WebSphere Liberty Profile and WebSphere Full Profile Classic – What’s New? And Common Problems

Session 17366
August 12, 2015
David Follis, Gary Picher, Mike Stephen
<table>
<thead>
<tr>
<th>Session</th>
<th>Title</th>
<th>Time</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>17363</td>
<td>Debug 101-Using ISA Tools for Apps in WebSphere Application Server z/OS</td>
<td>Monday 11:15</td>
<td>Europe 11</td>
</tr>
<tr>
<td>17367</td>
<td>WebSphere Liberty on Windows and z/OS (Among Other Things) Hands-On Lab</td>
<td>Tuesday 10:00</td>
<td>Asia 5</td>
</tr>
<tr>
<td>17361</td>
<td>ABCs of WAS</td>
<td>Tuesday 1:45</td>
<td>Oceanic 7</td>
</tr>
<tr>
<td>17368</td>
<td>z/OS Connect: Opening up z/OS Assets to the Cloud and Mobile Worlds</td>
<td>Tuesday 3:15</td>
<td>Oceanic 7</td>
</tr>
<tr>
<td>17362</td>
<td>Configuring Timeouts for WebSphere Application Server on z/OS</td>
<td>Wednesday 8:30</td>
<td>Oceanic 7</td>
</tr>
<tr>
<td>17366</td>
<td>WebSphere Liberty and WebSphere Application Server Classic - What's New?</td>
<td>Wednesday 11:15</td>
<td>Oceanic 7</td>
</tr>
<tr>
<td>17364</td>
<td>IBM Installation Manager for z/OS System Programmers: Web-based Installs, Fix Packs, and How iFixes Really Work</td>
<td>Thursday 4:30</td>
<td>Oceanic 7</td>
</tr>
<tr>
<td>17365</td>
<td>JSR 352 - The Future of Java Batch and WebSphere Compute Grid</td>
<td>Friday 10:00</td>
<td>Oceanic 6</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Agenda

- WAS 8.5.5 Family – Where are we now?
- WebSphere Liberty Overview
- WebSphere Liberty – What's New?
- WebSphere Classic – What's new?
- Common Problems – Liberty and Classic
Recent WAS History and Java EE

Going on 16 years of leading the industry and setting the bar for app Server functionality.

**WAS v8 GA**
- Mobile connectivity
- Easy Migration
- WAS / RAD Tooling Bundles

**WAS v8.5**
- WAS Liberty Profile (a new runtime architecture optimized for Cloud & Mobile)
- z/OS Extensions & Optimizations
- Intelligent Mgmt for scaling & resiliency included

**WAS v8.5.5**
- WAS Liberty Profile Updates for new open techs & prog. models
- JEE6 Web Profile
- WAS Liberty Core
- Caching included with WAS ND (WXS)

**WAS v8.5.5.x & Bluemix**
- Liberty Repository for continuous del.
- JEE7 elements and more
- Liberty Admin Ctr
- Enhanced Security to support Mobile & Cloud
- zOS Connect

**WAS v8.5.5.x & BlueMix**
- Java EE 7 certified
- Java SE 8
- WAS Docker
- WDT remote cloud deploy
- Liberty ND Autoscaling and dynamic routing
- WAS as a Service on Softlayer
- ...

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
WebSphere Liberty Overview
WebSphere Liberty v8.5.5

**WAS v8.5.5** delivers significant enhancements to Liberty profile, including new programming models, administration, qualities of service and development tools.

- **Dynamic Server Profile**
  Not static like Web Profile; configured by app at a fine-grained level.

- **"Developer First" Focus**
  Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

- **Start fast, run efficiently**
  Starts in <3s; Mem footprint <50MB; (TradeLite benchmark).

- **Integrated tools**
  Powerful tools in WDT Eclipse feature. Enhanced for v8.5.5 prog models, Maven integration, ++

- **Java EE7 Certified**
  Create apps for the Java EE7 Full Profile standard.

- **Unzip install and deploy**
  IM or unzip to install. New option to deploy "server package" of app + config + required subset of server runtime for highest density deploy.

