

Java: z/OS[®] Stand-Alone Applications

Steve Goetze
Kirk Wolf
Dovetailed Technologies, LLC.

<http://www.dovetail.com>

SHARE Session #8368 - February 14, 2007



Agenda

- Using the *IBM JZOS Batch Toolkit for z/OS SDKs* to run Java in batch jobs or started tasks
- Integrate Java applications with MVS datasets and services
- Use Eclipse effectively to develop, test, and 'one-click' deploy your application
- Reuse Java applications and skills in traditional z/OS workloads
- Exploit open source Java tools and libraries on z/OS, such as Apache Tomcat and Apache Commons
- Survey common application scenarios
- Shameless plugs for two Hands-on labs that follow this session
 - S8369 - z/OS Java Batch Applications
 - S8370 - z/OS Tomcat Exploring

Java[™] on z/OS[®]

- The IBM Java Developer Kits provide a Java VM and runtime environment for all z/OS Java-enabled subsystems, including WebSphere Application Server, CICS, DB2, and IMS.
- Currently five versions that run on z/OS:
 - IBM Developer Kit for OS/390, Java 2 Technology Edition, Version 1.3.1
 - IBM SDK for z/OS, Java 2 Technology Edition, Version 1.4
 - IBM 64-Bit SDK for z/OS, Java 2 Technology Edition, Version 1.4
 - IBM SDK 5.0 for z/OS, 31-bit and 64-bit versions
- The IBM JDKs also provide the standard *java* command line launcher for Unix System Services.
- The BPXBATCH utility can be used to run the *java* command line in a batch job (**Not recommended**)
- zAAP can significantly reduce the cost of running z/OS Java workloads

IBM JZOS Press Release

ARMONK, N.Y. –February 14, 2006 – IBM announced today its purchase of the JZOS batch technology developed by Dovetailed Technologies, LLC. The JZOS toolkit, which provides a simple, easy-to-use facility for running Java batch applications on the IBM mainframe flagship operating system, z/OS, is now available as a complimentary download on IBM alphaWorks.

... IBM intends to integrate JZOS technology into its SDK for z/OS, Java 2 Technology Edition products in

2006. The existing technology is available now at no charge at <http://www.ibm.com/alphaworks/tech/zosjavabatchtk>

IBM JZOS Batch Toolkit for z/OS SDKs

- A Java VM **Launcher** that runs as a traditional MVS program (jobstep or started task)
 - Java System.in, System.out, and System.err routed to DDs that support MVS datasets or JES spool files.
 - Controls output encoding separate from default JVM encoding (..more later)
 - Launches the Java VM in the original batch address space
 - Simplifies job accounting and WLM profiles
 - Can use job step DD names
 - Pass condition code from Java `System.exit(--code--)`
 - Flexible/scriptable configuration of environment variables and settings

IBM JZOS Batch Toolkit for z/OS SDKs

- A Java Native Interface (JNI) library **Toolkit**
 - Native "Wrappers" for "C" Library IO routines
 - Text/Binary/Record mode access to MVS datasets
 - FileFactory class for portable use of text files/datasets
 - Wrappers for other sytem services
 - Access jobname, userid, environment variables, ...
 - Text-command based dynamic allocation via BPXWDYN
 - Communicate with MVS console (WTO, START, STOP, MODIFY)
 - More to come...

You say codepage, I say charset

- Internally, Java VM Strings and chars are (always) Unicode
- The **file.encoding** system property:
 - Specifies the platform's file system character set encoding
 - Determines the *default* charset set when converting unicode characters to/from bytes
- It's common for Java code to (incorrectly) assume that file.encoding is ASCII
 - It's probably best when this happens just to run the JVM with `file.encoding="ISO-8859-1"` (ASCII/Latin)
 - But we (usually) don't want ASCII output!.... JZOS Launcher manages separate encoding.
- www.unicode.org/reports/tr17/ - "A character map may also be known as a charset, a character set, a code page"

IBM JZOS Batch Launcher Example

```
//JZOSBAT JOB (999,XXX), 'JAVA JZOS', CLASS=A, MSGLEVEL=(1,1)
```

```
// MSGCLASS=X,REGION=0M,NOTIFY=&SYSUID
//JAVAJVM EXEC PGM=JZOSVM14,
// PARM='com.foo.MyClass arg1 arg2'
//STEPLIB DD DSN=JZOS.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
//STDENV DD *
. /etc/profile
. ~/.profile
export CLASSPATH=~/.myapp
for i in ~/.myapp/lib/*.jar; do
    export CLASSPATH=$i:$CLASSPATH
done
//
```

File I/O

- Standard java.io package can only access zFs/HFS (Unix) files
- IBM Record IO (JRIO) classes are higher-level abstractions, but can be difficult to use as source is not available and documentation is limited.
- JZOS "ZFile" class provides thin JNI wrapper for "C" library I/O routines, which are well tested and documented. Java Exceptions are provided to access to z/OS diagnostic codes.

```
ZFile zFile = new ZFile("//DD:INPUT", "rb,type=record,noseek");
try {
    byte[] recBuf = new byte[zFile.getLrecl()];
    int nRead;
    while((nRead = zFile.read(recBuf)) > 0) {
        String line = new String(recBuf,0,nRead,
                                ZUtil.getDefaultPlatformEncoding());
        System.out.println(line);
    };
} finally {
    zFile.close();
}
```

