Ask the Experts
All About OSGi JVM™ servers in CICS®

21 October 2014

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS™

Agenda

- Introduce the panel of experts

- Introduce CICS Transaction Server "All About OSGi JVM servers in CICS" Topics

- Answer 6 question submitted by panel

- Answer 4 questions submitted by email

- Open telephone lines for questions

- Summarize highlights

Panel of Experts

Panelist	Role at IBM
Ivan D. Hargreaves IVANH@uk.ibm.com	**Technical Leader and Lead Developer of the JVM server and Liberty Profile technologies in CICS** Ivan has been lead designer and developer of the JVM server technology in CICS since its inception. His role is to facilitate modern Java applications in CICS to run in an industry standard way while leveraging the power of the CICS platform and seamlessly integrating with existing CICS capabilities, data and applications.
Dave Roberts david.roberts@uk.ibm.com	**Software Engineer, CICS Transaction Server for z/OS Development** Dave has over 25 years experience with IBM mainframes in Application Development, Development Support, Systems Programming and for the last 8 years in the IBM UK Hursley Laboratory. He is currently a Developer in the CICS TS Delivery team working on the integration of WebSphere Liberty profile into CICS JVM server.
Shayla Robinson shayla@us.ibm.com	**Advisory Software Engineer, CICS/CPSM Level 2 Support** Shayla is an Advisory Software Engineer in the United States. Shayla has been with IBM for 17 years. She spent the first 10 years as a CICS Level 2 Technical Support representative in Research Triangle Park, North Carolina before moving to CPSM Level 2 Support. Over the years Shayla has delivered several presentations and demonstrations on using CICS and Java.
Phil Wakelin Phil_Wakelin@uk.ibm.com	**Software Engineer, CICS Transaction Server Strategy and Planning** Phil works for IBM UK in Hursley, and is a member of the CICS strategy and planning team. He has worked with many CICS technologies for the last 20 years, and is currently responsible for planning new functionality in the area of CICS Java support. He is the author of many white papers, SupportPacs and IBM Redbooks in the area of CICS Transaction Processing, and is a certified IBM Certified System Programmer - CICS Transaction Server V5.2 for z/OS.

Introduction

- We will be covering a number of questions on the general topic:
  All About OSGi JVM servers in CICS

  - 1) Setting up an OSGi JVM server
  - 2) Why do I need to migrate to an OSGi JVM server
  - 3) Moving applications to an OSGi JVM server
  - 4) Improving JVM server performance
  - 5) Automating deploying Java applications
  - 6) Troubleshooting Java applications
  - 7) Debugging the JVM server
  - 8) Ensuring the portability of applications
  - 9) Best practice for connecting to databases
 - 10) Batch jobs in CICS JVM server

Question 1

- How do you set up and configure a JVM server to run OSGi applications?

