



## Tivoli Workload Scheduler for z/OS

*Data set triggering*

*JZOS batch launcher implementation*

**Tivoli.** software



© 2009 IBM Corporation  
Updated August 26, 2009

### **JZOS batch launcher implementation.**

In this training module you will learn how to implement and verify the JZOS batch launcher.

The JZOS batch launcher is a feature of Java™ for z/OS® SDK version 5. Install and verify Java for z/OS SDK V5 before you implement the JZOS batch launcher version 2.3.0. For more information about the installation, see the *Java for z/OS SDK V5 installation and verification* training module.

## JZOS batch launcher overview

- IBM 64-bit Java for z/OS SDK V5 provides an optional feature for running Java programs in batch jobs under z/OS
  - ▶ Known as the JZOS batch launcher version 2.3.0
  - ▶ Install the 64-bit Java for z/OS SDK V5 before you use the JZOS batch launcher
- TWS for z/OS uses the batch launcher to support EDWA data set triggering
  - ▶ Supports XML-based data set selection criteria for data set triggering
  - ▶ Supports Tivoli Dynamic Workload Console reporting
- The JZOS batch launcher provides an executable module, a JCL procedure, and sample JCL batch job in the **J5.0\_64/mvstools** directory
  - Implement these items under z/OS

### JZOS batch launcher overview.

Tivoli Workload Scheduler for z/OS 8.5 uses the JZOS batch launcher to enable Java programs in batch mode under z/OS.

Tivoli Workload Scheduler for z/OS 8.5 uses these Java programs to process XML and to support historical run data archiving for Tivoli Dynamic Workload Console reporting.

The XML defines data set selection criteria for data set triggering. For more information about using the XML, see the *XML Implementation* training module.

The files for implementing the JZOS batch launcher are created in the **J5.0\_64/mvstools** directory when you install the Java for z/OS SDK V5.

## mvstools directory

- Assume the Java for z/OS SDK V5 is installed in the path:  
**/usr/lpp/java/J5.0\_64**

- List the J5.0\_64/mvstools directory contents

```
JAYHILL:/usr/lpp/java/J5.0_64/mvstools #>ls -al
total 912
drwxr-xr-x  3 ROOT    OMVS      8192 Apr  8 19:11 .
drwxr-xr-x  9 ROOT    OMVS      8192 Apr  8 19:15 ..
-rw-r--r--  1 ROOT    OMVS    442368 Apr  8 19:11 JVMLDM56
drwxr-xr-x  5 ROOT    OMVS      8192 Apr  8 19:11 samples
JAYHILL:/usr/lpp/java/J5.0_64/mvstools #>
```

- The directory contains an executable load module **JVMLDM56** and a subdirectory called **samples** that provides a sample procedure and sample JCL
- JVMLDM56 is the Java virtual machine used in batch jobs to run Java programs under z/OS

### mvstools directory.

For the remainder of this training module, assume the Java for z/OS SDK V5 is installed in the path, **/usr/lpp/java/J5.0\_64**.

If you list the contents of the **J5.0\_64/mvstools** directory, the first file you see in the directory is the JVMLDM56 file.

JVMLDM56 is an executable load module that provides the Java virtual machine for z/OS batch jobs. Copy this module into an executable load library **PDSE** under z/OS.

The default load library used by the SMP/E installation is SYS1.SIEALNKE.

The JZOS batch launcher uses a subdirectory called **samples** that contains a sample procedure and a sample JCL.

## mvstools/samples/jcl directory

- The **J5.0\_64/mvstools/samples/jcl** directory

```
JAYHILL: /usr/lpp/java/J5.0_64/mvstools/samples/jcl #>ls -al
total 64
drwxr-xr-x  2 ROOT    OMVS      8192 Apr  8 19:11 .
drwxr-xr-x  5 ROOT    OMVS      8192 Apr  8 19:11 ..
-rw-r--r--  1 ROOT    OMVS      1877 Apr  8 19:11 JVMJCL56
-rw-r--r--  1 ROOT    OMVS      1771 Apr  8 19:11 JVMPRC56
-rw-r--r--  1 ROOT    OMVS      1320 Apr  8 19:11 PROCLIB.EKM2
-rw-r--r--  1 ROOT    OMVS      1075 Apr  8 19:11 PROCLIB.EKM2ENV
JAYHILL: /usr/lpp/java/J5.0_64/mvstools/samples/jcl #>
```

- Contains the two sample JCL files **JVMJCL56** and **JVMPRC56**
- JVMPRC56 is a JCL procedure in SYS1.PROCLIB
  - Executes the Java virtual machine load module JVMLDM56
- JVMJCL56 is a sample JCL batch job that executes the JVMPRC56 JCL procedure to run a Java program class

### mvstools/samples/jcl directory.

The **J5.0\_64/mvstools/samples/jcl** directory contains the two sample JCL files for implementing the JZOS batch launcher.

