

CICS and Liberty



Web Apps using WebSphere Application Server Liberty Profile Technology in CICS

Catherine Moxey
IBM

Thursday, February 7, 2013
Session Number 12445



Disclaimers

© IBM Corporation 2013. All Rights Reserved.



IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. Any information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

The session and materials has been prepared by IBM or the session speaker and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Complete your sessions evaluation online at SHARE.org/SFEval





Trademarks

IBM, the IBM logo, ibm.com, AppScan, CICS, Cloudburst, Cognos, CPLEX, DataPower, DB2, FileNet, ILOG, IMS, InfoSphere, Lotus, Lotus Notes, Maximo, Quickr, Rational, Rational Team Concert, Sametime, Tivoli, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. 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 IBM trademarks is available on the Web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml.

Coremetrics is a trademark or registered trademark of Coremetrics, Inc., an IBM Company.

SPSS is a trademark or registered trademark of SPSS, Inc. (or its affiliates), an IBM Company.

Unica is a trademark or registered trademark of Unica Corporation, an IBM Company.

Java and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates. Other company, product and service names may be trademarks or service marks of others. References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.



Session Abstract

- CICS TS V5.1 ~~open beta~~ offers a fast and lightweight Java™ web container, providing developers with the rich features of Java Servlet and JavaServer Pages (JSP) specifications, and fast local access to your existing CICS applications and data. Built on WebSphere® Application Server Liberty profile technology, this web container runs in the CICS JVM server environment. A wide range of Java development tools can be used to develop web applications, such as WebSphere Application Server Developer Tools for Eclipse (WDT), and Rational® Developer for System z. This session will demonstrate these features and show the integration between the web container and CICS resources.



Agenda

- What is the WebSphere Application Server Liberty Profile?
- Liberty Profile on z/OS
- Liberty Web Container in CICS TS
- Step-by-step running a Web app in CICS
- Liberty Features in CICS
- Summary and Q&A

Web container in CICS TS V5.1



Modern interfaces - Build rich web experiences for critical applications

Feature

Embedded web container for Servlets & JSPs

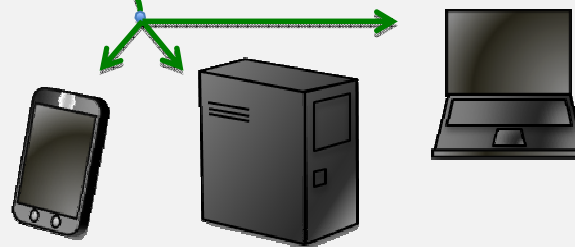
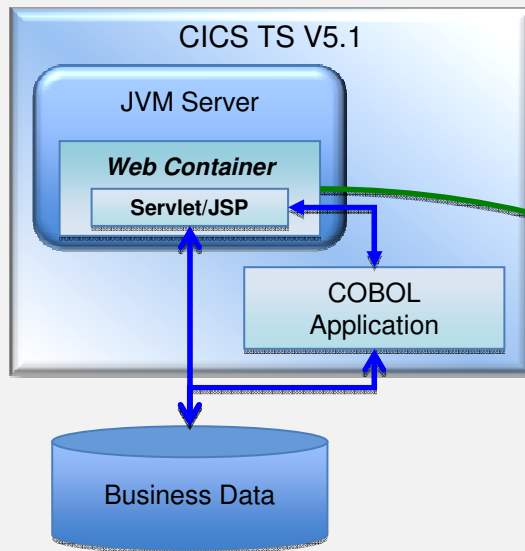
Benefit

Deployment of lightweight Java servlets and Java Server Pages (JSP)

Improved performance through local access to CICS applications and data

Rapid roll-out of interface updates through OSGi-packaged deployments

Full integration with first-class applications and platforms

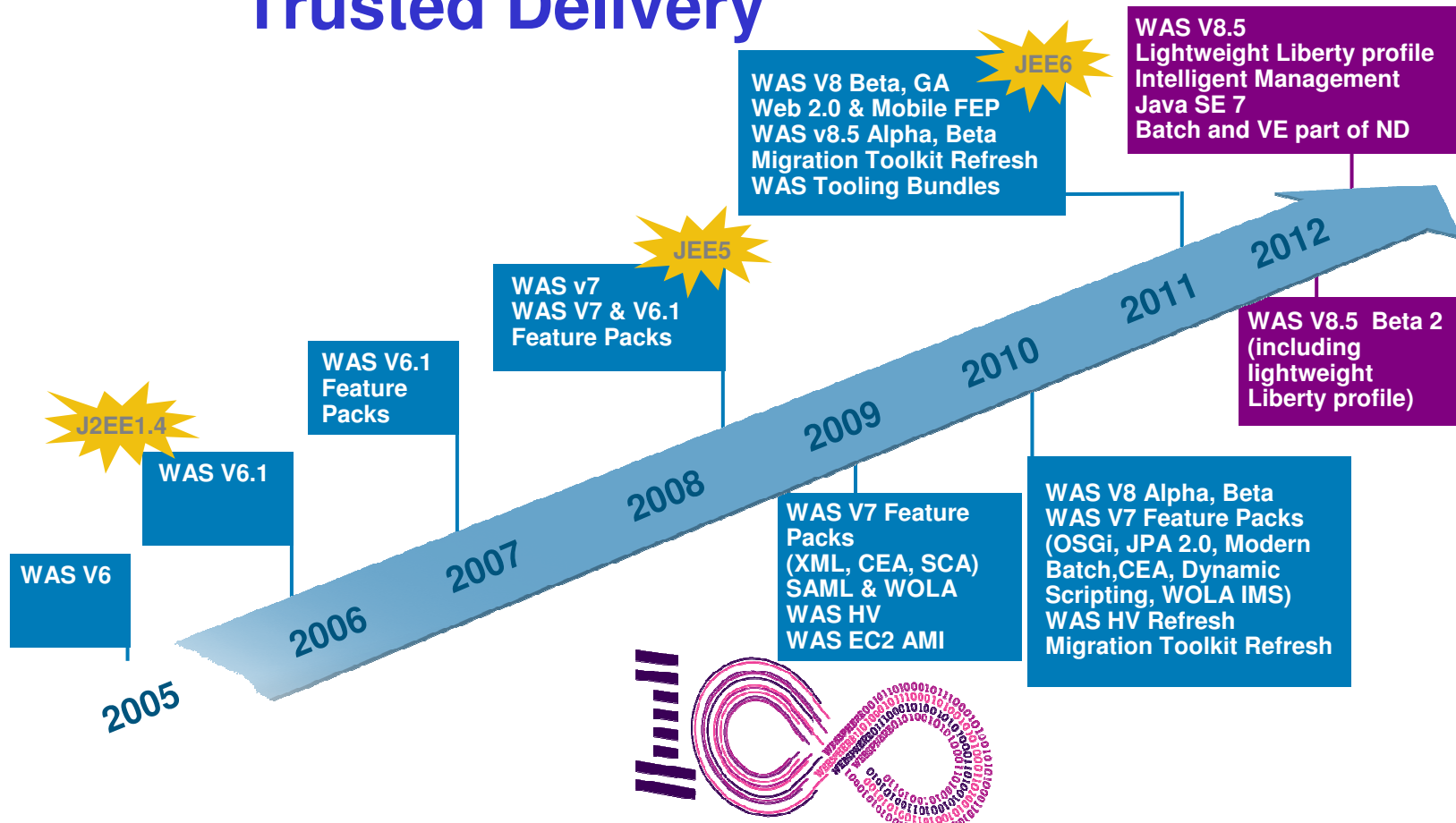




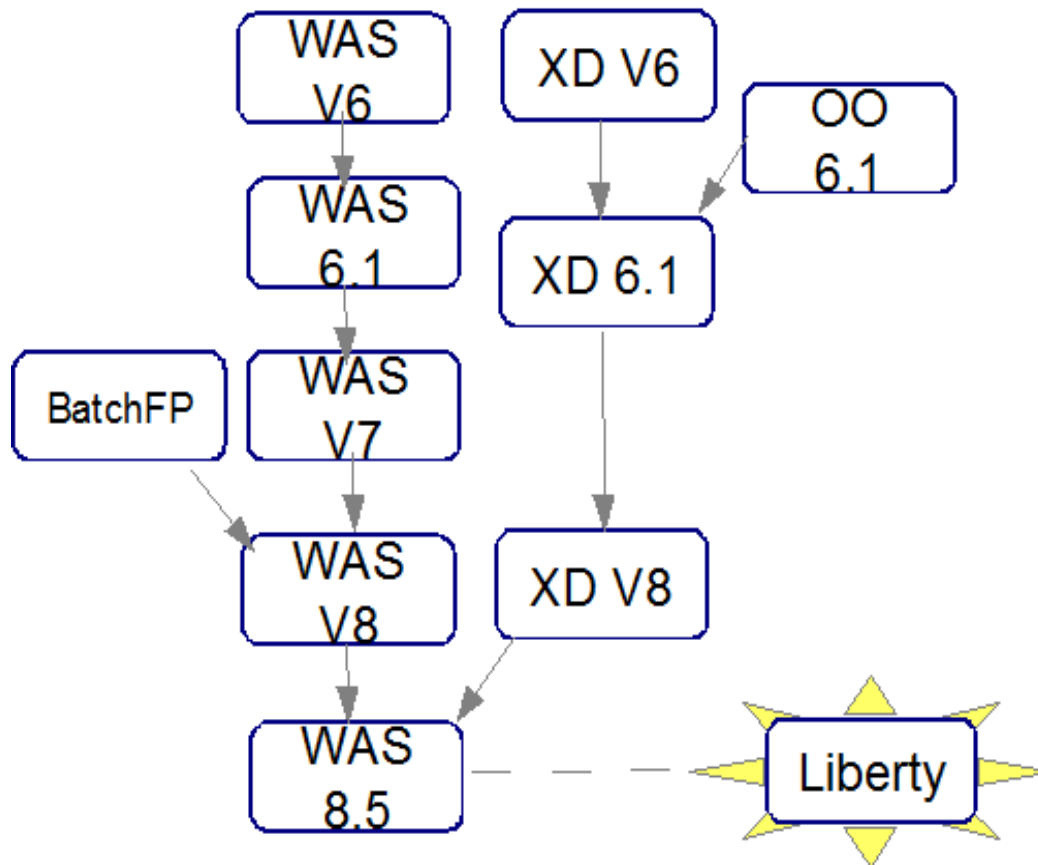
Agenda

- What is the WebSphere Application Server Liberty Profile?
- Liberty Profile on z/OS
- Liberty Web Container in CICS TS
- Step-by-step running a Web app in CICS
- Further details of Liberty in CICS
- Liberty Features in CICS
- Summary and Q&A