JVM server configuration parameters

- **CICS SIT parameters**
  - USSHOME – zFS directory of CICS USS components
  - JVMPROFILEDIR – zFS directory for JVM Server profiles
- **LE Runtime Options**
  - Default DFHAXRO
  - Specifies runtime options to control program processing in LE enclave
- **JVMSERVER Resource Definition**
  - Encapsulates JVM and provides control over its lifecycle
- **JVM Profile**
  - JVM server options
    - Defines JVM server runtime configuration
    - Required parameters:
      - JAVA_HOME
      - WORK_DIR
    - Configuration parameters:
      - OSGI_BUNDLES,
      - OSGI_FRAMEWORK_TIMEOUT
      - STDIN, STDERR, STDOUT and JVMTRACE
      - APAR PI19204 (STDOUT=//DD:) for JES routing
  - JVM Options
    - Defines JVM command-line options & system properties
 - USS Environment Variables
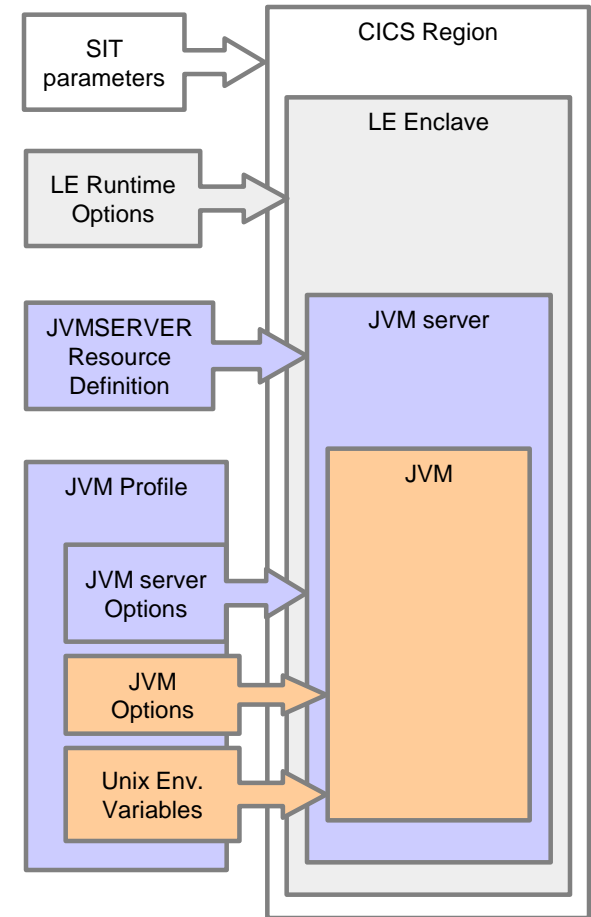
Java™

Minimum Steps to Setup a JVM server

- Edit CICS JCL
  - add SDFJAUTH to the STEPLIB
  - include JVMPROFILEDIR SIT parameter pointing to JVM server profile location
- Edit JVM server profile (DFHOSGI sample) JAVA_HOME and WORK_DIR
- Create JVMSERVER resource definition using CEDA or CICS Explorer
- Install and Enable JVMSERVER into CICS
  - A JVM is only created when the JVMSERVER is enabled

The JVM server is now ready to run CICS Java applications!

Install JVMSERVER

Set ENABLED

Run Java Applications

JVM

Set DISABLED

Discard JVMSERVER
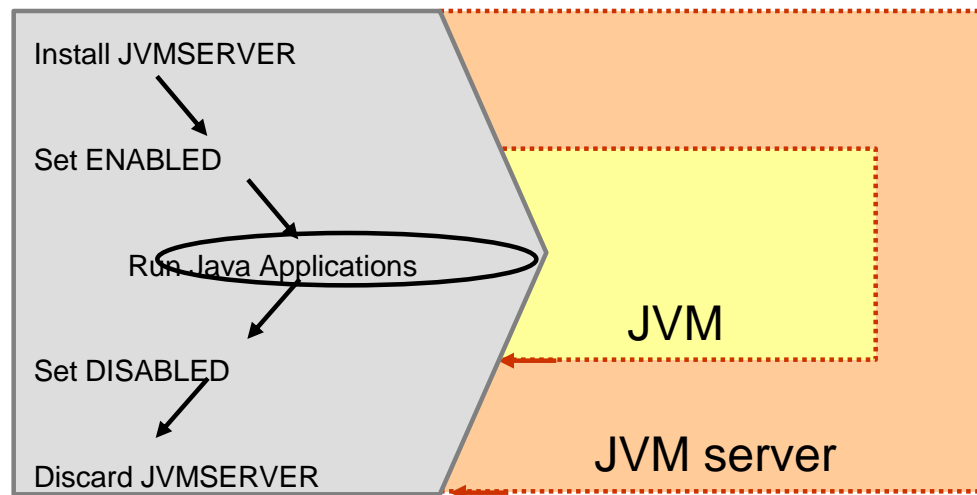
JVM server

Question 2

- Why do I need to migrate from JVM pool to an OSGi JVM server in CICS?
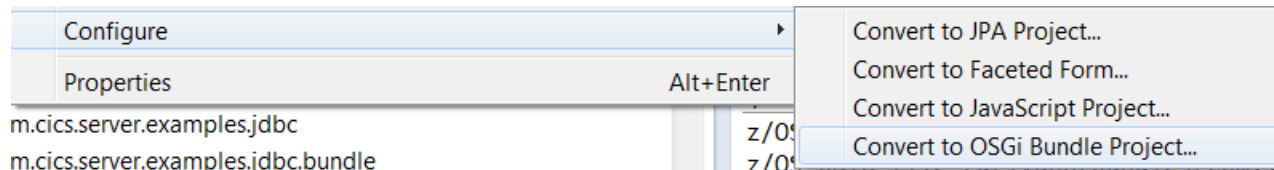
Migration from JVM pool

