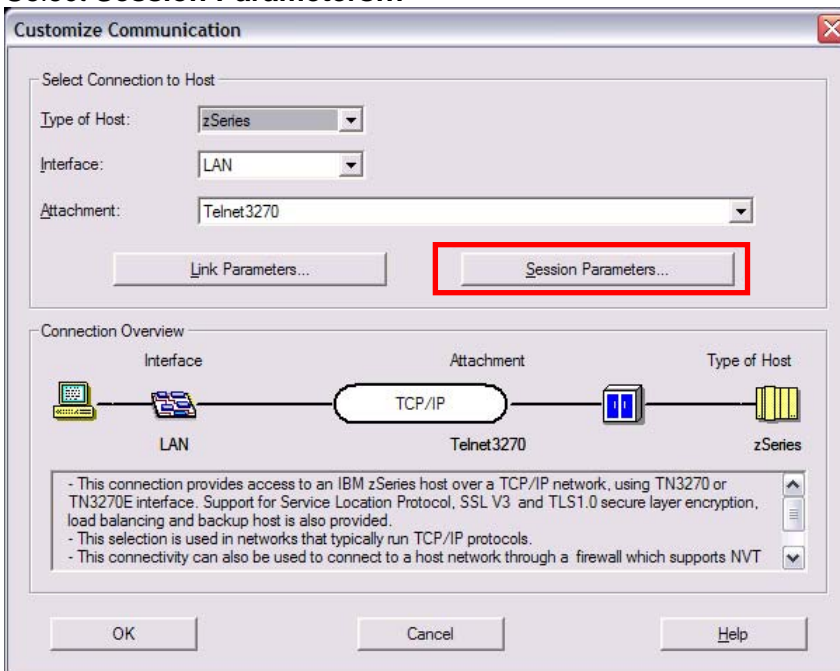
- 1) To establish a 3270 via TSO, Select **Programs** from the Windows Start menu → **IBM Personal Communications** → **Start or Configure Session**
- 2) Select **New Session...**
- 3) Select **Link Parameters...** in the appearing window



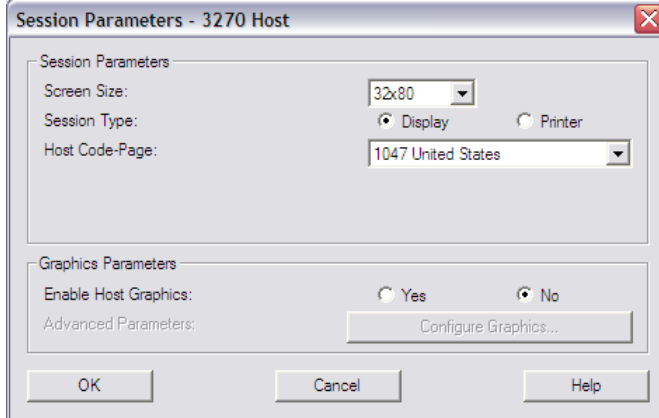
- 4) Enter 139.18.4.35 as Primary Host Name and select **OK**.