WebSphere Application Server: 14 Years of Leadership and Trusted Delivery



WAS 8.5 main components



- XD = Extended Deployment
 - a.k.a. Compute Grid (CG)
- OO = Operations Optimization
 - Virtual Enterprise (part of XD 6.1)
- BatchFP = Modern Batch FP
 - Subset of XD (CG)
 - Included in WAS V8
- WAS 8.5 includes CG, XD, VE, plus:
 - Liberty Profile
 - Java 7 SDK
 - XCT (Cross-Comp. Trace)
 - ISA (IBM Support Asst.)

What is the 'Liberty' Profile?

A lightweight, dynamic, composable runtime

- **Lightweight**
 - Server install is only about 55 MB
 - Extremely fast server starts – typically well under 5 seconds
- **Dynamic**
 - Available features are user selected and can change at runtime
 - Restarts are not required for server configuration changes
- **Composable**
 - Features are implemented as loosely coupled components with lazily resolved optional and mandatory dependencies
 - The availability of features and components determines what Liberty *can do and what's available to applications*

What is the 'Liberty' Profile?

An easy to configure runtime environment

- Simple, extensible, and sparse configuration model
- Configuration can live in a single XML document
- Configuration is by exception
- Defaults are provided by contributing feature
- Only modifications to the defaults are required
- Flexible configuration structure
- 'Includes' mechanism allows for shared configuration elements
- Variable indirection mechanism allows for customization when distributed across multiple JVMs
- Easily managed by version control systems if desired



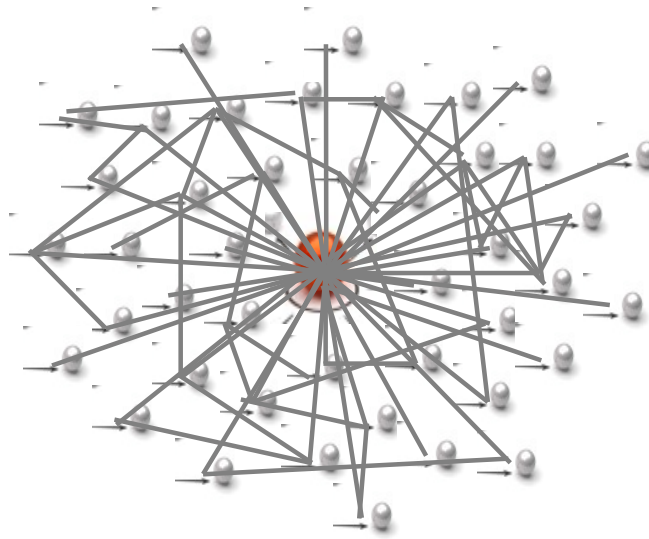
What is the 'Liberty' Profile?

- A transportable runtime for your applications
 - Use “server package” to generate an archive that contains a tested, self-contained, pre-configured server instance that includes your application
- Enables an application-centric deployment model that allows for easy scale-out
- Light-touch admin builds on the ND job manager infrastructure to manage Liberty server instances
- A runtime environment with fidelity to full WAS (“tWAS”)
 - Liberty is WebSphere
 - Applications that are developed and tested on Liberty will run on the full profile

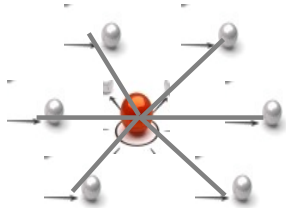


WebSphere®

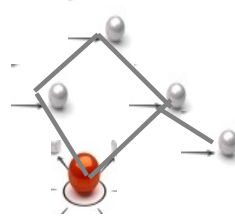
If this is tWAS...



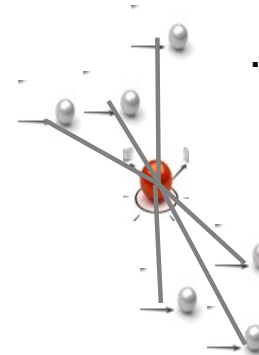
...this is Liberty (WAS)



...so is this



...or even this!



Configuration by Exception

- This is the entire configuration needed to run Liberty as a Web-container with Servlet support.

```
<server description="new server">  
  <featureManager>  
    <feature>servlet-3.0</feature>  
  </featureManager>  
  
  <application id="BasicWeb" location="BasicWeb.war"  
    name="BasicWeb" type="war"/>  
  
</server>
```

Lightweight Configuration

```
<server description="tradeLiteServer">
  <featureManager>
    <feature>jsp-2.2</feature>
    <feature>jdbc-4.0</feature>
  </featureManager>
```

Features control what's available in the runtime.

```
<logging consoleLogLevel="INFO" />
```

Singleton configurations specify properties for runtime services when there's only one instance

```
<application type="war"
  id="tradelite"
  name="tradelite"
  location="${shared.app.dir}/webcontainer/tradelite.war" />
```

Instance configurations allow multiple instances of resources and applications to be declared

```
<include location="jdbc-drivers.xml" />
<include location="${user.home}/custom.xml" optional="true" />
```

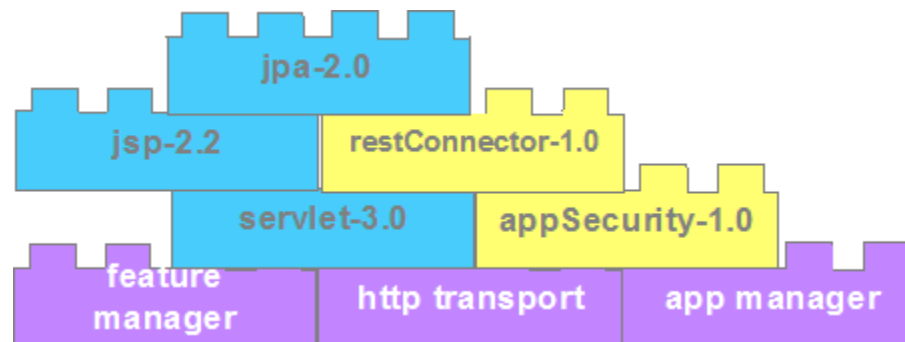
Includes can be used to implement an extensible configuration model

```
<dataSource id="jdbc/DerbyTradeDataSource"
  jndiName="jdbc/TradeDataSource"
  jdbcDriverRef="DerbyEmbedded">
  <properties databaseName="${shared.resource.dir}/data/tradedb" />
</dataSource>
</server>
```

References can be used in multiple elements to point to and share a common definition

Composability – Based on features

```
<server description="composabilityIsKey">  
  <featureManager>  
    <feature>appSecurity-1.0</feature>  
    <feature>jsp-2.2</feature>  
    <feature>restConnector-1.0</feature>  
    <feature>jpa-2.0</feature>  
  </featureManager>  
</server>
```





Liberty and traditional/full profile

There are functional differences between traditional WAS and the Liberty profile – Liberty provides a useful subset of traditional WAS

Liberty Profile

- Bean validation
- Blueprint
- Java API for RESTful Web Services
- Java Database Connectivity (JDBC)
- Java Naming and Directory Interface (JNDI)
- Java Persistence API (JPA)
- Java Server Faces (JSF)
- Java Server Pages (JSP)
- JMX
- Monitoring
- OSGi JPA
- Remote connector
- Secure Sockets Layer (SSL)
- Security
- Servlet
- Session Persistence
- Transaction
- Web application bundle (WAB)
- z/OS Security (SAF)**
- z/OS Transactions (RRS)**
- z/OS Workload Management**

Traditional WAS Profile

Everything Liberty has...



- Enterprise Java Beans (EJBs)
- Messaging (JMS)
- Web Services
- Service Component Arch (SCA)
- Java Connector Architecture (JCA)
- Clustering
- WebSphere Optimized Local Adapters
- Administrative Console
- WSADMIN scripting
- Multi-JVM Server Model**

And much more ...

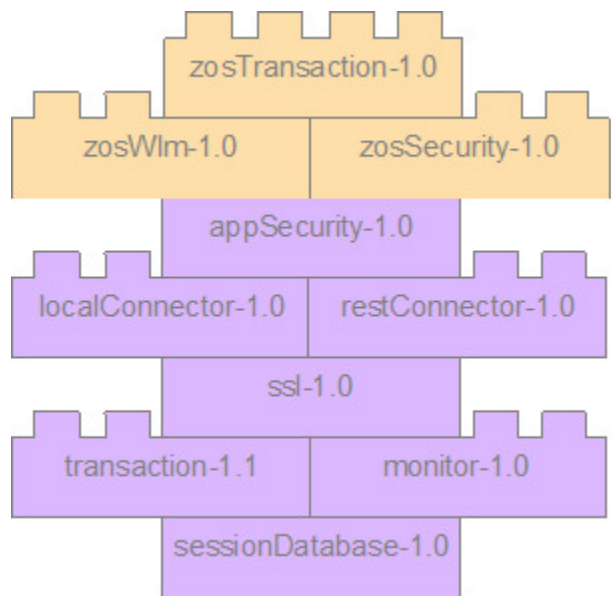


Agenda

- What is the WebSphere Application Server Liberty Profile?
- **Liberty Profile on z/OS**
- Liberty Web Container in CICS TS
- Step-by-step running a Web app in CICS
- Further details of Liberty in CICS
- Liberty Features in CICS
- Summary and Q&A

What is the WAS for z/OS Liberty Profile?