1. CICS TS V5.1 removed support for JVM pool and EJB™/CORBA in CICS

- CICS TS V4.2 supports both JVM pool and JVM server
- Validate your Java application in a V4.2 JVM server before you move to CICS TS V5

2. JVM server is a standard JVM, so:
- Better portability
- Better scalability

3. OSGi provides key application advantages
- Support for dynamic updates to applications without JVM restart
- Ability to run multiple version of components within the same JVM

Question 3

- How do you migrate existing Java applications from the CLASSPATH to OSGi bundles?

Convert Java Projects to OSGi bundles

- Using Eclipse and CICS Explorer SDK, convert your existing Java projects into OSGi bundles (right-click *configure | Convert to OSGi Bundle Project*)

Configure	►	Convert to JPA Project...
Properties	Alt+Enter	Convert to Faceted Form...
m.cics.server.examples.jdbc	z/O!	Convert to JavaScript Project...
m.cics.server.examples.jdbc.bundle	z/O!	Convert to OSGi Bundle Project...

- Create a new **CICS bundle project**

- Include the OSGi bundle in CICS bundle (or add OSGi bundle project reference)

- Right click, *Export Bundle Project to z/OS® Unix® File System, choose unique location on z/FS*

- Create a new CICS bundle definition in CICS, set the bundle-path, and the install CICS bundle

Avoid common migration pitfalls

- **OSGi bundle conversion technique proves too complex** due to 3rd party JAR files:
  - Seek OSGi compliant versions of those 3rd party libraries
  - Consider wrapping existing JAR files inside an OSGi bundle. Place them on the Bundle-classpath (see Manifest.mf) and expose the packages with an explicit Export declaration
  - Alternatively use File | New | Other | Plug-in from Existing JAR Archives

- **Existing Java classes are not threadsafe**. Things to look for:
  - The use of static variables. They will not be reset across application calls
  - Objects shared between threads. Ensure synchronised keyword is used appropriately
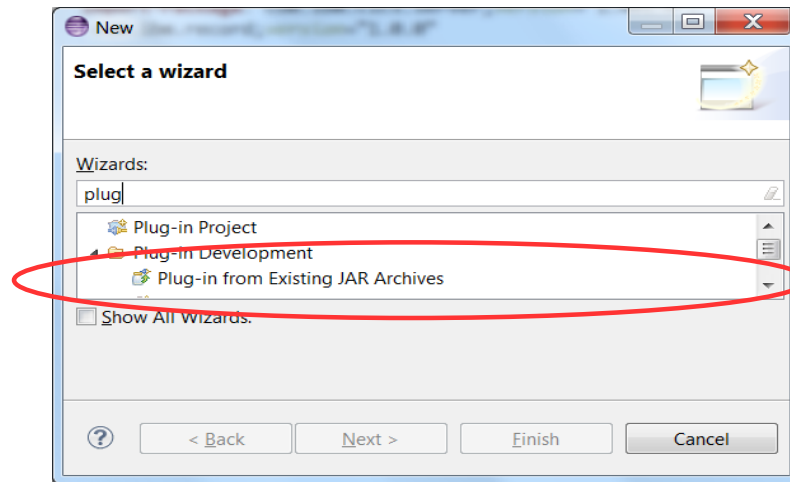
Avoid common migration pitfalls (continued...)