- 5) Select **Session Parameters...**

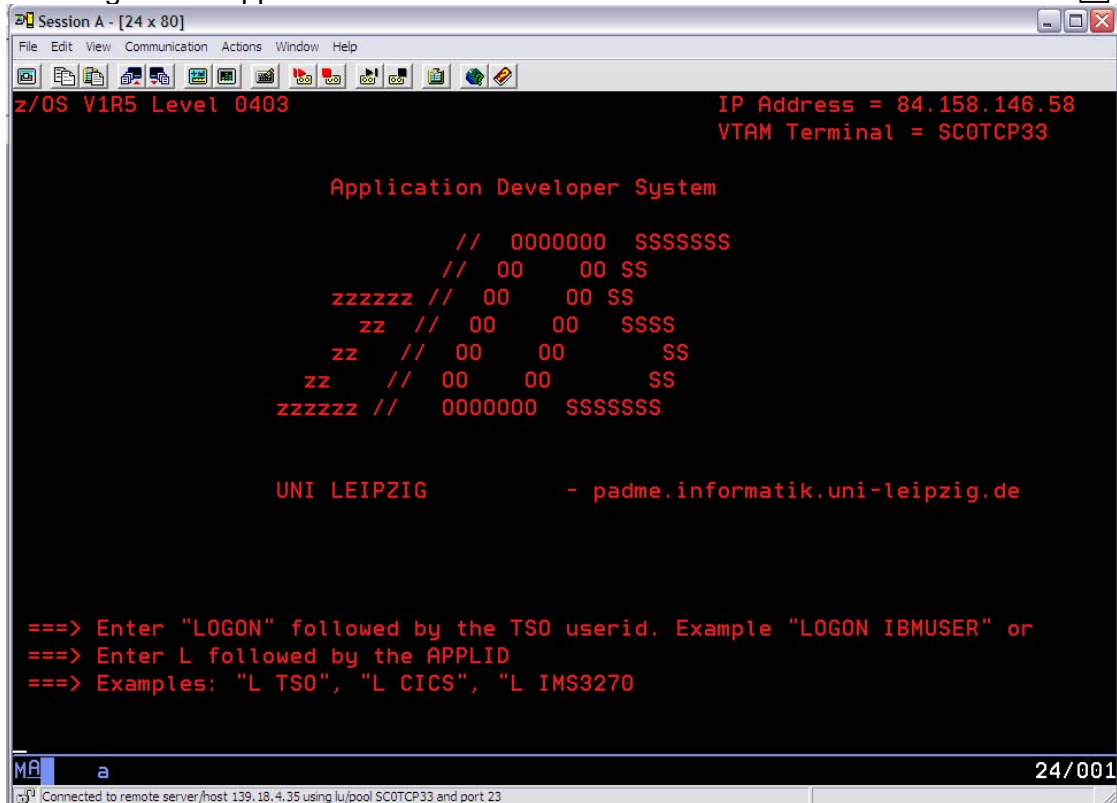


- 6) Select **32x80** as screen size and **1047** as Host Code-Page. Click **OK**.



- 7) Select **OK**.

- 8) You should now be able to login to TSO. Enter **TSO** and press Enter when the following screen appears:



- 9) Login with your credentials.

End of lab ☺

2.2 Basic information

The following table shows the most important properties of the workshop host system.

	Value	Comment
Userid	YOURUSERID	
Password	YOUR PW	
IP	139.18.4.35	
Ports	139.18.4.35:23 139.18.4.35:21	Telnet 3270 FTP
Java home directories	/usr/lpp/java/IBM/J1.3 /u/prak021/jbatch/local/java/J1.4 /u/prak021/jbatch/local/java/J5.0	Java 1.3 Java 1.4 Java 5.0
User's home directory	/u/YOURUSERID	
Workshop files	/u/prak021/jbatch	
User's HLQ	YOURUSERID	
Master home directory	/u/prak021	
Master HLQ	PRAK021	
JZOS Proclib	SYS1.PROCLIB	

3 HelloWorld (terminal based)

In this lab, you will create your first Java HelloWorld application on z/OS with a terminal session.

3.1 Verify Java installation

Before you start with HelloWorld, you have to verify that Java is installed correctly on z/OS.

1) Logon to TSO as described at the end of chapter 2.1.

2) Open an OMVS shell:

```
TSO OMVS
```

3) In the Unix shell type:

```
java -fullversion
```

4) If Java reports its version the JVM seems to be OK

End of lab ☺

3.2 Your first Java HelloWorld

This lab will show you how to develop a Java Hello World on the terminal.

1) Logon to TSO if you have not done so far.

2) Open an OMVS shell if you have not done so far:

```
TSO OMVS
```

3) Change to your home directory /u/YOURUSERID and create a new directory myjava. After each step, select enter:

```
cd /u/YOURUSERID  
mkdir myjava
```

4) Create a new Java file in this directory and edit it. After each step, select enter:

```
cd myjava  
oedit HelloWorld.java
```

5) Insert the following text into the new file:

```
class HelloWorld  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello World!");  
    }  
}
```

6) Save and exit with:

```
F3
```

7) Compile the Java file by entering the following command:

```
javac HelloWorld.java
```

8) This will create a new file HelloWorld.class. Enter the following command to run the HelloWorld example in the Java Virtual Machine (JVM):

```
java HelloWorld
```

9) You should see a *HelloWorld* on the command line.

10) Exit OMVS by entering

```
Exit
```

End of lab ☺

4 JZOS Labs

4.1 Installation of JZOS

- 1) Download the distribution .zip file to your workstation from <http://www.alphaworks.ibm.com/tech/zosjavabatchtk/download>.

Note: We recommend to download the jzos124.zip file from the /u/prak021/jbatch/setup directory on the host via FTP.

- 2) Extract the .zip contents to a local directory on your workstation (for example c:\jzos).
- 3) Transfer the extracted files from your workstation to the z/OS system in your home directory (see step 4)). Please be sure that you upload the following files in binary (no translation).