- The WAS for z/OS Liberty profile is Liberty with optional, independently enabled extensions that exploit z/OS facilities
- Only enable exploitation of z/OS features you need
- Only configure the z/OS functions you use
- Focus of V8.5 is basic integration and exploitation



z/OS Feature Sets



Why Liberty on z/OS?

Simplification

- Liberty environments don't need significant z/OS configuration and customization
 - RRS, WLM, and SAF exploitation and configuration is optional
 - No authorized code is **required to host applications**
- Liberty runs in a single process instead of 3+ started tasks
 - Significantly reduced resource consumption
 - No started task definitions are **required**
 - No need to create users and groups for controllers, servants
- Server instances can be quickly created or cloned
 - server create *serverName [options]*
 - server package *serverName [options]*



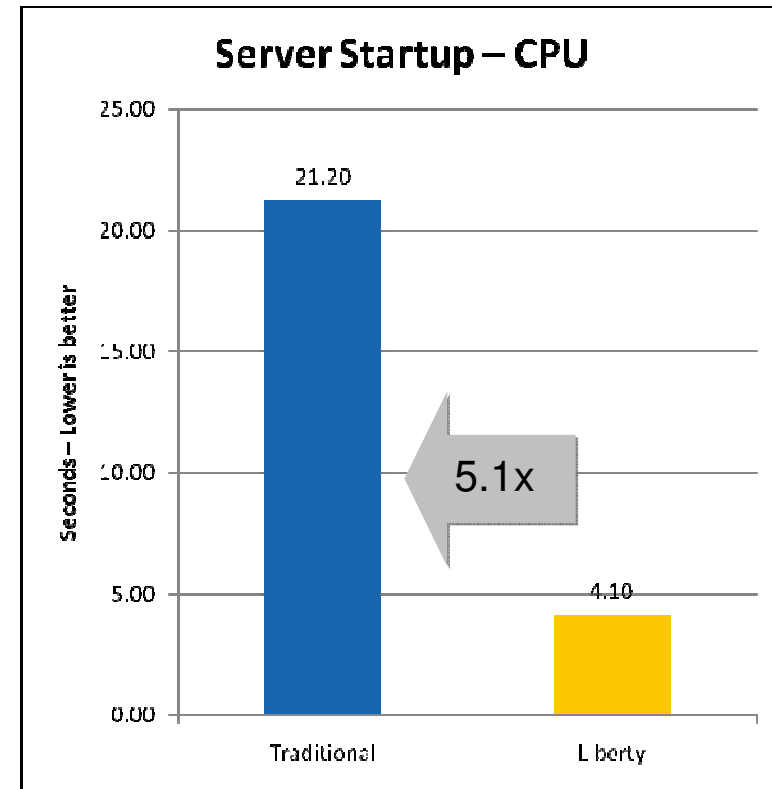
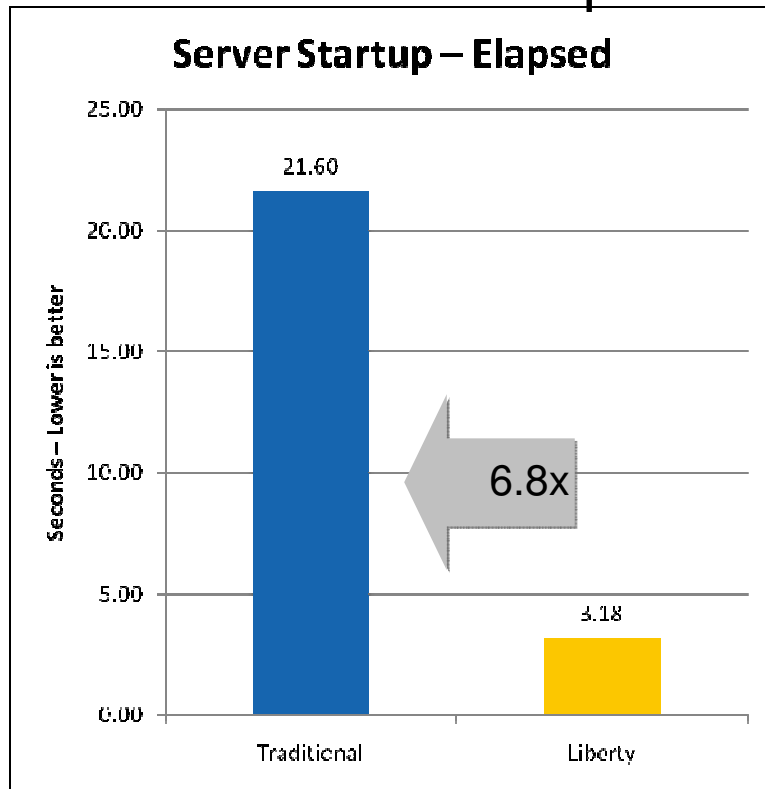
Why Liberty on z/OS?

Application portability and stack consistency

- Liberty behaves *exactly the same on all platforms out of the box*
 - z/OS specific behavior must be configured if desired
- Administration is the same for all platforms out of the box
 - Server operations are controlled by the same server script
 - Logs, trace, and configuration live in the hierarchical file system and are tagged with the appropriate code page for easy viewing and editing
 - Existing server configurations can be brought to z/OS from distributed without modification
- An extremely light-weight, single process runtime
 - Removes deployment and runtime complications introduced by the split process, multi-JVM runtime of traditional WAS for z/OS

Liberty on z/OS – start-up time

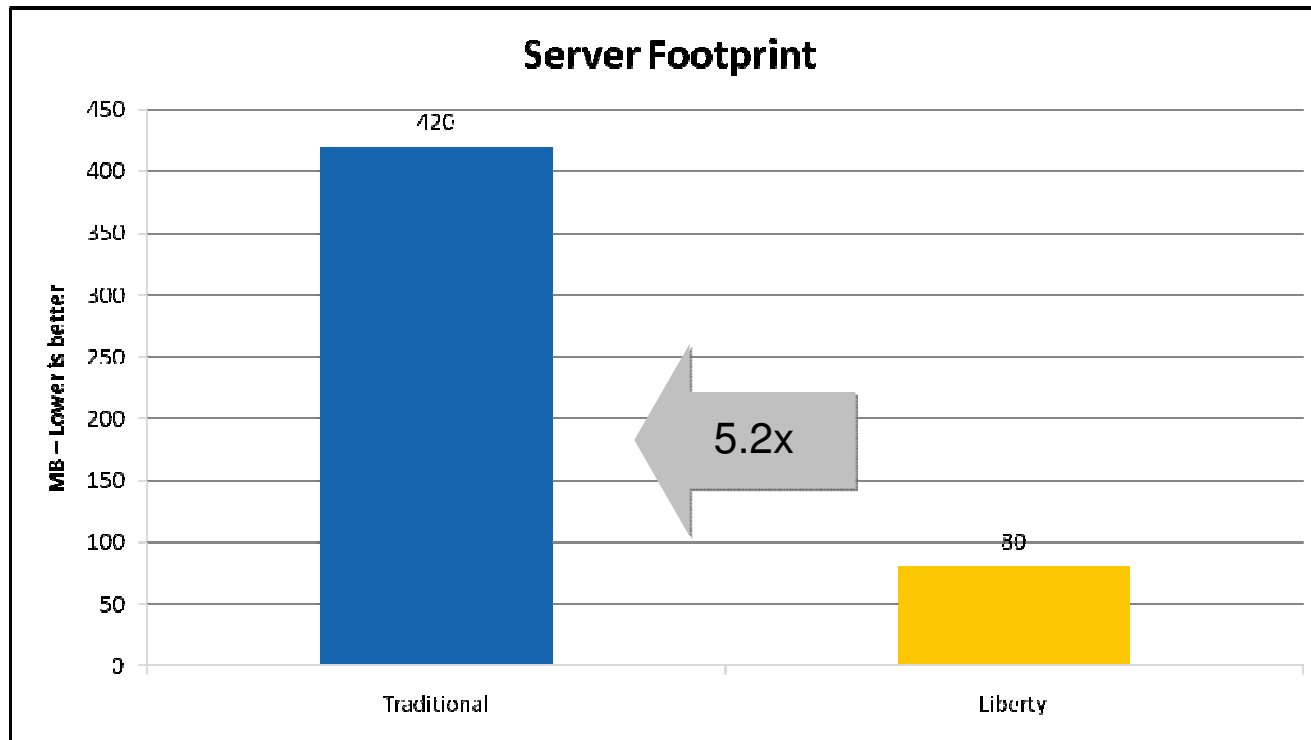
Performance: Start-up time – 3.2 seconds!