- **You have a common Library of functions used by more than one application and want it to be installed and available to all applications**
  - Install the bundle using the 'middleware bundles' option. OSGI_BUNDLES=mybundle.jar in the JVM profile

- **You experience class-loader issues with third-party Java components (Perhaps using Spring, or Log4j)**
  - Ensure APAR PI25045 is installed, allowing CICS to manage your Thread Context Class Loader (TCCL)
  - Or, set (and reset) the Thread Context Class Loader to the application's class-loader manually around the 3rd party API calls that require it

```
ClassLoader cachedcontextClassLoader = Thread.currentThread().getContextClassLoader();
try
{
    ClassLoader classLoader = application.class.getClassLoader();
    Thread.currentThread().setContextClassLoader(classLoader);
    // 3rd party API
}
finally
{
    Thread.currentThread().setContextClassLoader(cachedcontextClassLoader);
}
```

Question 4

- How do you tune a JVM server for better performance?

Obtaining JVM server tuning information

- **Run a typical peak workload**

- **Use standard CICS processes to obtain JVMSERVER resource statistics** (see next slide for reference)
  - Check there were no excessive Thread waits (increase thread limit)
  - Check Heap occupancy – if high, increase JVM HEAP
  - If Heap occupancy is not approaching HEAP max and GC happens often - it may indicate there are many transient objects in your Java application.

- **Use IBM HealthCenter**
  - Eclipse based, low runtime overhead
  - *Garbage collection,* saw-tooth peaks increasing? Potential memory leak
  - *Profiling*, may help address CPU hotspots within user application

- **Advanced: Re-compile LE runtime options setting REPORT(YES)**
  - LE produces a detailed storage report, you may consider tuning the LE sub-pools based on this advice – exercise caution.

JVMSERVER statistics (SMF 110)

```
JVMSERVER Name. . . . . . . . . . :   IVANSRVR
JVMSERVER Enable Status . . . . . :   Enabled
JVMSERVER JVM profile name. . . . :   IVAN64
JVMSERVER LE runtime options. . . :   DFHLERO
JVMSERVER use count . . . . . . . :        15
JVMSERVER thread limit. . . . . . :       256
JVMSERVER current threads . . . . :         0
JVMSERVER peak threads. . . . . . :         8
JVMSERVER thread limit waits. . . :         0
JVMSERVER thread limit wait time. :   00:00:00.00000
JVMSERVER current thread waits. . :         0
JVMSERVER peak thread waits . . . :         0
JVMSERVER system thread use count :       103
JVMSERVER system thread waits . . :         0
JVMSERVER system thread wait time :   00:00:00.00000
JVMSERVER current sys thread waits:         0
JVMSERVER peak system thread waits:        0
JVMSERVER JVM creation time . . . :   11/26/2010 11:30:12.85130
JVMSERVER current heap size . . . :     3,523K
JVMSERVER initial heap size . . . :       16M
JVMSERVER maximum heap size . . . :       12G
JVMSERVER peak heap size. . . . . :     4,412K
JVMSERVER heap occupancy. . . . . :     1,117K
JVMSERVER Garbage Collection (GC) :-Xgcpolicy:gencon
JVMSERVER no. of major GC events. :         0
JVMSERVER Elapsed time in major GC:         0
JVMSERVER Heap freed by major GC. :         0
JVMSERVER no. of minor GC events. :         2
JVMSERVER Elapsed time in minor GC:         5
JVMSERVER Heap freed by minor GC. :     6,624K
```