- **Liberty Extensions**
  Add custom features and integrate 3rd party components via Liberty extensions interface.

- **Small Download**
  50MB for Web Profile features.

- **Dynamically Extensible**
  Install new features from repository (local or remote) with no svr restart.

- **Lightweight cluster Mgmt**
  Liberty servers can join a lightweight cluster for workload balancing and high availability.

- **Fidelity to full profile WAS**
  Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Continuous Delivery of new function

- Bluemix and beta drivers every month
- GA features delivered regularly via Liberty Repository
- Java EE7 features started 4Q2014
- Java EE7 Full Profile supported in 2Q2015

Java EE 7 Full Profile support

- servlet-3.1
- websocket-1.0
- jsonp-1.0
- openid-2.0
- couchdb-1.0
- + others

- jsp-2.3
- el-3.0
- spnego-1.0
- adminCenter+
- plus....
- -java 8
- -docker4dev

Jca-1.6
AdminCenter-1.0
Zconnect-1.0
+ others

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Continuous consumption!

Zero migration for unchanged apps on WAS Liberty, regardless of Java EE version

```
java -jar wlp-developers-runtime-8.5.5.2.jar
```

8.5.5.2

Bring your config

```
java -jar wlp-developers-runtime-8.next.jar
```

8.next

Choice: Features limit runtime behavior change
Existing features remain unchanged eg. servlet-3.0
New feature versions contain updates, eg. servlet-3.1
No custom properties for behavior switching

Point to your existing JRE
Within supported software levels. Java 6 will not last forever
New Java EE features may require higher versions of Java
Online Repository for Extending Liberty

- An **online repository** to deliver **Liberty platform extensions**
- Content includes:
  - Open source project integration
  - Individual Liberty features
  - Samples
- Enables more **fine-grained**, early access content, simplest possible integration with external open source
- Install using Liberty command-line tools or WDT/RAD

On [www.wasdev.net](http://www.wasdev.net)
WebSphere Liberty – What's New?
Optimized Local Adapters provides an efficient low-latency mechanism to exchange data bi-directionally between Liberty z/OS and other address spaces:

- Very efficient byte-array transfer
- First available with Liberty z/OS 8.5.5.2
- Bi-directional
  - Outbound -- Java in WAS invokes program in external
  - Inbound -- Program in external invokes Java in WAS
- Identity assertion to / from CICS
- Supplied JCA resource adapter for applications going outbound
- Supplied native APIs for cases where their usage is indicated
  - COBOL, C/C++, PL/I, High Level Assembler
  - 31-bit and 64-bit modules
- See WP101490 for "Primer" on API usage
- WP101490 Techdoc for much more
Integrating Java/Non-Java: WOLA

Enabling some of your GP work to be refactored to Java

WOLA is a technology that enables integration of Java and non-Java assets in a way that provides high throughput and security.
There are many points of similarity, but some points of difference:

**Similarities**

- Outbound JCA programming interfaces are the same
- Inbound native API programming model is the same
- CICS Link Server Task function very similar in design and operation
- Security assertion with CICS both directions supported
- Supplied samples nearly identical

**Differences**

- Global transaction not supported with Liberty WOLA
  - Applications that start global transactions will need to be modified before using with Liberty WOLA
- Target EJB for inbound must be EJB 3.x and has Liberty-specific design requirements
- Round-Robo and Alternate JNDI not supported
  - Relied on function of traditional WAS not present in Liberty
- Liberty WOLA and IMS not supported
- No WOLA MODIFY commands or SMF 120.10 for WOLA
z/OS Connect – What is it?

It’s about getting REST and JSON into your mainframe environment in a way that enables you to best take advantage of the assets that exist there:

REST – Representational State Transfer … the use of HTTP URLs that map to a ‘service’, such as ‘query account’ or ‘update data’

JSON – JavaScript Object Notation … a standard of representing data as a set of name/value pairs. This is passed back and forth along with REST request/responses

Where z/OS Connect fits

- Feature in Liberty Profile Server for z/OS
- Function IBM wrote to run in Liberty Profile
- No charge function provided with license entitlement to WAS z/OS, CICS or IMS

Mobile Ecosystem → z/OS Connect

CICS
IMS
Batch
Other
Why z/OS Connect?