- Liberty – 64bit IBM Java 6.0.1, 64/64MB min/max heap, 60MB shared class cache, TradeLite installed
- Traditional – 64bit IBM Java 6.0.1, 1SR, 128/256MB min/max CR heap, 256/512MB min/max SR heap, 75MB CR shared class cache, 75MB SR shared class cache, no applications installed

Liberty on z/OS – memory footprint

Performance: Memory footprint – 80% reduction



- Liberty – 64bit IBM Java 6.0.1, 64/64MB min/max heap, 60MB shared class cache, TradeLite installed
- Traditional – 64bit IBM Java 6.0.1, 1SR, 128/256MB min/max CR heap, 256/512MB min/max SR heap, 75MB CR shared class cache, 75MB SR shared class cache, no applications installed



Agenda

- What is the WebSphere Application Server Liberty Profile?
- Liberty Profile on z/OS
- **Liberty Web Container in CICS TS**
- Step-by-step running a Web app in CICS
- Further details of Liberty in CICS
- Liberty Features in CICS
- Summary and Q&A



Web Container built on WebSphere Application Server Liberty technology

New in CICS TS V5.1 is a **fast and lightweight production-ready Java web container** that provides developers with the rich features of Java **servlet and JavaServer Pages (JSP)** specifications, combined with CICS qualities of service.

In CICS TS V5.1, significant performance improvements can be delivered, when compared with network-connected web interfaces, by **co-locating rich web interfaces with fast local access to your existing CICS applications and data.**

....

This new Java web container is **built on WebSphere Application Server Liberty profile technology**, which enhances compatibility and portability of web applications and tools in either environment. A wide range of Java development tools can be used to build these web applications, including WebSphere Application Server Developer Tools for Eclipse, and Rational Developer for System z.



FAST. LIGHTWEIGHT. LOCAL.



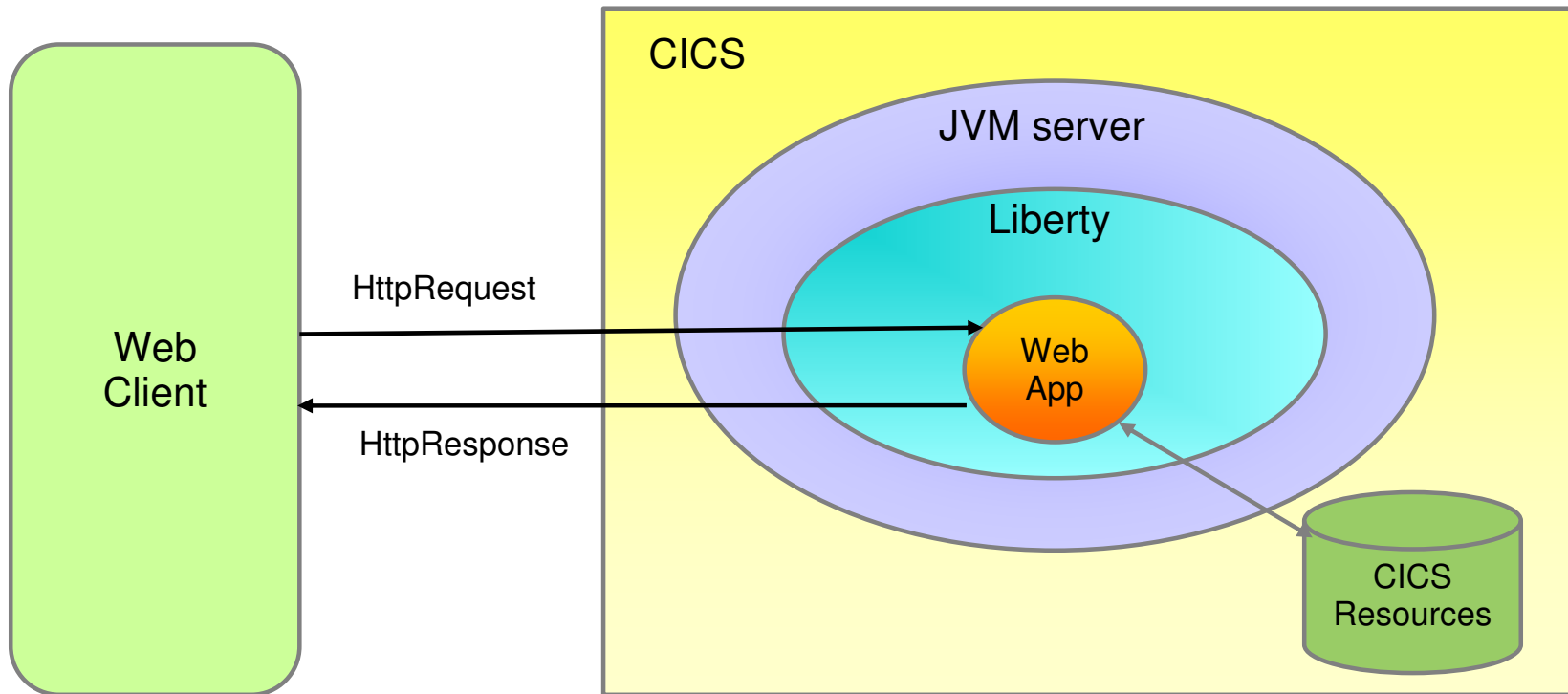
Why and how...

- Provides “off the shelf” Web-server capabilities (JSPs and Servlets)
- JSP and Web servlets have direct, local, access to CICS data and resources
- Servlets can take advantage of existing CICS OSGi applications to provide a Dynamic Web front end

Design Approach

- As little customization as possible.
 - Do things the Liberty way first, and if appropriate, only the Liberty way
 - Ensure Server.xml can be configured dynamically by the user
 - Support Liberty monitored drop-ins directory for applications
- Provide CICS enhancements only where absolutely necessary (security, tasks, JDBC, MQ)
- Provide End-to-end Development and Deployment experience to enable non-mainframe professionals to develop for CICS
- Fully compatible with existing CICS OSGi Java applications running within the same JVM server

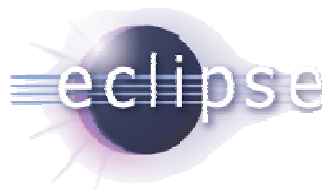
Liberty Technology in CICS JVM Server



Specifications



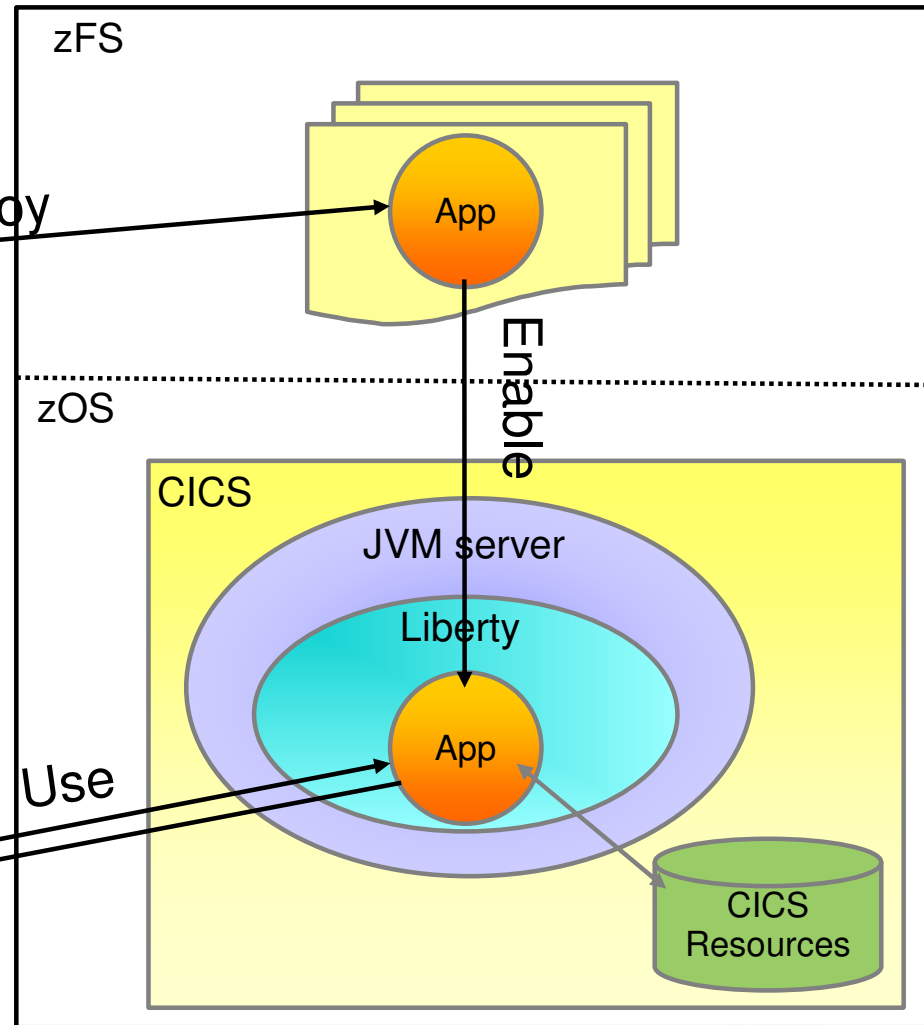
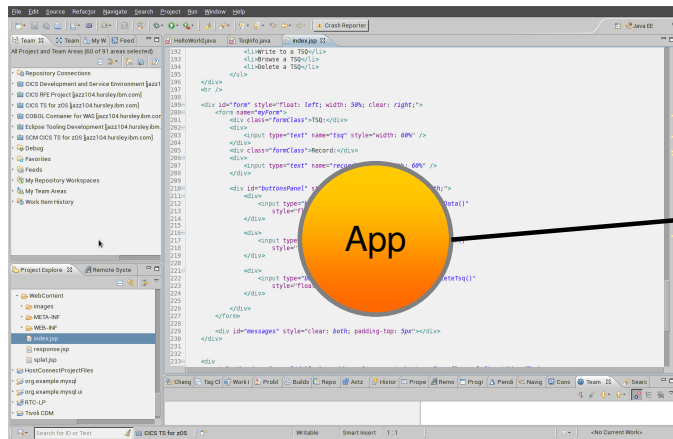
- Java 7 (64-bit)
- Equinox 3.7 as the OSGi framework.
 - Implements the OSGi R4.3 specification
- WAS Liberty Profile 8.5.0
- IBM CICS SDK for WebSphere Application Server Liberty profile v5.1
- Eclipse 3.6.2