File	Host dataset
jzos.pax	'YOURUSERID.jzos.pax'
jzos.loadlib.xmit	'YOURUSERID.jzos.loadlib.xmit'
jzos.sampjcl.xmit	'YOURUSERID.jzos.sampjcl.xmit'

- 4) If you are uploading using FTP, use the following commands:

```
ftp 139.18.4.35
(login with your hostname, userid and password)
example:
```

```
Command Prompt - ftp 129.35.161.131
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ftp 129.35.161.131
Connected to 129.35.161.131.
220-FTPD1 IBM FTP CS U1R6 at TESTMUS.DEMOPKG.IBM.COM, 13:21:40 on 2007-04-20.
220 Connection will close if idle for more than 5 minutes.
User (129.35.161.131:(none)): FHKMSTR
331 Send password please.
Password:
230 FHKMSTR is logged on. Working directory is "FHKMSTR.".
ftp>
```

```
ftp> bin
ftp> quote site recfm=fb
ftp> quote site lrecl=80
ftp> put jzos.sampjcl.xmit 'YOURUSERID.JZOS.SAMPJCL.XMIT'
ftp> quote site cyl
ftp> quote site pri=2
ftp> put jzos.loadlib.xmit 'YOURUSERID.JZOS.LOADLIB.XMIT'
ftp> put jzos.pax 'YOURUSERID.JZOS.PAX'
ftp> quit
```

- 5) Log on to TSO with your userid and your password.

- 6) Enter the following TSO command to unload the load module library into a PDSE:

```
tso receive inda('YOURUSERID.JZOS.LOADLIB.XMIT')
```

- 7) When prompted with the message: "Enter restore parameters...", enter:

```
da('YOURUSERID.JZOS.LOADLIB')
```

Note: if you get an error *Unable to issue message with prompt for reply*, then enable prompting with the command:

```
tso profile prompt
```

Note: If the XMIT file is unloaded to YOURUSERID.PRIVATE.JZOS124.LOADLIB, delete that dataset and try the followig instead:

Enter the following TSO command to unload the load module library into a PDSE:

```
tso receive inda('YOURUSERID.JZOS.LOADLIB.XMIT')
```

When prompted with the message: "Enter restore parameters...", enter:

```
da(JZOS.LOADLIB)
```

- 8) Repeat to unload the sample JCL PDS:

```
tso receive inda('YOURUSERID.JZOS.SAMPJCL.XMIT')
```

When prompted with the message: "Enter restore parameters...", enter:

```
da('YOURUSERID.JZOS.SAMPJCL')
```

- 9) From the z/OS Unix shell, change to the directory under which you want to create the JZOS HFS files (/u/YOURUSERID) and unload the pax dataset: (This will create a directory named "jzos" relative to the current directory, which will be referred to later as JZOS_HOME)

```
pax -rvf "'/YOURUSERID.JZOS.PAX'"
```

- 10) If pax was successful please delete the following data sets:

- YOURUSERID.JZOS.SAMPJCL.XMIT
- YOURUSERID.JZOS.LOADLIB.XMI

End of lab ☺

4.2 HelloWorld with JZOS

- 1) Tailor the JZOS batch launcher proc contained in 'YOURUSERID.JZOS.SAMPJCL(EXJZOSVM)', updating it to point to the JZOS load module library that you unloaded in step 6) of chapter 4.1.
- 2) Following the instructions contained in 'YOURUSERID.JZOS.SAMPJCL', tailor member RUN13, RUN14 or RUN50 depending on the available Java JVM:

```
//XXXXXXXX JOB ← JOB-Name
//PROCLIB JCLLIB ORDER=YOURUSERID.JZOS.SAMPJCL
...
//JAVA EXEC PROC=EXJZOSVM,VERSION='13', ← or VERSION='14' or VERSION='50'
...
export JZOS_HOME=/u/YOURUSERID/jzos
export JAVA_HOME=/usr/lpp/java/IBM/J1.3 ← Or directory of newer JVM version
...
```

- 3) SUBMIT the modified JCL and check the job log.