This represents another component to configure and maintain in your environment. So what value does it bring?

- Provides a common and consistent entry point for mobile access to one or many backend systems
- Java, so runs on specialty engines
- Shields backend systems from requiring awareness of RESTful URIs and JSON data formatting
- Provides point for authorization of user to invoke backend service
- Provides point for capturing usage information using SMF
- Simplifies front-end functions by allowing them to pass RESTful and JSON rather than be aware of or involved in data transformation

You could enable Mobile access without z/OS Connect. z/OS Connect simplifies and makes the environment more consistent and manageable.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
One part of z/OS Connect is a servlet that runs in Liberty Profile z/OS.

A ‘Service Provider’ SPI can be implemented to provide connectivity to backend systems.

z/OS Connect provides the ability to transform JSON to the layout required by backend with a ‘Transformer’ SPI.

‘Interceptors’ are callout points where software can be invoked to do things such as SAF authorization and SMF activity recording.

Backend systems supported are CICS, IMS and Batch (DB2 announced).

z/OS Connect is software function that runs in WebSphere Liberty for z/OS.

z/OS Connect is described and configured in the Liberty server.xml file.

z/OS Connect is designed to accept RESTful URIs with JSON data payloads.
Java Road Map – Language Updates

Java 5.0
New Language features:
- Autoboxing
- Enumerated types
- Generics
- Metadata

Java 6.0
Performance Improvements
Client WebServices Support

Java 7.0
- Support for dynamic languages
- Improve ease of use for SWING
- New IO APIs (NIO2)
- Java persistence API
- JMX 2.x and WS connection for JMX agents
- Language Changes

Java 8.0
- Language improvements
  Closures for simplified fork/join

IBM Java 8 (J9 R28)
- Improvements in
  - Performance
  - RAS
  - Monitoring
    - z13™ Exploitation
    - SIMD
    - SMT
    - Crypto acceleration

IBM Java 7R1 (J9 R27)
Improvements in
- Performance
- RAS
- Monitoring
  - zEC12™ Exploitation
  - zEDC for zip acceleration
  - SMC-R integration
  - Transactional Execution
  - Runtime instrumentation
  - Hints/traps
  - Data Access Accelerator

IBM Java 6.0.1/Java 7 (J9 R26)
Improvements in
- Performance
- GC Technology
  - z196™ Exploitation
  - OOO Pipeline
  - 70+ New Instructions
  - JZOS/Security Enhancements

IBM Java 6.0 (J9 R24)
Improvements in
- Performance
- Serviceability tooling
- Class Sharing
- XML parser improvements
  - z10™ Exploitation
  - DFP exploitation for
    - BigDecimal
    - Large Pages
    - New ISA features

IBM Java 5.0 (J9 R23)
Improved performance
- Generational Garbage Collector
- Shared classes support
- New J9 Virtual Machine
- New Testarossa JIT technology
  - First Failure Data Capture
  - Full Speed Debug
  - Hot Code Replace
  - Common runtime technology
    - ME, SE, EE

IBM Java 6.0.1/Java 7 (J9 R26)
Improvements in
- Performance
- GC Technology
  - z196™ Exploitation
  - OOO Pipeline
  - 70+ New Instructions
  - JZOS/Security Enhancements

IBM Java 6.0 (J9 R24)
Improvements in
- Performance
- Serviceability tooling
- Class Sharing
- XML parser improvements
  - z10™ Exploitation
  - DFP exploitation for
    - BigDecimal
    - Large Pages
    - New ISA features

IBM Java 5.0 (J9 R23)
Improved performance
- Generational Garbage Collector
- Shared classes support
- New J9 Virtual Machine
- New Testarossa JIT technology
  - First Failure Data Capture
  - Full Speed Debug
  - Hot Code Replace
  - Common runtime technology
    - ME, SE, EE

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

**Timelines and deliveries are subject to change.**

- New Java8 Language Features
  - Lambdas, virtual extension methods

- IBM z13 exploitation
  - Vector exploitation and other new instructions
  - Instruction scheduling

- General throughput improvements
  - Up-to 7% better application throughput
  - Significant improvements to ORB

- Improved crypto performance for IBMJCE
  - Block ciphering, secure hashing and public key
    - Up-to 4x improvement to Public Key using ECC
    - CPACF instructions: AES, 3DES, SHA1, SHA2, etc.