Web app Development and Deployment with CICS



Eclipse with Liberty Tools

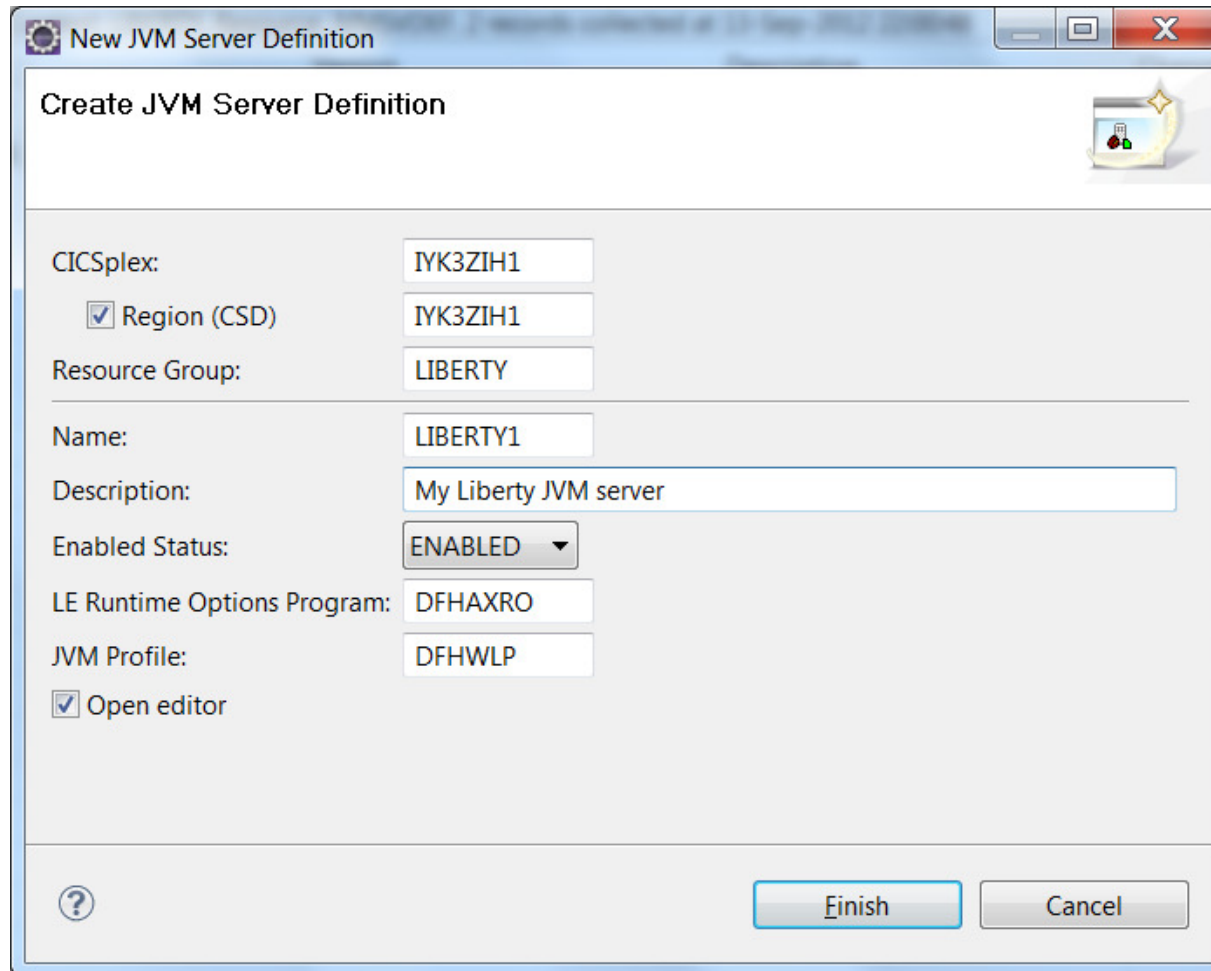




Agenda

- What is the WebSphere Application Server Liberty Profile?
- Liberty Profile on z/OS
- Liberty Web Container in CICS TS
- **Step-by-step running a Web app in CICS**
- Further details of Liberty in CICS
- Liberty Features in CICS
- Summary and Q&A

- 1) Create a JVM server resource in CICS Explorer, CEDA, or CICSplex SM BAS



New JVM Server Definition

Create JVM Server Definition

CICSplex: IYK3ZIH1

Region (CSD) IYK3ZIH1

Resource Group: LIBERTY

Name: LIBERTY1

Description: My Liberty JVM server

Enabled Status: ENABLED ▾

LE Runtime Options Program: DFHAXRO

JVM Profile: DFHWLP

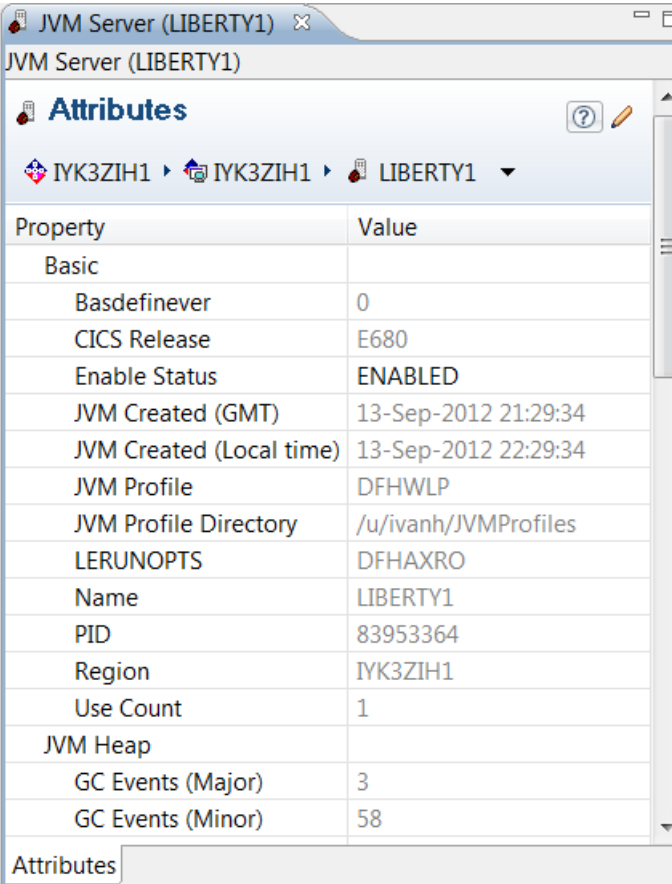
Open editor

? Finish Cancel

2) Configure the JVMProfile

- Copy the sample JVM profile DFHWLP
- Check JAVA_HOME is correct
- Use 'autoconfigure' for easier set-up on a non-Production system
 - *-Dcom.ibm.cics.jvmserver.wlp.autoconfigure=true*
- Configure a unique port number
 - *-Dcom.ibm.cics.jvmserver.wlp.server.https.port=27245*
- Point your JVM server definition at the new JVMProfile

3) Enable the JVM server



The screenshot shows a window titled "JVM Server (LIBERTY1)" with a sub-header "JVM Server (LIBERTY1)". Below the header is the "Attributes" section, which includes a breadcrumb path: "IYK3ZIH1 > IYK3ZIH1 > LIBERTY1". A table lists various properties and their values.

Property	Value
Basic	
Basdefinever	0
CICS Release	E680
Enable Status	ENABLED
JVM Created (GMT)	13-Sep-2012 21:29:34
JVM Created (Local time)	13-Sep-2012 22:29:34
JVM Profile	DFHWLP
JVM Profile Directory	/u/ivanh/JVMProfiles
LERUNOPTS	DFHAXRO
Name	LIBERTY1
PID	83953364
Region	IYK3ZIH1
Use Count	1
JVM Heap	
GC Events (Major)	3
GC Events (Minor)	58



4) Liberty is running! (check the logs).

Server defaultServer created.

Launching defaultServer (wlp-1.0.0.20120428-1251/websphere-kernel_1.0.0) on IBM J9 VM, version pmz6470sr1-20120302_01 (SR1) (en_US)

[AUDIT] CWWKE0001I: The server defaultServer has been launched.

[AUDIT] CWWKG0028A: Processing included configuration resource:
file:/u/ivanh/IYK3ZIH1/LIBERTY1/wlp/usr/servers/defaultServer/installedApps.xml

[AUDIT] CWWKG0028A: Processing included configuration resource:
file:/u/ivanh/IYK3ZIH1/LIBERTY1/wlp/usr/servers/defaultServer/cicsSecurity.xml

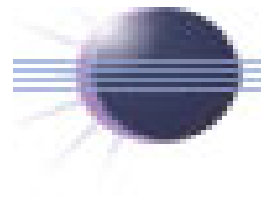
[AUDIT] CWWKZ0058I: Monitoring dropins for applications.

[AUDIT] CWWKF0011I: The server defaultServer is ready to run a smarter planet.