If everything was set up properly, the SYSOUT DD should contain output like this:

```
...
JZOSVM13(N): Copyright(C) IBM Corp. 2005. All Rights Reserved
JZOSVM13(N): Version 1.2.3
JZOSVM13(N): Java Virtual Machine created
JZOSVM13(N): Invoking com.dovetail.jzos.sample.HelloWorld.main()
JZOSVM13(N): com.dovetail.jzos.sample.HelloWorld.main() completed.
```

And the JOB STDOUT DD should contain:

```
Hello World!
```

End of lab ☺

4.2.1 Optional JZOS Lab 1.1 - diagnose problems

- 1) To diagnose problems with the JZOS batch launcher, change the LOGLVL parameter to '+I' :

```
// EXEC EXJZOSVM,LOGLVL='+I',
```

NOTE: Setting this logging level (+I) will dump the environment that is passed to the JVM. The trace level setting "+T" will produce many messages, some of which may be helpful in tracking down installation problems.

End of lab ☺

4.3 More MVS Java Programs

This chapter covers different Java programs that access native MVS resources. The Java source files for this chapter can be found here: /u/prak021/jbatch/source.

4.3.1 Write to operator console with Java

This chapter shows to write messages to the operator console from Java

- 1) Copy /u/prak021/jbatch/source/Wto.java to /u/YOURUSERID/Wto.java:

```
cp /u/prak021/jbatch/source/Wto.java /u/YOURUSERID/myjava/Wto.java
```

- 2) Have a look at the source code:

```
cd /u/YOURUSERID/myjava
oedit Wto.java
```

- 3) Close the file with F3.

- 4) Compile that file with javac:

```
javac Wto.java -classpath /u/YOURUSERID/jzos/jzos.jar
```

- 5) Exit the OMVS shell by entering

```
Exit
```

- 6) Copy the JCL that you have used in chapter 4.2, step 2) into a new JCL 'YOURUSERID.SAMPLES.JCL(WTO)'

- 7) Modify JAVACLS and Classpath in the JCL:

```
//XXXXXXXX JOB ← JOB-Name
//PROCLIB JCLLIB ORDER=YOURUSERID.JZOS.SAMPJCL
...
//JAVA EXEC PROC=EXJZOSVM,VERSION='13', ← or VERSION='14' or VERSION='50'
// JAVACLS='Wto'
//STDENV DD *
...
export JZOS_HOME=/u/YOURUSERID/jzos
export JAVA_HOME=/usr/lpp/java/IBM/J1.3 ← Or directory of newer JVM version
...
# Customize your CLASSPATH here
CLASSPATH=/u/YOURUSERID/myjava
...
```

- 8) Submit the JCL and check the output with SDSF.

- 9) Go to the operator console and check for the WTO. It should look similar to this one:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
-----
Display Filter View Print Options Help
-----
SDSF SYSLOG 6368.108 MVS1 MVS1 04/25/2007 0W 164322 COLUMNS 38 117
COMMAND INPUT ==> SCROLL ==> CSR
JOB09345 00000090 $HASP373 JZOS STARTED - INIT 1 - CLASS A - SYS MVS1
JOB09345 00000090 IEF403I JZOS - STARTED - TIME=12.34.58
JOB09345 00000090 @ZJAV000I Current job is 'JZOS'.
JOB09345 00000290 - --TIMINGS (MINS.)--
-----PAGING COUNTS-----
JOB09345 00000290 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CLOCK
PAGE SWAP VIO SWAPS
JOB09345 00000290 -JZOS JAVA JAVAJVM 00 3167 .00 .00 .01
0 0 0 0
JOB09345 00000090 IEF404I JZOS - ENDED - TIME=12.34.59
JOB09345 00000290 -JZOS ENDED. NAME- TOTAL CPU TIME=
ELAPSED TIME= .01
JOB09345 00000090 $HASP395 JZOS ENDED
00000090 $HASP309 INIT 1 INACTIVE ***** C=A
INSTREAM 00000290 LOGON
TSU09346 00000281 $HASP100 FHKMSTR ON TS0INRDR
TSU09346 00000090 $HASP373 FHKMSTR STARTED
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
MA b 05/021
Connected to remote server/host 129.35.161.131 using lu/pool TCP00142 and port 23
    
```

4.3.2 Handling condition codes with Java

This chapter shows to write create a little Job net based on Java return codes.

- 10) Copy /u/prak021/jbatch/source/ConditionCode.java to /u/YOURUSERID/ConditionCode.java:

```
cp /u/prak021/jbatch/source/ConditionCode.java
/u/YOURUSERID/myjava/ConditionCode.java
```

- 11) Have a look at the source code:

```
cd /u/YOURUSERID/myjava
oedit ConditionCode.java
```

- 12) Close the file with F3.

- 13) Compile that file with javac:

```
javac ConditionCode.java
```

- 14) Exit the OMVS shell by entering

```
Exit
```

- 15) Copy the JCL that you have used in chapter 4.2, step 2) into a new JCL 'YOURUSERID.SAMPLES. JCL(CC)'

- 16) Modify JAVACLS, ARGS and Classpath in the JCL:

```
//XXXXXXXX JOB                                ← JOB-Name
//PROCLIB JCLLIB ORDER=YOURUSERID.JZOS.SAMPJCL
...
//JAVA EXEC PROC=EXJZOSVM,VERSION='13',      ← or VERSION='14' or VERSION='50'
// JAVACLS='ConditionCode',
// ARGS='1'
//STDENV DD *
...
export JZOS_HOME=/u/YOURUSERID/jzos
export JAVA_HOME=/usr/lpp/java/IBM/J1.3     ← Or directory of newer JVM version
...
# Customize your CLASSPATH here
CLASSPATH=/u/YOURUSERID/myjava
...
```

- 17) In the JCL, add the following script which starts a HelloWorld if the return code of the Java ConditionCode program is '1'

```
//*****
// IF (RC = 1) THEN
//*****
//HELLO EXEC PROC=EXJZOSVM,VERSION='13',
// JAVACLS='com.dovetail.jzos.sample.HelloWorld'
//STDENV DD *
# This is a shell script which configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
. /etc/profile
export JZOS_HOME=/u/YOURUSERID/jzos
export APPL_HOME=/u/YOURUSERID/myjava
export JAVA_HOME=/usr/lpp/java/IBM/J1.3
export PATH="$PATH":"${JAVA_HOME}"/bin:
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin/classic
LIBPATH="$LIBPATH":"${JZOS_HOME}"
export LIBPATH="$LIBPATH":
# Customize your CLASSPATH here
# Add JZOS required jars to end of CLASSPATH
for i in "${JZOS_HOME}"/*.jar; do
```



```
CLASSPATH="$CLASSPATH": "$i"
done
export CLASSPATH="$CLASSPATH":
# Set JZOS specific options
# Use this variable to specify encoding for DD STDOUT and STDERR
#export JZOS_OUTPUT_ENCODING=Cp1047
# Use this variable to prevent JZOS from handling MVS operator commands
#export JZOS_ENABLE_MVS_COMMANDS=false
# Use this variable to supply additional arguments to main
#export JZOS_MAIN_ARGS=""
# Configure JVM options
IJO="-Xms16m -Xmx128m"
IJO="$IJO -Djzos.home=${JZOS_HOME}"
# Uncomment the following if you want to run without JIT
#IJO="$IJO -Djava.compiler=NONE"
# Uncomment the following if you want to run with Ascii file encoding..
IJO="$IJO -Dfile.encoding=ISO8859-1"
export IBM_JAVA_OPTIONS="$IJO "
export JAVA_DUMP_HEAP=false
export JAVA_PROPAGATE=NO
//*****
// ENDIF
//*****
//
```

- 18) Submit the JCL and check the output with SDSF.
- 19) Modify ARGS='O' in the JCL
- 20) Again, submit the JCL and check the output with SDSF. The HelloWorld should **not** be invoked.

End of lab ☺

4.4 Tomcat with JZOS (optional)

The Apache Tomcat servlet container can be installed quickly and easily using the JZOS toolkit.

- 1) Create a new directory /u/prak021/jbatch/users/YOURUSERID

```
mkdir /u/prak021/jbatch/users/YOURUSERID
```

- 2) Download the .zip version of the binary distribution of Tomcat and upload it (in binary mode) to /u/prak021/jbatch/users.

Apache Jakarta download page: <http://jakarta.apache.org/site/binindex.cgi>
binary .zip distribution: <http://tomcat.apache.org/download-55.cgi>

Note: You can also find the binary zip installation files in the /u/prak021/jbatch/setup directory and copy it directly to your home directory /u/YOURUSERID!

```
cp /u/prak021/jbatch/setup/apache-tomcat-5.5.15.zip  
/u/prak021/jbatch/users/YOURUSERID
```

- 3) From a z/OS Unix shell, change to your home directory extract the Tomcat zip file.

Note: this will create a directory "apache-tomcat-5.5.15" under the current directory!