- Significantly improved application ramp-up
  - Up-to 50% less CPU to ramp-up to steady-state
  - Improved perf of ahead-of-time compiled code

- Improved Monitoring
  - JMX beans for precise CPU-time monitoring

- Enhancements to JZOS Toolkit for Java batch

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Liberty z/OS – SSL-Enabled DayTrader3.0 (8.5.5.5)

Liberty plans to support Java 8 in 1Q15 – with 8.5.5.5

Secure Application Server with SSL (clear key)
1 CP and 4 zIIPs

2.62x improvement in throughput with IBM Java 8 and IBM z13

(Controlled measurement environment, results may vary)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Distributed ID to SAF Identity Mapping (8.5.5.5)

- Single or multiple distributed identities can be automatically mapped to SAF credential (UserID).
- Delivered by APAR PI26630
- Enable in server.xml:
  `<safCredentials mapDistributedIdentities="true"/>
- The following example illustrates the syntax for creating a distributed identity filter that uses the RACMAP command:
  - RACMAP ID(<SAFUser>) MAP USERDIDFILTER(NAME(<distributedUserId>))
  - REGISTRY(NAME(<distributedRealmName>)) WITHLABEL(<someLabel>)

In the example:
The `<SAFUser>` element is the SAF user in z/OS security.
The `<distributedUserId>` element is the distributed identity.
The `<distributedRealmName>` element is the realm name of the distributed identity.
The `<someLabel>` element is a field that describes this distributed identity filter.
You’ll see how this is implemented in an upcoming section of this presentation.
Java Batch (Beta) – Liberty support

Liberty Profile 8.5.5.6 and above
IBM’s fast, lightweight, composable server runtime
Dynamic configuration and application updates
JVM Stays Active Between Jobs
Avoids the overhead of JVM initialize and tear down for each job
IBM Extensions to JSR 352
JSR 352 is largely a *programming* standard
IBM extensions augment this with valuable *operational* functions
Includes:
Job logs separated by job execution
REST interface to JobOperator
Command line client for job submission
Integration with enterprise scheduler functions
Multi-JVM support: dispatcher and endpoint servers provide a distributed topology for batch job execution

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
The JSR 352 standard calls for a JobRepository to hold job state information, but it does not spell out implementation details.

IBM JSR 352 provides three options for this:

An in-memory JobRepository
For development and test environments where job state does not need to persist between server starts.

File-based Derby JobRepository
For runtime environments were a degree of persistence is desired, but a full database product is not needed.

Relational database product JobRepository
For production and near-production environments where a robust database product is called for.

Table creation is automatic. Relatively easy to drop one set of tables and re-configure to use a different data store.
The `batchManager` command line interface client provides:

A way to submit, monitor and control jobs remotely using a command line interface.

- On the same system, or a different system ... different OS ... doesn't matter: TCP/IP and REST/JSON.
- Uses the REST interface on the IBM Java Batch server.
- Which means the same security model is in effect: SSL, authentication, role-based access.
- External schedulers can use this to submit and monitor job completion.
- `batchManager` parameters allow the script to “wait” for Java to complete. Parameters allow for discovery of job log information, and a mechanism to retrieve the job log for archival if desired.
Multi-JVM Support: Get Jobs Based on Endpoint Criteria

A property in the server.xml defines the “message selector” criteria to use to pick up messages. You can designate – by server – what criteria to use.

Submit jobs and have them run only when intended server starts and picks up the submission request.

Have jobs run in intended servers based on selection criteria of your choice.