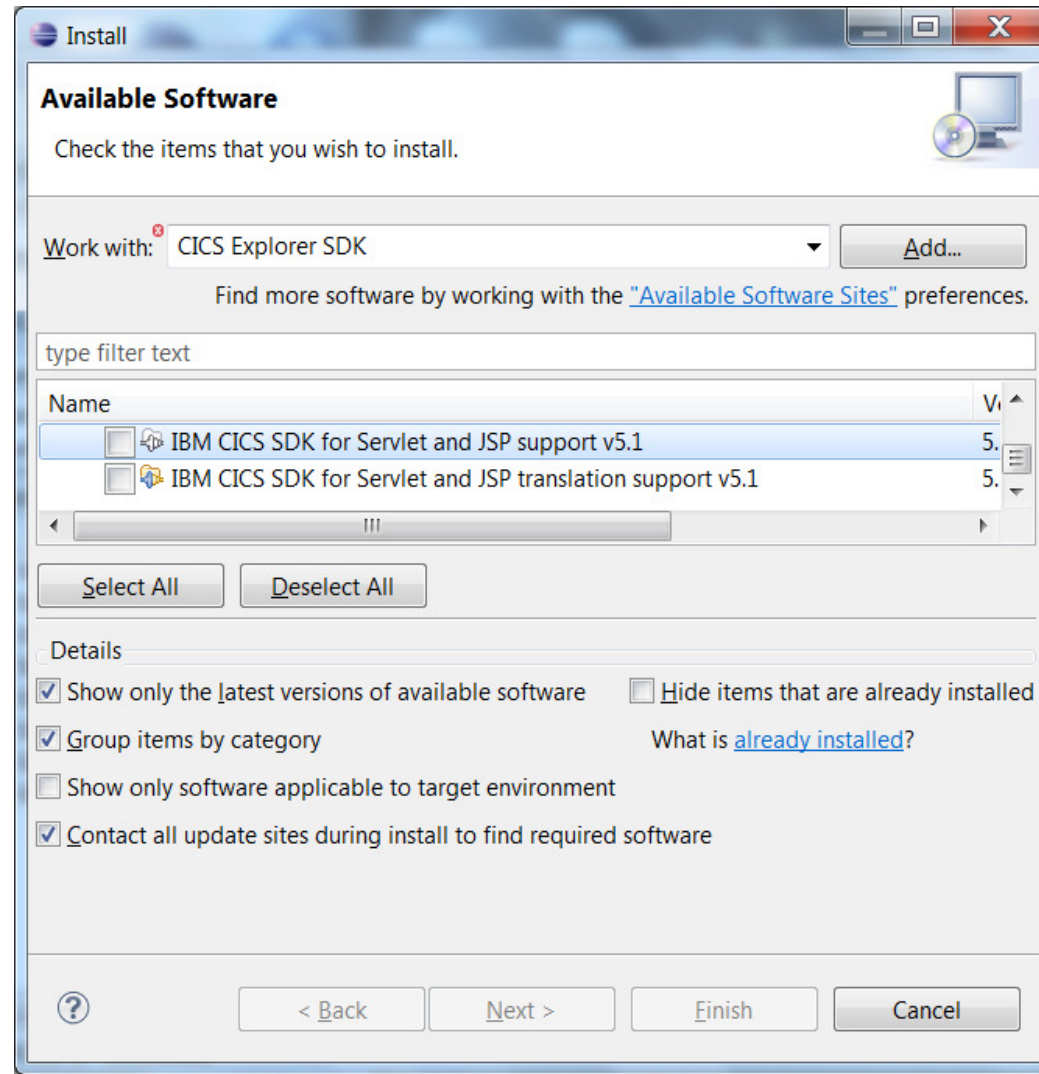
5) Install Eclipse 3.6 / 3.7 - preferably JEE version, but Classic will suffice.

<http://www.eclipse.org/downloads/packages/release/helios/sr2>

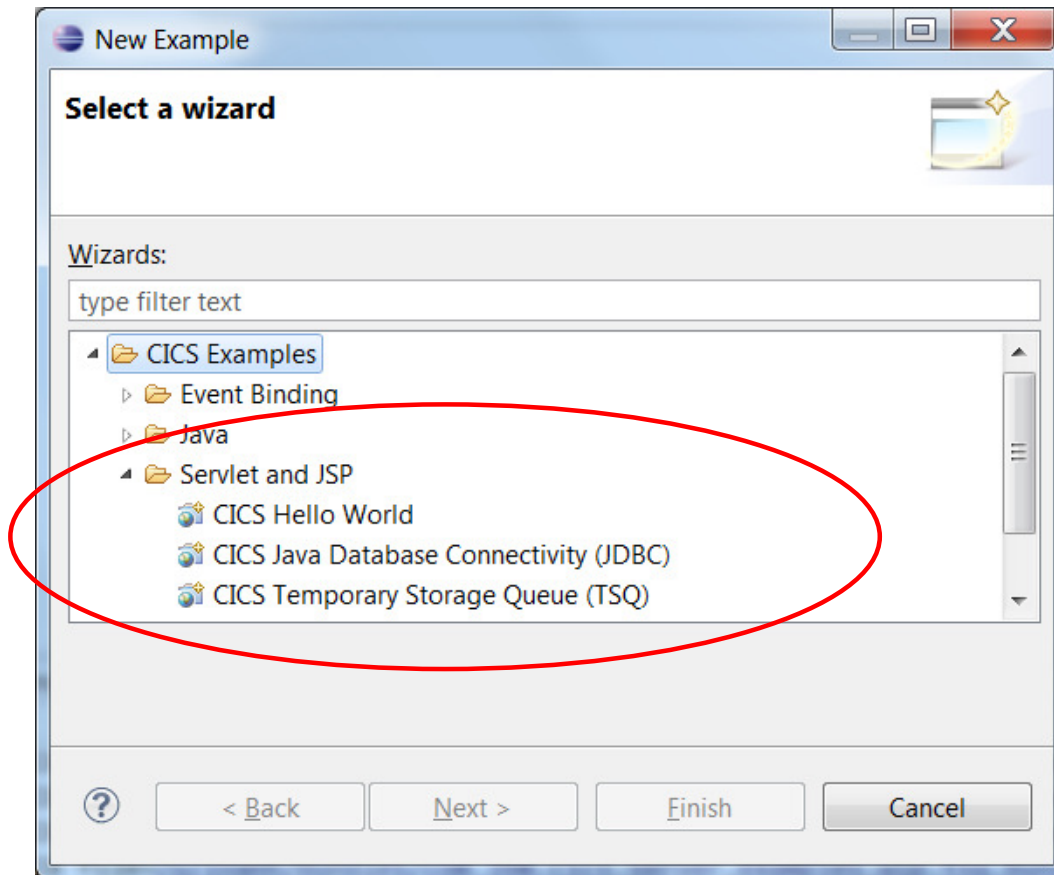
<http://www.eclipse.org/downloads/packages/release/indigo/sr2>



6) Install IBM CICS Explorer SDK for Servlet and JSP support v5.1



- 7) In the CICS Explorer SDK, open the Java perspective.
Create a Dynamic Web Project, or choose one of the Servlet Examples (by opening the 'New Example' wizard):

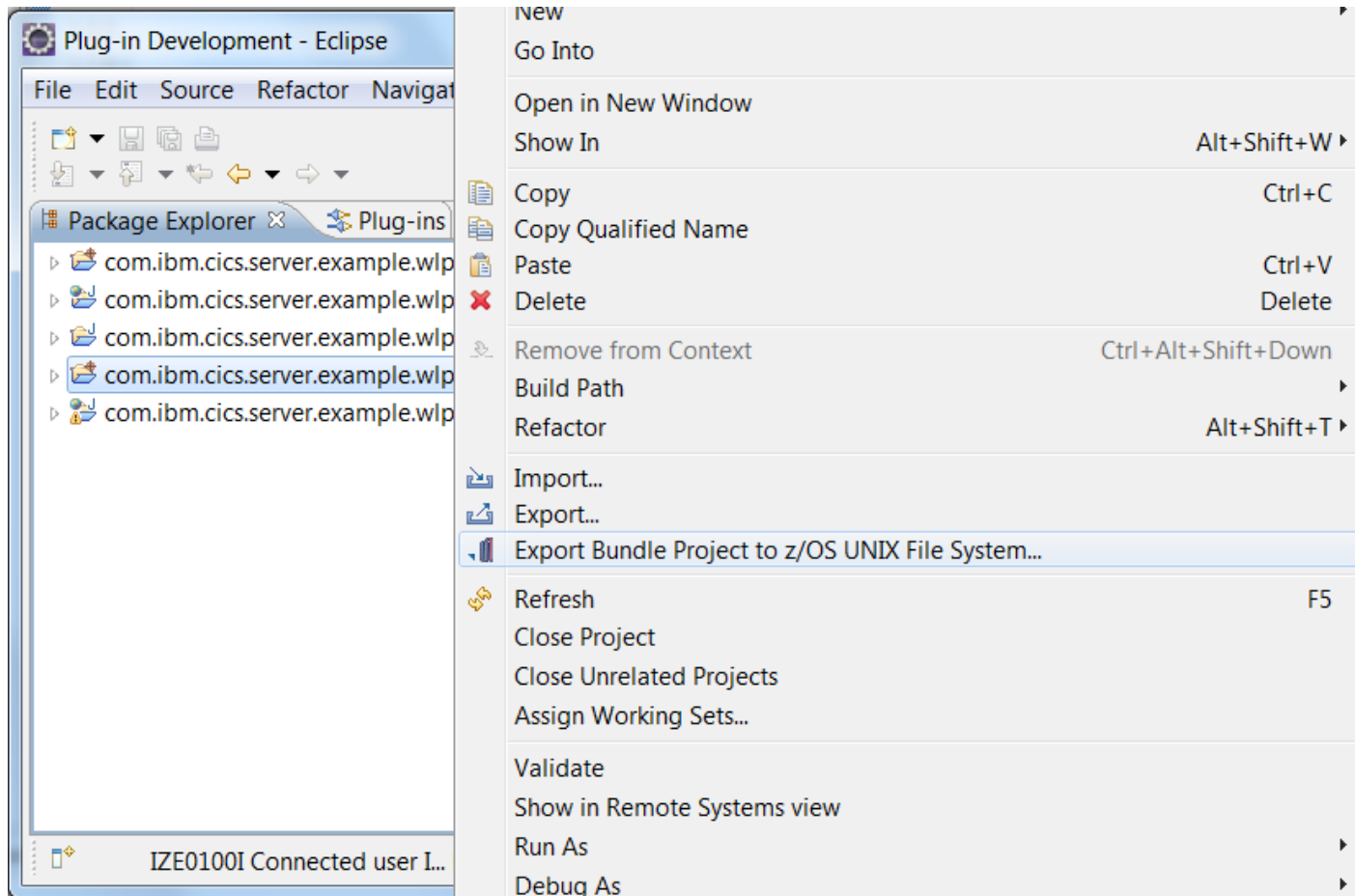


8) Select the CICS Temporary Storage Queue (TSQ) example.