Portable File I/O

"C" I/O routines allow MVS datasets to be processed in "text stream" mode
 Portable code to process either PC/Unix files or datasets (on z/OS)

```
import com.dovetail.jzos.FileFactory;
...
BufferedReader brdr = FileFactory.newBufferedReader(inFileName);
BufferedWriter bwtr = FileFactory.newBufferedWriter(outFileName);
try {
    String line;
    while ((line = brdr.readLine()) != null) {
        bwtr.write(line);
    }
}
```

```

        bwtr.newLine();
    }
} finally {
    brdr.close();
    bwtr.close();
}

```

File names, at runtime could be: "c:\mypath.txt", "//MVS.DATASET.NAME", or "//DD:DDNAME".
The FileFactory will select either a ZFile or a java.io.File as appropriate

MVS Console interface

- WTO

```

ZUtil.wto("FOO1233E processing terminated",
          0x0020, // routecde
          0x4000); // descriptor code

```

- Operator commands

- START - Java started task can access options
- STOP(P) - Causes the JZOS batch launcher to issue a Java System.exit()
- MODIFY(F) - Can send commands into your Java apps via a callback interface

Deploying z/OS Java Batch Apps

Runtime components of a batch Java application

- zFS/HFS files
 - (Binary) ".class" files
 - (ASCII) ".properties", configuration XML, and other "classpath resource" files
 - JAR files (ZIP archives of class and resource files)
- MVS PDS members
 - JCL
 - "control cards", "parms", etc

Where to write and compile your code

- on z/OS ?
 - ISPF ?
 - vi ?
 - ...
- Or...on your workstation
 - Your favorite editor + ANT scripts to build and deploy
 - an IDE

- Syntax/code assisted typing
- Code hyper-navigation
- Integrated (remote) debuggers
- Ant integration, wizards, tools

Eclipse

- A *great* open-source Java IDE.
www.eclipse.org
- The basis for the WebSphere Studio, Rational Developer, other IDEs
- Code, compile, and test on workstation...
 - Code-assist editors
 - Models code structure; hyper-navigation
- ...one-click deploy your application to z/OS using Ant

Eclipse setup and usage

- Download, install, and configure Eclipse (see how-to at dovetail.com)
- Download the `BatchSample` Eclipse project from dovetail.com
- Customize `zos.properties` and move to the workspace directory
- Edit and compile (and unit-test) your code
- Run `deploy.xml` to deploy Java application Jars, JCL, etc. to z/OS
- Submit JCL
- Can be extended to handle web applications, see `TomcatSample` project and "S8369 z/OS Tomcat Exploring"

Common z/OS Java Stand-Alone Recipes

- XML processing with Java jobsteps interleaved into traditional batch schedules
- Java jobsteps as web-services client
- Apache Tomcat as a JSP/Servlet container
 - Web-based UI for z/OS apps and tools
 - Soap web services server
 - ...
- DB2 batch processing
- Long-running Java started tasks
- ...

Recipe: XML Processing with Java Jobs

No time to prepare a complicated application?...

Quickly whip up a Java program to convert field-structured datasets to and from XML.

Ingredients:

- IBM JZOS Batch Launcher
- Java XML parser (built-in SDK 1.4+)
- Open source XML parsing helpers, like Apache Commons "Digester"
- Stir in a tiny bit of your own code...
- Serve in a batch job, exchanging datasets and XML with other batch jobsteps

Recipe: XML Processing Example

The Apache Commons XML Digester makes XML parsing easy:

```
InputStream inputStream = FileFactory.newInputStream(inputFilename);
Digester digester = new Digester();
digester.push(this);
digester.addCallMethod("congress/member", "handleCongressMember", 7);
digester.addCallParam("congress/member/district", 0);
digester.addCallParam("congress/member/name", 1);
digester.addCallParam("congress/member/party", 2);
digester.addCallParam("congress/member/contact/address", 3);
digester.addCallParam("congress/member/contact/phone", 4);
digester.addCallParam("congress/member/contact/district-phone", 5);
digester.addCallParam("congress/member/contact/url", 6);
digester.parse(inputStream);
```

The Apache Digester uses SAX event parsing, so it's efficient even for very large XML documents.

Recipe: XML Processing Example

```
//JOBNAME JOB (), 'ME',MSGCLASS=H,NOTIFY=&SYSUID
//PROCLIB JCLLIB ORDER=
//*
//PARSEXML EXEC PROC=EXJZOSVM,
// JAVACLS='com.dovetail.jzos.sample.ParseXmlToFile',
// ARGS='//DD:XMLIN //DD:OUTFILE'
//XMLIN DD DSN=MY.XML.DATA,DISP=SHR
//OUTFILE DD DSN=MY.RECORDS,DISP=(NEW,CATLG),
// DCB=(RECFM=VB,LRECL=200,BLKSIZE=0),
// SPACE=(CYL,(1,1),RLSE)
//STDENV DD *
```

...