Not limited to system, not limited to platform … may span systems and platforms.

---

Liberty Profile

Queue

Dispatcher

SIBus or MQ

Endpoint

server.xml

---

1. ... messageSelector="com.ibm.ws.jbatch_applicationName = 'BatchJobA'"

2. ... messageSelector="com.ibm.ws.jbatch_applicationName = 'BatchJobA' OR com.ibm.ws.jbatch_applicationName = 'BatchJobB'"

3. ... messageSelector="com.ibm.ws.jbatch_applicationName = 'BatchJobA' AND com.ibm.ws.jbatch_myProperty = 'myValue'"

---

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
WebSphere Classic – What's New?
Dispatch Progress Monitor (DPM)

Look at the response times of a lot of requests over time....

Most will be fairly short

And collect the configured doc (javacore, etc) at that interval

But occasionally one takes much longer than normal

Set the DPM Interval to a value greater than the normal response time

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Dispatch Progress Monitor (DPM) Settings

dispatch_timeout="_____"
queue_timeout_percent ="_____"
request_timeout="_____"
stalled_thread_dump_action="_____"
cputimeused_limit="_____"
cputimeused_dump_action="_____"
dpm_interval="_____"
dpm_dump_action="_____"
SMF_request_activity_enabled="__"
SMF_request_activity_timestamps="__"
SMF_request_activity_security="__"
SMF_request_activity_CPU_detail="__"
classification_only_trace="__"
message_tag="_____"
timeout_recovery="_____">

DPM stands for Dispatch Progress Monitor. It is a function that will process a dump action every $n$ seconds.
dpm_interval is the interval period expressed in seconds
dpm_dump_action is the same as we just saw for the other dump action: svcdump, javacore, heapdump, traceback, javatdump and none

This function has a set of MODIFY commands that may be used to clear DPM settings or reset to XML settings
See WP102023 for the details on these MODIFY actions for DPM
Message routing

- Messages are written as:
  - WTOs to the console
  - WTOs to the log
  - Writes to SYSOUT or Logstream
- The destination for a message is determined by the code that issues it

- New environment variables override the code
  - Can move the message to a new place entirely, or duplicate it elsewhere
- Force messages (by ID) to a chosen target
- Or 'NONE' to suppress entirely
- Update dynamically with MODIFY
- Use DISPLAY to see current configuration
PM74923 – Server Output Management

Provides a means of routing SYSPRINT and SYSOUT to a UNIX file rather than JES with better file management than simple output routing we had before.

In development and test environments Java developers wish to view server output from UNIX file, not JES. Ability to push server output to UNIX file has existed for some time, but output file management was very limited.

Introduced In

- V7.0.0.29
- V8.0.0.6
- V8.5.0.2

With PM74923 comes two new variables:

- `DAEMON_redirect_server_output_dir` (for the Daemon)
- `redirect_server_output_dir` (for everything else)

The output file names will be uniquely named by WAS F <server>, ROLL_LOGS will close existing and start new

Use whatever UNIX file viewing mechanism you wish

More detail, including how the IHS Apache webserver can be used to list output files, control access to those files, and browse the files

ibm.com/support/techdocs/atasmastr.nsf/WebIndex/WP102267

More detail, including how the IHS Apache webserver can be used to list output files, control access to those files, and browse the files
Common Problems

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
collector.sh on WebSphere App Server z/OS

- collector.sh translates its output ASCII > EBCDIC, which causes problems on z/OS

- Issue these 3 USS commands from a directory outside of your WebSphere config directory (ie. /tmp)

  export IBM_JAVA_OPTIONS=-Dfile.encoding=ISO8859-1
  WAS_HOME/DeploymentManager/profiles/default/bin/collector.sh
  WAS_HOME/AppServer/profiles/default/bin/collector.sh

- May see ‘deprecated’ message… don’t be alarmed…
‘WebSphere Application Server Configuration Visualizer [Report]’
Collecting HFS Configuration Files

Save time and help expedite problem resolution by sending in all configuration files in once.