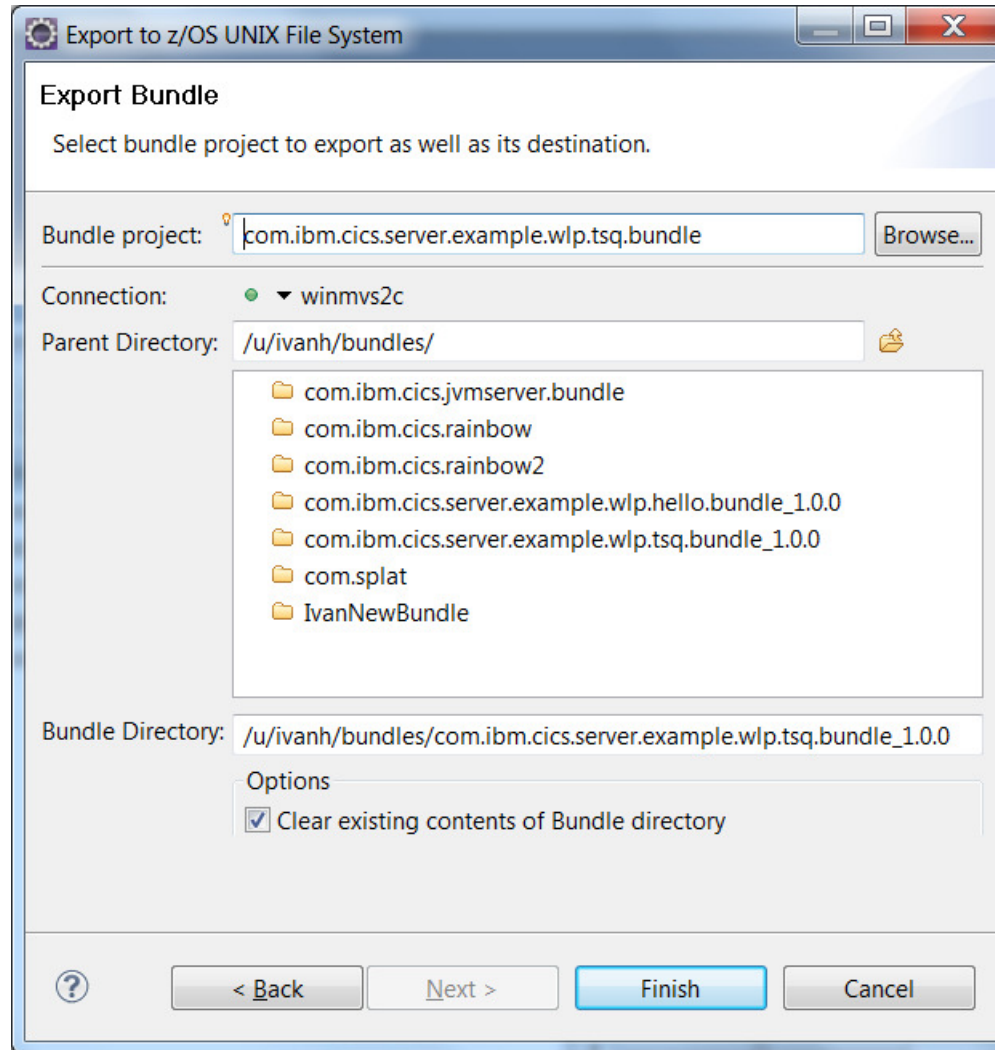
The wizard will create the appropriate projects for the example selected.

For the TSQ example, the wizard creates a plug-in project, a dynamic web project, and a CICS bundle project that references the plug-in project and the dynamic web project.