JVMSERVER resource

Thread information

JVM Heap

JVM Garbage collection

Question 5

- How do you automate the deployment of a Java application in an OSGi bundle into a production CICS system?
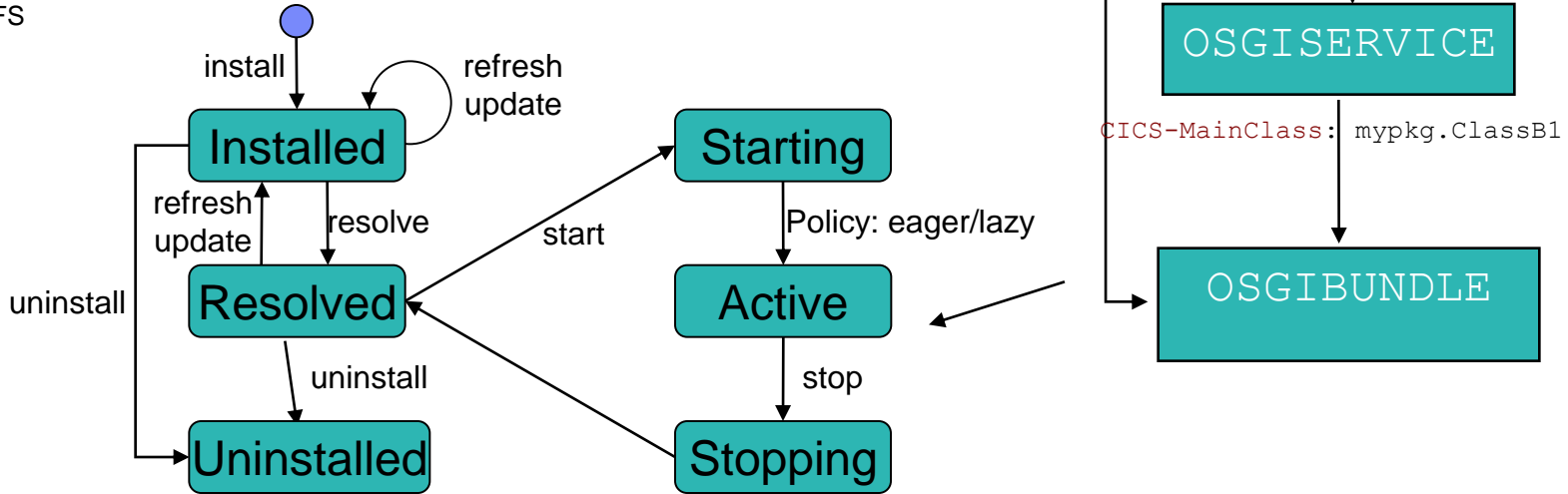
CICS Bundle lifecycle for OSGI bundles

BUNDLE

```
BUNDLEDIR=/var/cicsts/bundledir
com.ibm.cics.bundle
```

PROGRAM

Java Class = mypkg.ClassB1
JVMSERVER=MYJVM

INSTALL

ENABLE

Bundle **enable**
causes activation of
OSGi bundle JAR
from zFS

`BUNDLEPART – B1`
`PARTYPE=OSGIBUNDLE`

`OSGISERVICE`

`CICS-MainClass: mypkg.ClassB1`

`OSGIBUNDLE`

install

refresh
update

Installed

refresh
update

resolve

uninstall

Resolved

uninstall

Uninstalled

Starting

start

Policy: eager/lazy

Active

stop

Stopping

OSGi bundle versioning process

1. Export CICS bundle project from Eclipse with new version of OSGi bundle to zFS

2. Create/update BUNDLE definition to refer to zFS location of #1 in bundledir

3. Install/enable new BUNDLE definition

4. Disable old BUNDLE definition

5. Discard old BUNDLE definition (optional)

* Updated OSGi service active as soon as bundle enabled

*Note: Activation of versioned OSGi bundle on enable assumes PTF for APAR PI24367 has been applied

Automating Java BUNDLE versioning

Step	BUNDLE	OSGi bundle	zFS bundledir
1. Startup	A0 installed	A V1.0	/bundlehome/A/1.0
2. Patch n	A1 created dynamically	A V1.n	/bundlehome/A/1.n
3. Patch n+1	A2 created dynamically A1 disabled/deleted	A V1.n+1	/bundlehome/A/1.n+1
4. Cold start after patch n	A0 installed	A V1.n	/bundlehome/A/1.n

Small **wait** advised after CREATE and SET to ensure all bundle parts activated

1. A0 is installed by CSD grouplist install
2.
   2a. Export new CICS bundle project v1.1 to zFS /bundlehome/A/1.1
   2b. CREATE BUNDLE(A1) with bundledir=/bundlehome/A/1.1
   2c. SET BUNDLE(A1) ENABLE
   2d. INQUIRE BUNDLE(A1), if enabled then
       SET BUNDLE(A0) DISABLED
       DISCARD BUNDLE(A0)
       Repeat 2b-d in all CICS regions
   2e. Update CSD BUNDLE defn A0 to point to /bundlehome/A/1.1 for next cold start
3. As #2 with BUNDLE A2 with bundledir=/bundlehome/A/1.2

OSGi service V1.0 *active*

OSGi service V1.1 *active*

OSGi service V1.0 *removed*

Question 6

- What are the common steps to troubleshoot JVM servers and Java applications?