**WebSphere Classic**
Run the collector shell script to obtain files with extensions xml, xmi, props, properties, ffdc logs, and others.

```
export IBM_JAVA_OPTIONS=-Dfile.encoding=ISO8859-1
/WAS_HOME/AppServer/profiles/default/bin/collector.sh
```

Ignore messages about deprecation.
This will create a jar labeled hostname.cellname.nodename.default-WASenv.jar in the directory that you ran the command from.

**WebSphere Liberty**
Run the server dump command to create a zip file containing server.xml, server.env, ffdc logs.
Ensure you have your JAVA_HOME set to the JVM location and WLP_USER_DIR set to the Liberty Profile directory.

```
export JAVA_HOME=/usr/lpp/zWebSphere/Liberty/V8R5/java/java_1.7_64
export WLP_USER_DIR=/WebSphere/Liberty
/usr/lpp/zWebSphere/Liberty/V8R5/bin:>server dump <servername>
```

This will create a zip file called servername.dump-year.month.day_hour.minute.second.zip in the WLP_USER_DIR/severs/servername directory.
Enabling Trace for WebSphere Classic

WebSphere Classic

Using the administrative console

Logging and tracing > servername > Change log detail levels

Trace output will typically go to SYSPRINT in the job output or if High Performance Extensible Logging (HPEL) is enabled, trace output will go to the file system in WAS_HOME/profiles/default/logs/servername/tracedata directory.

Enabling Trace for WebSphere Classic
Enabling Trace for WebSphere Liberty

Add the logging XML tag with your traceSpecification to server.xml

Edit ASCII file server.xml

/WebSphere/Liberty/servers/server1/server.xml

```xml
<server description="new server">
    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
    </featureManager>
    <httpEndpoint id="defaultHttpEndpoint" host="localhost" httpPort="9080" httpsPort="9443" />
    <logging traceSpecification="*=info:com.ibm.ws.*=all"/>
</server>
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Dynamic Tracing for WebSphere Classic and Liberty

**WebSphere Classic**

Enabling a trace using the modify command

F BBOS1,TRACEJAVA='com.ibm.ws.*=all'
BB000211I MODIFY COMMAND TRACEJAVA='com.ibm.ws.*=all' COMPLETED SUCCESSFULLY

Resetting a trace using the modify command

F BBOS1,TRACEINIT
BB000211I MODIFY COMMAND TRACEINIT COMPLETED SUCCESSFULLY

**WebSphere Liberty** *(requires servers to be started using angel BBZANGL and server BBGZSRV procs)*

Enabling a trace using the modify command

F BBGZSRV,LOGGING='com.ibm.ws.*=all'
+CWWKB0005I: COMMAND RESPONSES COMPLETED SUCCESSFULLY FROM Logging
+CWWKB0002I: MODIFY COMMAND LOGGING='com.ibm.ws.*=all' COMPLETED SUCCESSFULLY.

Resetting a trace using the modify command

F BBGZSRV,LOGGING=RESET
+CWWKB0005I: COMMAND RESPONSES COMPLETED SUCCESSFULLY FROM Logging
+CWWKB0002I: MODIFY COMMAND LOGGING=RESET COMPLETED SUCCESSFULLY.
Obtaining a Javacore

WebSphere Classic

wsadmin.sh/Jython
wsadmin.sh [-host host_name] [-port port_number] [-user userid][-password password]
jvm = AdminControl.completeObjectName('type=JVM,process=SERVER_NAME,*')
AdminControl.invoke(jvm, 'dumpThreads')

F BBOS1, JAVACORE
BBOO0211I MODIFY COMMAND JAVACORE COMPLETED SUCCESSFULLY
JVMDUMP034I User requested Java dump using
'/SY1/var/WebSphere/home/WSCFG1/javacore.20150227.190936.67008.0001.txt' through JVMRI

BBOO0211I MODIFY COMMAND JAVACORE COMPLETED SUCCESSFULLY
JVMDUMP034I User requested Java dump using
'/SY1/var/WebSphere/home/WSSR1/javacore.20150227.190937.67012.0001.txt' through JVMRI

WebSphere Liberty

server javadump servername
/usr/lpp/zWebSphere/Liberty/V8R5/bin:>server javadump server1
Dumping server server1.
Server server1 dump complete in
/WebSphere/Liberty/servers/server1/javacore.20150224.102206.83952322.0001.txt.

F BBGZSRV, JAVACORE
JVMDUMP034I User requested Java dump using
'/WebSphere/Liberty/servers/defaultServer/javacore.20150224.162944.66238.0002.txt'
through com.ibm.jvm.Dump.JavaDump
+CWWKB0005I: COMMAND RESPONSES COMPLETED SUCCESSFULLY FROM Javacore Command Handler.
+CWWKB0002I: MODIFY COMMAND JAVACORE COMPLETED SUCCESSFULLY.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Obtaining a Portable Heapdump PHD

**WebSphere Classic**