The **JVMPRC56** file is a JCL procedure that can be executed from a JCL batch job. JVMPRC56 executes the Java virtual machine load module JVMLDM56. Copy the JVMPRC56 file as a member into SYS1.PROCLIB or some other JCL procedure library PDS under z/OS and customize the file before you use it.

**JVMJCL56** is a sample JCL batch job that executes the JVMPRC56 JCL procedure to run a Java program class in batch mode under z/OS. Copy the JVMJCL56 file to a z/OS JCL file or a partitioned data set member and customize the file for your environment.

Use the JVMJCL56 sample to verify and test the JZOS batch launcher.

## Implementing the JZOS batch launcher ( 1 of 3)

1. Copy the J5.0\_64/mvstools/JVMLDM56 file to a z/OS **PDSE** load module library with a member name of **JVMLDM56** using the following commands from within OMVS:

```
cd /usr/lpp/java/J5.0_64/mvstools
cp -X JVMLDM56 '//`USER.LINKLIB2(JVMLDM56)`'
```

2. Copy the sample JCL procedure J5.0\_64/mvstools.samples/jcl/JVMPRC56 into a z/OS PDS procedure library with member name **JVMPRC56**

```
cd /usr/lpp/java/J5.0_64/mvstools/samples/jcl
cp JVMPRC56 '//`USER.PROCLIB(JVMPRC56)`'
```

3. Copy the sample JCL job J5.0\_64/mvstools.samples/jcl/JVMJCL56 into a z/OS PDS procedure library with member name **JVMJCL56**

```
cd /usr/lpp/java/J5.0_64/mvstools/samples/jcl
cp JVMJCL56 '//`USER.JCLLIB(JVMJCL56)`'
```

### Implementing the JZOS batch launcher (1 of 3).

Change directories to the **mvstools** subdirectory under the **J5.0\_64** directory where the z/OS Java SDK V5 is installed.

Use the USS **cp** command as shown on the slide to copy the executable load module file JVMLDM56 into an executable load library of type PDSE as a member. The **-X** parameter on the copy tells the **cp** command that the file is an executable.

If successful, you return to the USS OMVS prompt with no messages displayed.

Change directories to the **mvstools/samples/jcl** subdirectory under the **J5.0\_64** directory where the z/OS Java SDK V5 is installed. Use the USS **cp** command as shown on the slide to copy the sample JCL procedure file JVMPRC56 into a JCL procedure library PDS member.

If successful, you return to the USS OMVS prompt with no messages displayed.

From that same **mvstools/samples/jcl** subdirectory use the USS **cp** command as shown on the slide to copy the sample JCL batch job file JVMJCL56 into a JCL job library PDS member.

If successful, you return to the USS OMVS prompt with no messages displayed.

## Implementing the JZOS batch launcher ( 2 of 3)

Customize the **JVMPC56** JCL procedure

```

/*****
//JVMPC56 PROC JAVACLS=,           < Fully Qfied Java class..RQD
//  ARGV=,                         < Args to Java class
//  LIBRARY='USER.LINKLIB2',       < STEPLIB FOR JVMPC56 module
//  VERSION='56',                  < JVMPC56 version: 56
//  LOGLVL='',                     < Debug LVL: +I(info) +T(trc)
//  REGSIZE='0M',                  < EXECUTION REGION SIZE
//  LEPARM=''
//JAVAJVM EXEC PGM=JVMPC56,REGION=&REGSIZE,
//  PARM='&LEPARM/&LOGLVL &JAVACLS &ARGV'
//STEPLIB DD DSN=&LIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=*             < System stdout
//SYSOUT DD SYSOUT=*               < System stderr
//STDOUT DD SYSOUT=*               < Java System.out
//STDERR DD SYSOUT=*               < Java System.err
//CEEDUMP DD SYSOUT=*
//ABNLIGNR DD DUMMY
/**
/**The following DDs can/should be present in the calling JCL
/**
/**STDIN DD                        < OPTIONAL - Java System.in
/**STDENV DD                       < REQUIRED - JVM Environment script
/**MAINARGS DD                     < OPTIONAL - Alt. method to supply args
// PEND

```

### Implementing the JZOS batch launcher (2 of 3).

Tailor the JVMPC56 JCL procedure for your installation.