JVMSERVER resource does not enable

⚠ ■ Messages in the CICS logs:
  ■ MSGUSR – CICS messages

```
DFHSJ0210 10/13/2014 11:05:44 IYK3ZDRD An attempt to start a JVM for the JVMSERVER resource DFH$JVMS has
failed. Reason code: JVMPROFILE_ERROR.
DFHSJ0914 E 10/13/2014 11:05:44 IYK3ZDRD ROBERTD JVMSERVER DFH$JVMS is DISABLED because the Language
Environment Enclave was not created.
```

  ■ SYSPRINT or SYSnnnnn - JVMSERVER start-up messages

```
2014/10/13 10:05:44.494156 UTC DFH$JVMS *Exc*: Attempt to open JVM profile DFHOSGI has failed. Runtime error
message = EDC5129I No such file or directory.
```

⚠ ■ JVM logs on the UNIX file system
  ■ JVMPROFILE properties:
  ■ WORK_DIR, STDOUT, STDERR and JVMTRACE

⚠ ■ CICS trace – AP=ALL and SJ=ALL
  ■ Re-check the logs and AUX™ trace for exceptions

The application is not available

⚠️ ▪ **Messages and exceptions in the CICS and JVM logs**

```
DFHSJ1004 10/13/2014 12:33:12 IYK3ZDRD CICSUSER An exception has been thrown by the route method of class
com/ibm/cics/router/Router running in JVMSERVER JVMS1. Exception 'No CICS-MainClass services with the classname
'com.ibm.myprog.Main' could be found. Either the implementing Bundle has not been enabled, or the MANIFEST of
the Bundle does not contain a CICS-MainClass (Entry point) for this class.'.
```

▪ **Did the CICS BUNDLE resource install?**

```
DFHRL0103 E 10/13/2014 12:37:24 IYK3ZDRD CEDA The CICS resource lifecycle manager failed to create the BUNDLE
resource MYBUN1 because the manifest /wrong/directory/name/META-INF/cics.xml was not found.
```

▪ **Did the CICS-MainClassService get registered?**

```
2014/10/13 11:44:37.442000 UTC I: Registered 'CICS-MainClass' service for class name
'com.ibm.cics.server.examples.jdbc.JDBC3270' with alias 'com.ibm.cics.server.examples.jdbc.JDBC3270'.
2014/10/13 11:44:37.443000 UTC I: Registered 'CICS-MainClass' service for class name
'com.ibm.cics.server.examples.jdbc.JDBC3270' with alias 'JDBC3270'.
```

▪ **CICS Explorer**

OSGi Services ⊠

CNX0211I Context: IYK3ZDRE. Resource: OSGISERV. 2 records collected at 13 Oct 2014 12:51:28

Service Name	OSGi Bundle	Ver...	Servic...	Bu...	Bundle Part
com.ibm.cics.server.examples.jdbc.JDBC3270	com.ibm.cics.server.examples.jdbc	1.0.1	ACTIVE	JDBC	com.ibm.cics.server.examples.jdbc
JDBC3270	com.ibm.cics.server.examples.jdbc	1.0.1	ACTIVE	JDBC	com.ibm.cics.server.examples.jdbc

The application does not work as expected

- **Messages in the CICS logs:**
  - MSGUSR – CICS messages

```
DFHSJ0904 10/13/2014 13:13:55 IYK3ZDRE CICSUSER TC11 TRN1 MYPROG1 Exception 'java.lang.NullPointerException'
occurred creating object reference for class com.ibm.cics.dummy.BadProgram.

DFHAC2236 10/13/2014 13:13:55 IYK3ZDRE Transaction TRN1 abend AJ05 in program MYPROG1 term TC11. Updates to
local recoverable resources will be backed out.
```