```
wsadmin.sh/Jython
wsadmin.sh [-host host_name] [-port port_number] [-user userid][-password password]
jvm = AdminControl.completeObjectName('type=JVM,process=SERVER_NAME,*)
AdminControl.invoke(jvm, 'generateHeapDump')
```

F BBOS1, HEAPDUMP
JVMDUMP034I User requested Heap dump using '/SY1/var/WebSphere/home/WS SR1/heapdump.20150227.211036.67012.0002.phd' through JVMRI

**WebSphere Liberty**

```
server javadump servername --include=heap
/usr/lpp/zWebSphere/Liberty/V8R5/bin:>server javadump server1 --include=heap
Dumping server server1.
Server server1 dump complete in
/WebSphere/Liberty/servers/server1/javacore.20150224.102223.83952322.0002.txt.
Server server1 dump complete in
/WebSphere/Liberty/servers/server1/heapdump.20150224.102223.83952322.0003.phd.
```

F BBGZSRV, HEAPDUMP
JVMDUMP034I User requested Heap dump using
'/WebSphere/Liberty/servers/defaultServer/heapdump.20150224.163149.66238.0003.phd'
through com.ibm.jvm.Dump.HeapDump
+CWWKB0005I: COMMAND RESPONSES COMPLETED SUCCESSFULLY FROM Heapdump Command Handler.
+CWWKB0002I: MODIFY COMMAND HEAPDUMP COMPLETED SUCCESSFULLY.
Security Vulnerabilities

Yes.....vulnerabilities get reported

   News media

   Network of friends / co-workers

   CVE – Common Vulnerabilities and Exposures

No... WebSphere App Server z/OS Level 2 cannot discuss them

So where do you get the information?

System z Security Portal

contains all pertinent information regarding security vulnerabilities for products on the z/OS platform, and how to resolve them
IBM recommends that you promptly install security and integrity PTFs
SECINT PTFs are included in RSUs periodically
The System z Security Portal can help you stay more current with SECINT
PTFs by providing SMP/E HOLDDATA you can use to identify these fixes before
they are marked RSU
The System z Security Portal also provides associated Common Vulnerability
Scoring System (CVSS) V2 ratings for new APARs*
To get this information, you must register!
Because widespread specifics about a vulnerability could increase the likelihood that
an attacker could successfully exploit it
In response to customer requests to maintain the confidentiality
Other requirements on the website
IBM recommends that you visit the System z Security Portal site at http://www-
03.ibm.com/systems/z/solutions/security_subintegrity.html to get the
information you need to register
Questions can be directed to: syszsec@us.ibm.com

* Note: According to the Forum of Incident Response and Security Teams (FIRST), the Common Vulnerability Scoring System (CVSS) is an “industry open standard designed to convey
vulnerability severity and help to determine urgency and priority of response.” IBM PROVIDES THE CVSS SCORES “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. CUSTOMERS ARE RESPONSIBLE FOR ASSESSING THE IMPACT OF ANY ACTUAL OR
POTENTIAL SECURITY VULNERABILITY IN THEIR SPECIFIC ENVIRONMENT. IBM DOES NOT PROVIDE A CVSS ENVIRONMENT SCORE. THE CVSS ENVIRONMENT SCORE IS
CUSTOMER ENVIRONMENT SPECIFIC AND WILL IMPACT THE OVERALL CVSS SCORE. CUSTOMERS SHOULD EVALUATE THE IMPACT OF ANY ACTUAL OR POTENTIAL
SECURITY VULNERABILITY IN THEIR SPECIFIC ENVIRONMENT.