```
cd /u/prak021/jbatch/users/YOURUSERID  
jar -xvf apache-tomcat-5.5.15.zip
```

- 4) Delete the zip file:

```
rm apache-tomcat-5.5.15.zip
```

- 5) For convenience, create a symbolic link to the Tomcat distribution:

```
ln -s apache-tomcat-5.5.15 tomcat
```

- 6) Modify ports in server.xml:

- o Download /u/prak021/jbatch/users/YOURUSERID/apache-tomcat-5.5.15/conf/server.xml via FTP in binary mode
- o Edit server.xml on your local workstation and change all ports depending on your userid, e.g. change 8080 to 80xx
- o Upload the modified server.xml to /u/prak021/jbatch/users/YOURUSERID/apache-tomcat-5.5.15/conf/server.xml via FTP in binary mode.

- 7) Create a new member TOMCAT in the dataset YOURUSERID.SAMPLES. JCL' and insert the following JCL content:

```
//TOMCAT JOB  
//PROCLIB JCLLIB ORDER=YOURUSERID.JZOS.SAMPJCL  
//...  
//JAVA EXEC PROC=EXJZOSVM,VERSION='50',  
// JAVACLS='org.apache.catalina.startup.Bootstrap',  
// ARGS='start'  
//STDENV DD *  
# This is a shell script which configures  
# any environment variables for the Java JVM.  
# Variables must be exported to be seen by the launcher.  
. /etc/profile
```

```
export JZOS_HOME=/u/YOURUSERID/jzos
export TOMCAT_HOME=/u/prak021/jbatch/users/YOURUSERID/apache-tomcat-5.5.15
export JAVA_HOME=/u/prak021/jbatch/local/java/J5.0
export PATH=/bin:${JAVA_HOME}/bin:
LIBPATH=/lib:/usr/lib:${JAVA_HOME}/bin:${JAVA_HOME}/bin/classic
LIBPATH=${LIBPATH}:${JZOS_HOME}
export LIBPATH=${LIBPATH}:

CLASSPATH=${JAVA_HOME}/lib/tools.jar
CLASSPATH=${CLASSPATH}:${TOMCAT_HOME}/bin/bootstrap.jar
CLASSPATH=${CLASSPATH}:${JZOS_HOME}/jzos.jar
CLASSPATH=${CLASSPATH}:${TOMCAT_HOME}/bin/commons-logging-api.jar
export CLASSPATH=${CLASSPATH}:
# Set JZOS specific options
# Use this variable to specify encoding for DD STDOUT and STDERR
#export JZOS_OUTPUT_ENCODING=Cp1047
# Use this variable to prevent JZOS from handling MVS operator commands
#export JZOS_ENABLE_MVS_COMMANDS=false
# Use this variable to supply additional arguments to main
#export JZOS_MAIN_ARGS=""
# Configure JVM options
# Note that Tomcat requires default ASCII file.encoding
IJO="-Xms64m -Xmx128m"
IJO=${IJO} -Dfile.encoding=ISO8859-1
IJO=${IJO} -Djzos.home=${JZOS_HOME}
IJO=${IJO} -Dcatalina.base=${TOMCAT_HOME}
IJO=${IJO} -Dcatalina.home=${TOMCAT_HOME}
IJO=${IJO} -Djava.io.tmpdir=${TOMCAT_HOME}/temp
IJO=${IJO} -Djava.endorsed.dirs=${TOMCAT_HOME}/common/endorsed"
# Uncomment the following if you want to run without JIT
#IJO=${IJO} -Djava.compiler=NONE"
export IBM_JAVA_OPTIONS=${IJO}
export JAVA_DUMP_HEAP=false
export JAVA_PROPAGATE=NO
export IBM_JAVA_ZOS_TDUMP=NO
//
```

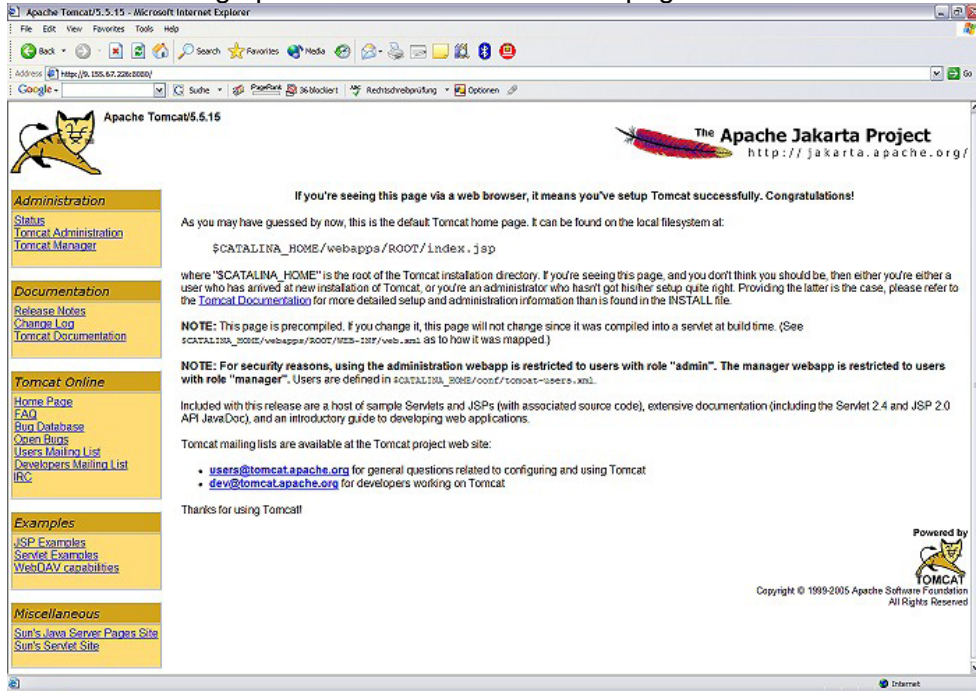
8) Submit the JCL

9) Check results with SDSF:
STDOUT DD should contain something like this (but will vary depending on the JDK version):