Recipe: Java job as web service client

Need to order some take-out for your hungry batch users?

Quickly write a Java batch program that calls a SOAP Web service.

- Get the "WDSL" (XML interface description) of your favorite tasty web service.
- With the free Apache Axis Web services toolkit, use "WSDL2Java" to generate the client Java "stub"

code to call the web service.

- Write a tiny Java program to run under the JZOS batch launcher...
 - Call the web service using the generated stub proxy....(the SOAP HTTP protocol and XML envelope parsing is done for you!)
 - Write the results to a dataset, DB2 database, etc...
- Try this yourself! - S8369 z/OS Batch Java Applications

Recipe: z/OS JSP/Servlet Container

Tired of EJB dog food? Try Apache Tomcat!

- A ubiquitous open source, pure-Java Servlet / JSP container
- Can be bundled/embedded in a turnkey Java web application
- Can be configured for DB2 JDBC connectivity and SAF authentication and authorization
- Not a full J2EE container -- no EJBs :-)
- No Sysplex clustering, etc
- Nicely supports modern web applications using JSP, STRUTS, Spring/Hibernate, etc
- Can act as a SOAP Web Services container using Apache Axis.
- An "on-ramp" to WebSphere, since....

Mainframe Tomcat in an hour or less!

1. Download and install [JZOS](#)
2. Download and install [Tomcat](#)
3. Tailor and submit the JZOS Tomcat JCL (just as easily a started task)
4. Open for business!
5. Graceful shutdown via MVS STOP command (SDSF "Y")

- Step-by-step instructions at www.jzos.com/docs
- Try it yourself!... SHARE Session #8370: Lab: z/OS Tomcat Exploring

Recipe: Java DB2 Batch Processing

Recipe, or Ingredient?

- IBM has over the years made significant improvements in the DB2 JDBC drivers
 - The new "Universal Driver" performs well on z/OS, in either "Type-2" or "Type-4" mode.
- Stable high-quality open-source JDBC frameworks can **significantly** benefit database applications
 - [Apache iBatis](#) - lightweight "Data Mapper" framework tames SQL by separating from Java code.
 - [Hibernate](#) - Full featured Object/Relational mapping framework
 - ...many others

Recipe: Long-running Java Jobs/STCs

- Tomcat is only one example, others seen in the wild include...

- Messaging / file transfer server for proprietary TCP/IP Socket protocol
 - Socket programming in Java is a sweet and reliable addition to many entrees.
 - JZOS toolkit interface to BPXWDYN facilitates dynamic dataset allocation.
 - DB2 tables for tracking status, queueing requests
- Long running task that polls DB2 table for new "work" in the form of datasets to process (transform to XML, etc)
- "MQ" message server
 - Queued/Asynchronous DB2 database updates initiated on other platforms
 - Reuse Java "JMS" api programming skills on z/OS

SDK 5.0 Shared Classes

- Facility with allows multiple JVMs (jobs) to share Java classes
- Cached in named shared memory, and dynamically updated
- Can reduce overall memory footprint
- Can reduce JVM startup time
- Caches can be private or shared across groups of users/jobs

For example, when using shared classes, Tomcat 5.5 can be started in *half* the time.

-->

In Summary

- z/OS Java has escaped Websphere; it's running loose in JCL!
- Open source Java software can do amazing things on z/OS
- Free development tools, like Eclipse, are fantastic for z/OS Java development
- **Reuse is King** - code, skills, investments...

References

- JZOS batch launcher and toolkit - <http://www.ibm.com/alphaworks/tech/zosjavabatchtk>
- Dovetailed Technologies - <http://dovetail.com>
- IBM developerWorks Article: Java Batch Jobs on z/OS and OS/390 <http://www.ibm.com/developerworks/eserver/library/es-java-batchz.html>
- IBM's home page for Java on z/OS - <http://www.ibm.com/servers/eserver/zseries/software/java>
- The zSeries Application Assist Processor (zAAP) - <http://www.ibm.com/servers/eserver/zseries/zaap>
- IBM's WebSphere Application Server for z/OS home page - http://www.ibm.com/software/webservers/appserv/zos_os390

References - continued

- z/OS Unix System Services home page - <http://www.ibm.com/servers/eserver/zseries/zos/unix/>
- z/OS Unix System Services users guide: Using BPXBATCH - <http://publibz.boulder.ibm.com/cgi-bin/bookmgr OS390/BOOKS/bpxza450/11.2>
- The Java Record IO package, which is part of the IBM SDK - <http://www.ibm.com/servers/eserver/zseries/software/java/jrio/overview.html>
- The z/OS C/C++ Programming Guide - <http://publibz.boulder.ibm.com/epubs/pdf/cbcpg130.pdf>
- Sun's documentation on the Java JNI Launcher interface - <http://java.sun.com/j2se/1.4.2/docs/guide/jni/>
- IBM's article on Writing a simple JNI program on OS/390 - <http://www.ibm.com/servers/eserver/zseries/software/java/os390jni.html>