Replace '`<HLQ>.JZOS.LOADLIB`' with the load library PDSE that contains the JVMPC56 module.

In the example on this slide `USER.LINKLIB2` has been assigned to the `LIBRARY` symbolic parameter. The `LIBRARY` parameter and `STEPLIB` DD statements are commented out by default.

If you use the z/OS Java SDK V5 file from the Internet and the **pax** command to install the z/OS Java SDK V5, uncomment the `LIBRARY` and `STEPLIB` statements. Modify the `LIBRARY` statement to point to the location of the JVMPC56 module.

If you use SMP/E to install the z/OS Java SDK V5, then the JVMPC56 load module is in the `LNKST` concatenation in the load library `SYS1.SIEALNKE`. In this case, do not uncomment the `STEPLIB` DD statement. In the example on the slide the `STEPLIB` DD statement is uncommented.

On the `EXEC` statement the **PGM=** parameter refers to the JVMPC56 load module when the version symbolic parameter has been applied.

## Implementing the JZOS batch launcher ( 3 of 3)

Customize the **JVMJCL56** sample batch job

```
//JVMJCL56 JOB '9999','Run Hello World',
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOT1)Y=&SYSUID
//*****
//*
//* Batch job to run the Java VM
//*
//* Tailor the proc and job for your installation:
//* 1.) Modify the Job card per your installation's requirements
//* 2.) Modify the PROCLIB card to point to this PDS
//* 3.) edit JAVA_HOME to point the location of the SDK
//* 4.) edit APP_HOME to point the location of your app (if any)
//* 5.) Modify the CLASSPATH as required to point to your Java code
//* 6.) Modify JAVACLS and ARGS to launch desired Java class
//*
//*****
//JAVA EXEC PROC=JVMPRC56,
// JAVACLS='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 JAVA_HOME=/usr/lpp/java/J5.0_64
export PATH=/bin:"${JAVA_HOME}"/bin:
```

### Implementing the JZOS batch launcher (3 of 3).

Review the JVMJCL56 sample job. Read and follow the comments in the job for customizing the job. The job runs the HelloWorld Java class.

If you installed the Java for z/OS SDK V5 in **/usr/lpp/java/J5.0\_64**, all you need to do is add a job card. The job executes the JVMPRC56 procedure passing the JAVACLS symbolic parameter, which specifies the Java class to be executed. In this case the class is the HelloWorld class.

The STDENV DD \* statement provides the USS commands to establish the Java environment. This statement includes exporting the JAVA\_HOME path as shown in the slide. If you installed Java for z/OS SDK V5 in the **/usr/lpp/java/J5.0\_64** directory, the existing

STDENV DD \* statements work as defined. In this case the only customization required for the job is a job card. A job card has been added in this example as shown in the screen capture on this slide.

## Verifying the JZOS batch launcher setup

- Submit the JVMJCL56 sample job
- The job should end with condition code zero (cc=0)
- Here is the expected output from the job sysout

```
JVMJZBL1001N JZOS batch Launcher Version: 2.3.0 2008-05-12
JVMJZBL1002N Copyright (C) IBM Corp. 2005. All rights reserved.
java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition (build pmz64devifx-20090408 (SR
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390x-64 j9vmmz6423ifx-20090324
J9VM - 20090319_32038_BHdSMr
JIT - 20081112_1511ifx1_r8
GC - 200811_07)
JVMJZBL1023N Invoking HelloWorld.main()...
JVMJZBL1024N HelloWorld.main() completed.
JVMJZBL1021N JZOS batch launcher completed, return code=0
```

- **Hello World** text is the expected output from stdout

### Verifying the JZOS batch launcher setup.

After customizing the JVMJCL56 job and adding a job card, submit the job. The job completes with a condition code zero.

Review the job output in SDSF. The job produces sysout messages with the JZOS batch launcher and Java for z/OS SDK version information as shown in the screen capture on the slide.

The job **stdout** contains the Hello World text produced by the HelloWorld Java class that was executed. This output verifies that the JZOS batch launcher is working.



## Summary

In summary you should now be able to:

- Implement and verify the Java for z/OS SDK V5 JZOS batch launcher

### **Summary.**

In summary you should now be able to implement and verify the Java for z/OS SDK V5 JZOS batch launcher.

## Feedback

### Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_jzos-batch-launcher-implementation.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_jzos-batch-launcher-implementation.ppt)

This module is also available in PDF format at: [../jzos-batch-launcher-implementation.pdf](..jzos-batch-launcher-implementation.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.

## Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:  
Tivoli z/OS

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

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

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.