```
[INFO] Http11Protocol - Initializing Coyote HTTP/1.1 on http-8080
[INFO] Catalina - Initialization processed in 129605 ms
[INFO] StandardService - Starting service Catalina
[INFO] StandardEngine - Starting Servlet Engine: Apache Tomcat/
[INFO] StandardHost - XML validation disabled
[INFO] StandardHost - Create Host deployer for direct deployment ( non-jmx )
[INFO] StandardHostDeployer - Processing Context configuration file URL
...
[INFO] Catalina - Server startup in 454422 ms
```

10) Test Tomcat:
Go to <http://139.18.4.35:80xx/>

This should bring up the Tomcat administration page:



11) Purge the TOMCAT Job

4.4.1 Install JZOS Samples

- 1) To deploy applications, you first of all have to install the Tomcat Admin Page: unzip the /u/prak021/jbatch/setup/apache-tomcat-5.5.15-admin.zip to your Tomcat home:

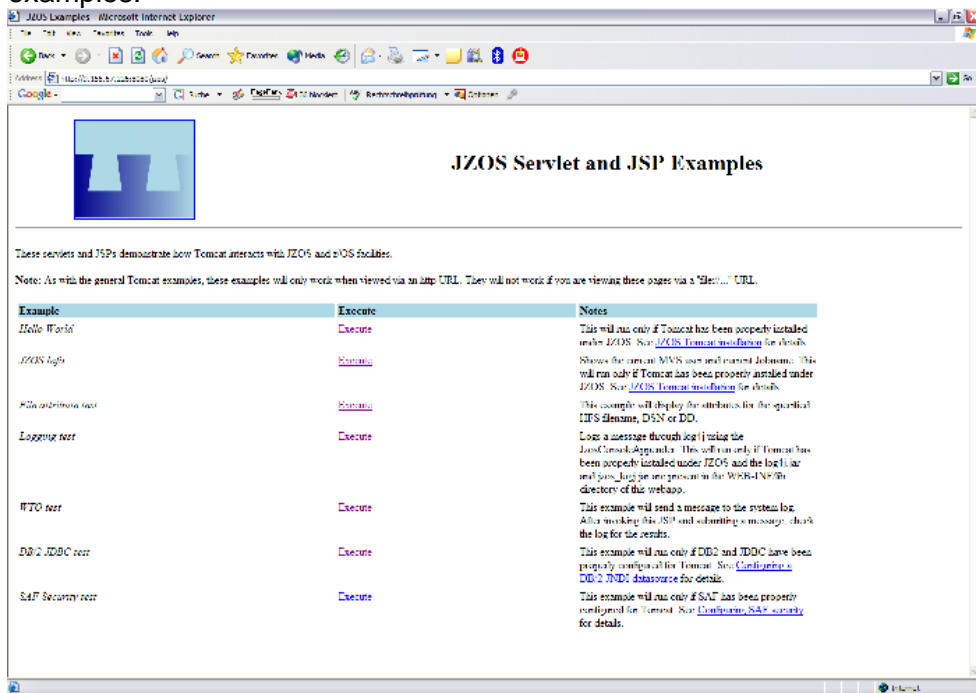
```
Cd /u/prak021/jbatch/users/YOURUSERID
jar -xvf /u/prak021/jbatch/setup/apache-tomcat-5.5.15-admin.zip
```

- 2) Download /u/prak021/jbatch/users/YOURUSERID/apache-tomcat-5.5.15/conf/tomcat-users.xml in binary mode via FTP to your workstation.
- 3) Add the following lines in the xml file:

```
<role rolename="admin"/>
<role rolename="manager"/>
...
<user username="admin" password="manager" roles="admin,manager"/>
```
- 4) Save the tomcat-users.xml file and upload again to the host in binary mode to /u/prak021/jbatch/users/YOURUSERID/apache-tomcat-5.5.15/conf
- 5) Restart Tomcat by submitting the TOMCAT job again.
- 6) Install and test JZOS servlet examples:

Download the jzos.war from /u/prak021/jbatch/setup to your local workstation and deploy it with the Tomcat Manager on the admin page (Login with username "admin" and password "manager").

- 7) After deploying the .war file, go to <http://139.18.4.35:80xx/jzos> and test the servlet examples.



- 8) Purge the TOMCAT Job

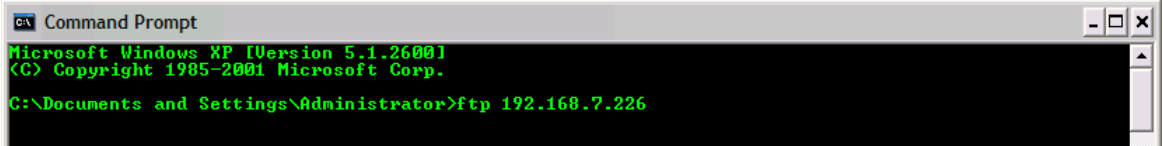
End of lab ☺

Appendix

A1 Basic FTP Tutorial

This tutorial explains how to use FTP for downloading files from the host.

- 1) In Windows, select **Start** → **Run** → Enter **cmd** and press Enter.
- 2) In the command line, enter ftp 139.18.4.35



```
Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrator>ftp 192.168.7.226
```

- 3) Enter YOURUSERID as username and <your password> as password.
- 4) Enter **lcd <local_dir>** to change to your local workstation directory where you want to place your downloaded files.
- 5) Enter **bin** to use binary mode.
- 6) Enter **cd <host_dir>** to change to the host directory where you want to download the files from.
- 7) Enter **get <destination_file>** to get the desired file.
- 8) If you have finished downloading all files, enter **quit** to leave the FTP session

A2 Java Syntax

Further information on Java can be found here:

- *Handbuch der Java-Programmierung*: <http://www.javabuch.de/> (German)
- *Java ist auch eine Insel and Java 2 und Praxisbuch Objektorientierung*: <http://www.galileocomputing.de/openbook> (German)

A3 z/OS Basics

A good introduction to z/OS basics can be found in the redbook *Introduction to the New Mainframe: z/OS Basics* under

<http://www.redbooks.ibm.com/abstracts/sg246366.html?Open>

Table of contents:

- Part 1. Introduction to z/OS and the mainframe environment
- Chapter 1. Introduction to the new mainframe
- Chapter 2. Mainframe hardware systems and high availability
- Chapter 3. z/OS overview
- Chapter 4. TSO/E, ISPF, and UNIX: Interactive facilities of z/OS
- Chapter 5. Working with data sets
- Chapter 6. Using JCL and SDSF
- Chapter 7. Batch processing and JES
- Part 2. Application programming on z/OS
- Chapter 8. Designing and developing applications for z/OS

- Chapter 9. Using programming languages on z/OS
- Chapter 10. Compiling and link-editing a program on z/OS
- Part 3. Online workloads for z/OS
- Chapter 11. Transaction management systems on z/OS
- Chapter 12. Database management systems on z/OS
- Chapter 13. z/OS HTTP Server
- Chapter 14. WebSphere Application Server on z/OS
- Chapter 15. Messaging and queuing
- Part 4. System programming on z/OS
- Chapter 16. Overview of system programming
- Chapter 17. Using SMP/E
- Chapter 18. Security on z/OS
- Chapter 19. Network Communications on z/OS
- Appendix A. A brief look at IBM mainframe history
- Appendix B. DB2 sample tables
- Appendix C. Utility programs
- Appendix D. EBCDIC - ASCII table
- Appendix E. Class Program
- Appendix F. Back matter
- Appendix G. Glossary