- **Java exceptions in dfhjvmtrc and dfhjvmerr**

```
java.lang.NullPointerException
at com.ibm.cics.dummy.BadProgram.main(BadProgram.java:12)
at . . .
at com.ibm.cics.router.Router.route(Router.java:1064)
```

- **EDF for the JCICS API calls and SQL calls**
- **Java Interactive Debugger**
  - `-agentlib:jdwp=transport=dt_socket,server=y,address=28219`
  - http://www-01.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.cics.ts.java.doc/topics/dfhpj94.html
- **OSGi Console to look at bundles and services**
  - `-Dosgi.console=winmvs2c.hursley.ibm.com:28210`
  - `-Dosgi.console.encoding=US-ASCII`

Question 7

- How would it be possible to debug (manual step-by-step execution) OSGi bundles on a single JVM in CICS by multiple developers? As far as I know, the JVM in CICS supports only one debug session at a time. This of course leads to decreased efficiency of developers.

Multi-developer, single server debug

- This is a simple No – The JPDA doesn't support multiple clients attached to a Java debugger, and its not designed for debugging multiple threads (CICS tasks)

- However you can run multiple JVM servers in a single CICS region one for each debugging client.

Alternatives to multi-developer debug

- Ensure that each developer gets their own JVM server for debug purposes.

- Mocking is another possible approach. Applications can run locally and 'stub' the CICS calls with temporary local calls. The JVM server is not required during this phase of development so independent debugging can continue. Frameworks such as Mockito and EasyMock exist to help with local mocking.

- A future consideration for CICS could be to 'remote' the JCICS API, such that Java application code could run locally - submitting only portions of the CICS API to run in a CICS JVM server. However, this capability does not exist today.

Question 8

- How to ensure portability of bundles, e.g. between JVM in CICS or JVM on WebSphere Application Server (WAS), when bundles use CICS specific API (JCICS services)? Are there best practices and guidelines?

Ensuring the portability of applications

- Yes there are best practices,

- 1) Encapsulate your CICS specific API in another OSGi bundle by implementing a common interface. The main development bundle can call the interface (perhaps via an OSGi service) and get the CICS specific version when in CICS, and a WAS specific version when in WAS.

- 2) Bundle fragments are also used to provide different implementations for different hosts. Encapsulating the CICS API or equivalent WAS operations in bundle fragments and installing the relevant implementation, may offer another solution.

Question 9

- What are the best practices to connect with database? What is the preferred approach, e.g. JDBC™, JPA™ (Hibernate), or perhaps, SQL-J?

Best practice for connecting to databases

- JDBC is the preferred approach

- JPA will use JDBC under the covers

- SQL-J is supported, but increasingly less popular, more complex to use, and doesn't offer as much performance advantage as it used to

Question 10

- Is it recommended to run a batch job in a CICS JVM? What should I pay special attention to, when planning to run a batch in CICS JVM?

Batch jobs in CICS JVM server

- CICS TS Feature Pack for Modern Batch V1.0 enables you to run batch applications inside the CICS JVM server in parallel with your existing CICS applications, combining your batch and online workloads for increased availability. This reduces the need to take critical CICS resources offline in order to process bulk updates, and helps you to efficiently manage your batch processing.

- An external job scheduler in WebSphere Application Server is used to control the submission of Batch jobs.

- If you are using long running tasks to process batch work in a CICS JVM server then CICS standard best practices apply:
    - Use Threadsafe APIs for efficiency
    - Use regular syncpoints
    - Give priority to online transactions/regions

Open Lines for Questions

Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line "wste subscribe" to get a list of mailing lists and to subscribe

2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

Summary

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:
  http://www.ibm.com/software/websphere/support/supp_tech.html

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
  http://www.ibm.com/developerworks/websphere/community/

- Join the Global WebSphere Community:
  http://www.websphereusergroup.org

- Access key product show-me demos and tutorials by visiting IBM® Education Assistant:
  http://www.ibm.com/software/info/education/assistant

- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:
  http://www.ibm.com/software/websphere/support/d2w.html

- Sign up to receive weekly technical My Notifications emails:
  http://www.ibm.com/software/support/einfo.html