* IBM DOES NOT PROVIDE A CVSS ENVIRONMENT SCORE. THE CVSS ENVIRONMENT SCORE IS CUSTOMER ENVIRONMENT SPECIFIC AND WILL IMPACT THE
OVERALL CVSS SCORE. CUSTOMERS SHOULD EVALUATE THE IMPACT OF ANY ACTUAL OR POTENTIAL SECURITY VULNERABILITY AND CAN CALCULATE A
CVSS ENVIRONMENT SCORE.
WebSphere Classic Migration

- Migrating large topology with many Applications
- You can Migrate the App Servers alone
- Generate job to migrate applications later
  - Break up the generated job when large number of Apps

- Knowledge Collection: Migration planning for WebSphere Application Server
  

- IBM WebSphere Application Server Migration Toolkit
  

Complete your session evaluations online at [www.SHARE.org/Orlando-Eval](http://www.SHARE.org/Orlando-Eval)
IBM HTTP Server on z/OS

- Domino Powered ➔ Apache Powered
- DGW no longer shipped in z/OS 2.2
- Documentation available
  - DGW Migration Frequently Asked Questions FAQ
    http://publib.boulder.ibm.com/httpserv/ihsdiag/dgw_migration_faq.html
  - Brief description of config directive differences
  - Redpaper – “IBM HTTP Server on z/OS”
  - 3.1.2 of Redpaper how you can download for FREE!!
/admin URL does not forward to /ibm/console

• Fixpack 8.5.5.6 and above

• web.xml for adminredirector.war was updated to work correctly when the disallowServeServletByClassname is defaulted to true

• New web.xml is used when new profiles are created after installing the fixpack(s)

• Profiles that existed prior need to be updated by running the script attached to Techdoc

/admin URL does not forward to /ibm/console

- Other options:
  - Reinstall the console using the deployConsole.py script
  - Just do a remove and install


- Just use /ibm/console URI instead of /admin URI
Java in WebSphere Classic

JDK 6.1
Installed with WebSphere Classic 8.5.0.0
Updated with WebSphere FixPack Maintenance

JDK 7.0
Delivered with WebSphere 8.5.0.0
Optionally installed
MUST install JDK 7.0 BASE Level FIRST, before installing JDK 7.0 FixPacks

JDK 7.1
Available at WebSphere 8.5.5.2
Base level JDK 7.1 (Must install base before Fix Packs JDK 7.1)
Java in WebSphere Liberty

No JDK installed

Liberty supports running with JAVA 8

PI45300: Ship Java 8 SR1 FP10 for WSAS Liberty profile V855X

Where to get it:
Java in WebSphere Blog Updates

Recent BLOG Entry:

WebSphere
    Application Server for z/OS: Your first cup of Java is included!

Where to get them

How to Install

How to switch to use them

This BLOG for Classic WebSphere, Liberty Blog coming soon
WAS Timeout – ABENDEC3 RC=04130004

Servant Region internal Worker thread default = 3

Cross memory deadlock within the Servant Region

SR threads make mbean request back to CR

   Seen admin related requests / use SR internal worker thread

These can then make additional mbean requests

   Using SR internal worker thread

With only 3, it can tie them up quickly

XD, Portal Server, BPM seem to use this function
Server custom property

private_bboo_internal_work_thread_pool_size

Admin console -> Environment -> WebSphere Variables -> <Select server> <scope>

Variable name: private_bboo_internal_work_thread_pool_size

Value: greater than 3 (the default)
   10 seems to be a good number
Questions?

Complete your session evaluations online at www.SHARE.org/Orlando-Eval