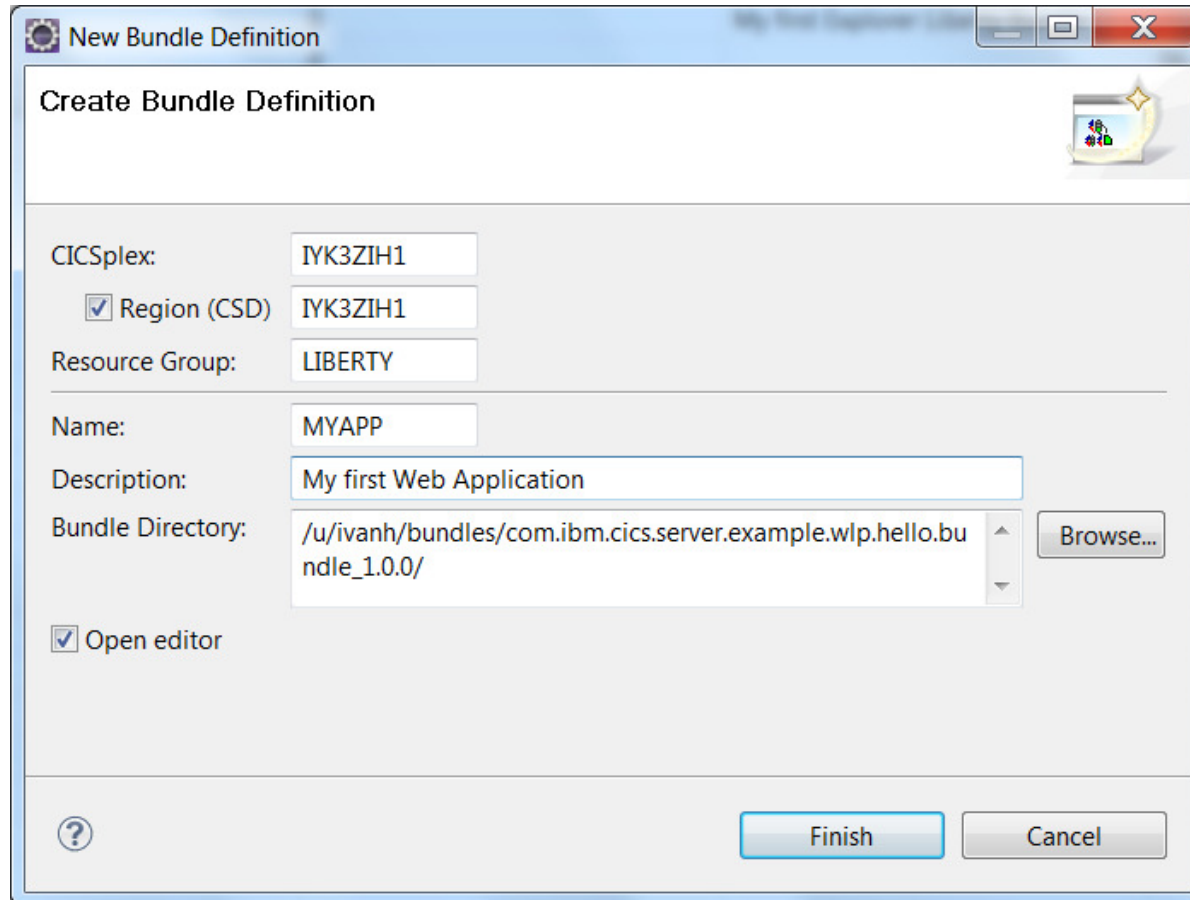
9) Export the CICS bundle project



10) Pick a zFS location for the CICS bundle project



11) Create a CICS bundle definition to control the life-cycle of the Application



New Bundle Definition

Create Bundle Definition

CICSplex: IYK3ZIH1

Region (CSD) IYK3ZIH1

Resource Group: LIBERTY

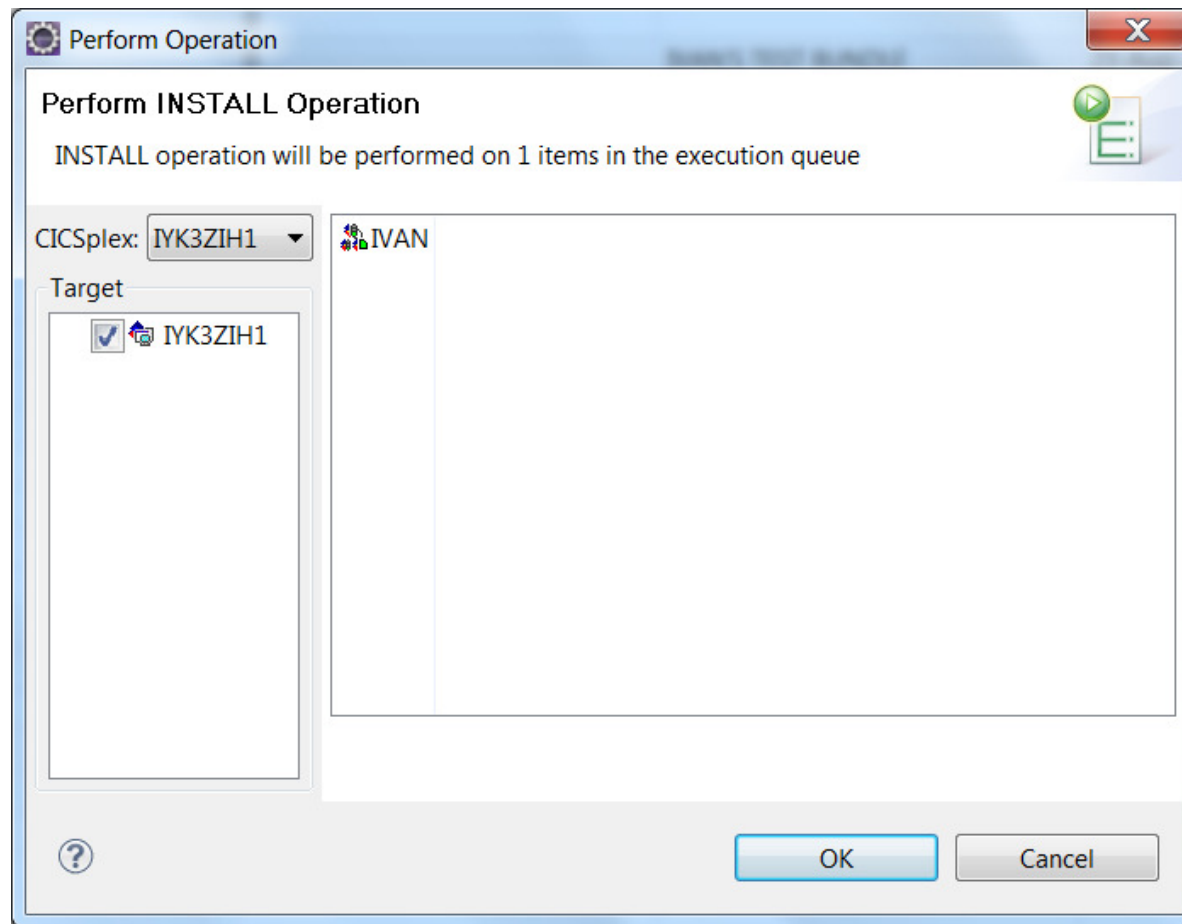
Name: MYAPP

Description: My first Web Application

Bundle Directory: /u/ivanh/bundles/com.ibm.cics.server.example.wlp.hello.bundle_1.0.0/

Open editor

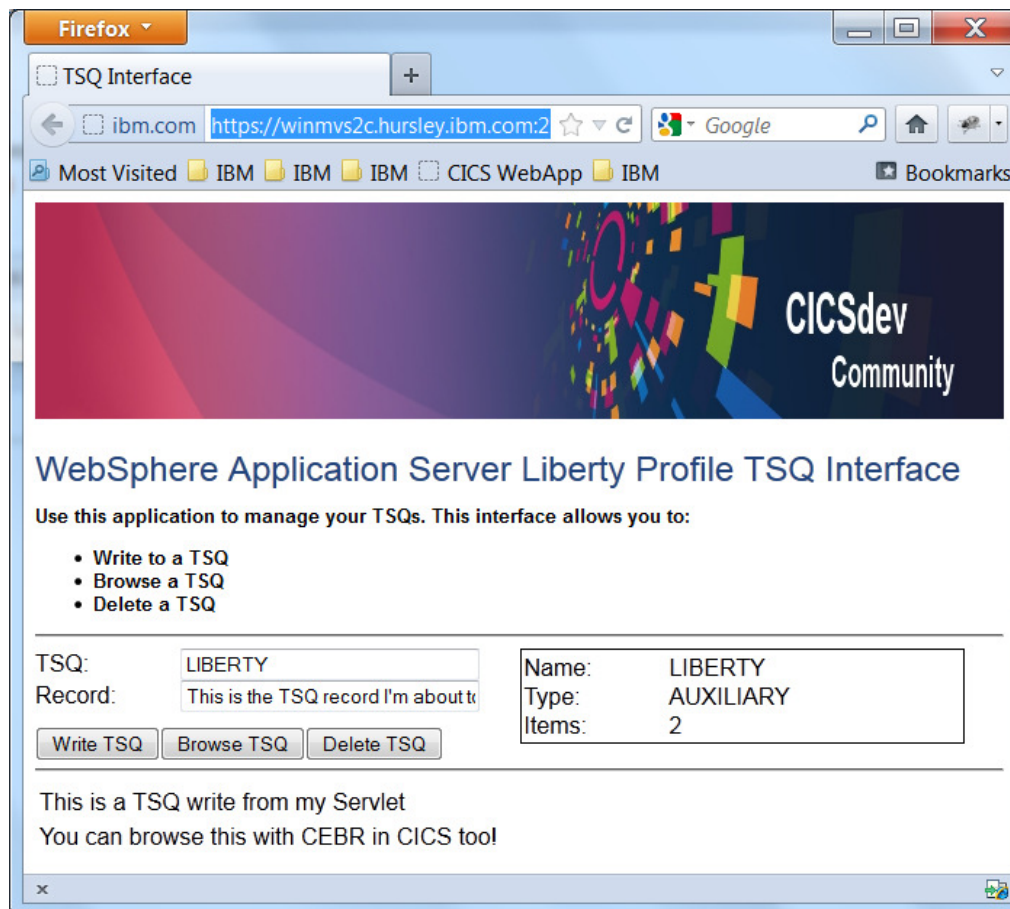
12) Install the CICS bundle definition



12) Run the application from your Web browser

CWWKT0016I: Web application available (default_host):

<http://winmvs2c.hursley.ibm.com:27245/com.ibm.cics.server.example.wlp.tsq.web/>





Agenda

- What is the WebSphere Application Server Liberty Profile?
- Liberty Profile on z/OS
- Liberty Web Container in CICS TS
- Step-by-step running a Web app in CICS
- **Further details of Liberty in CICS**
- Liberty Features in CICS
- Summary and Q&A

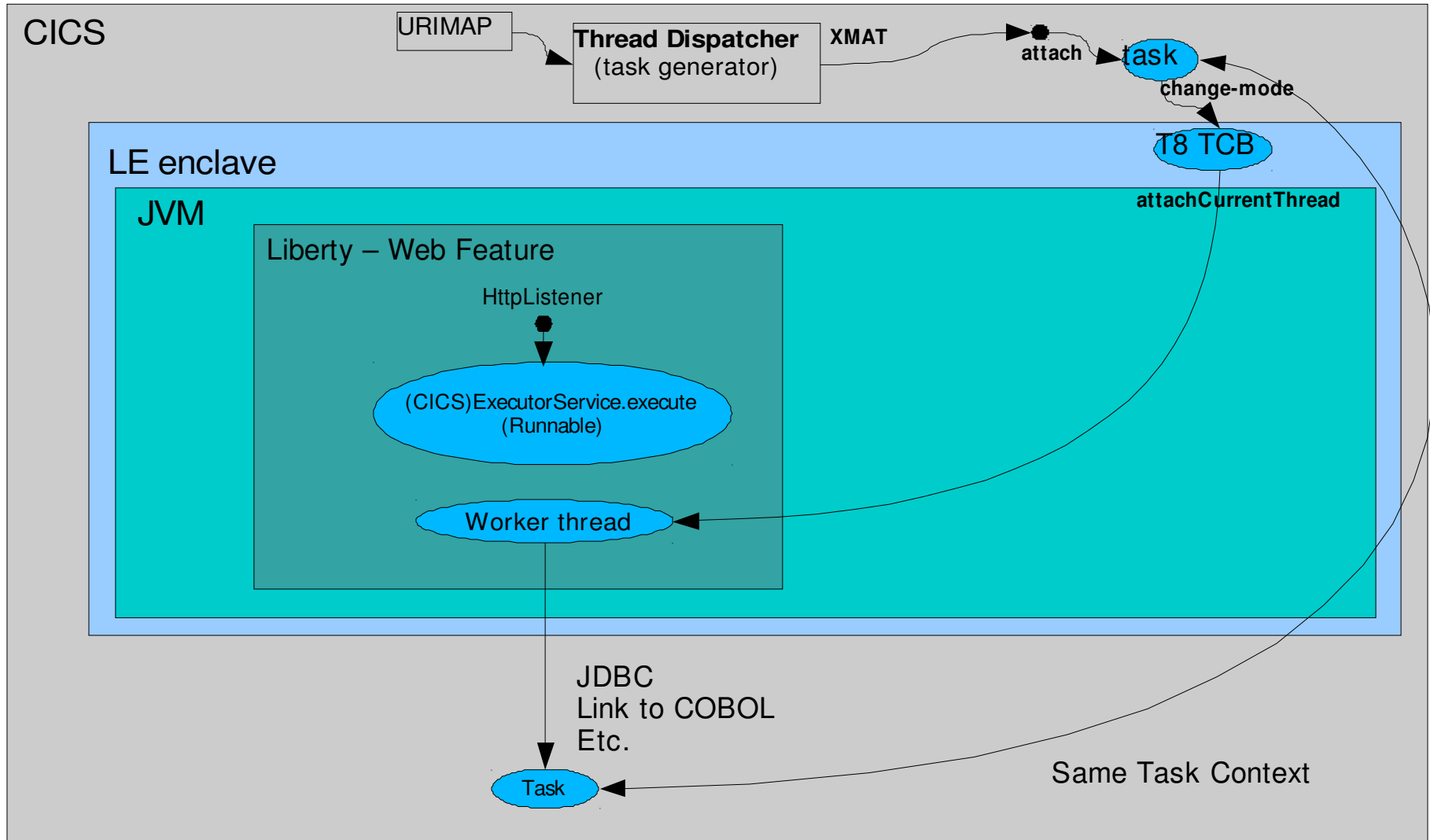
JSP/Servlets plus JCICS / JDBC / Cobol



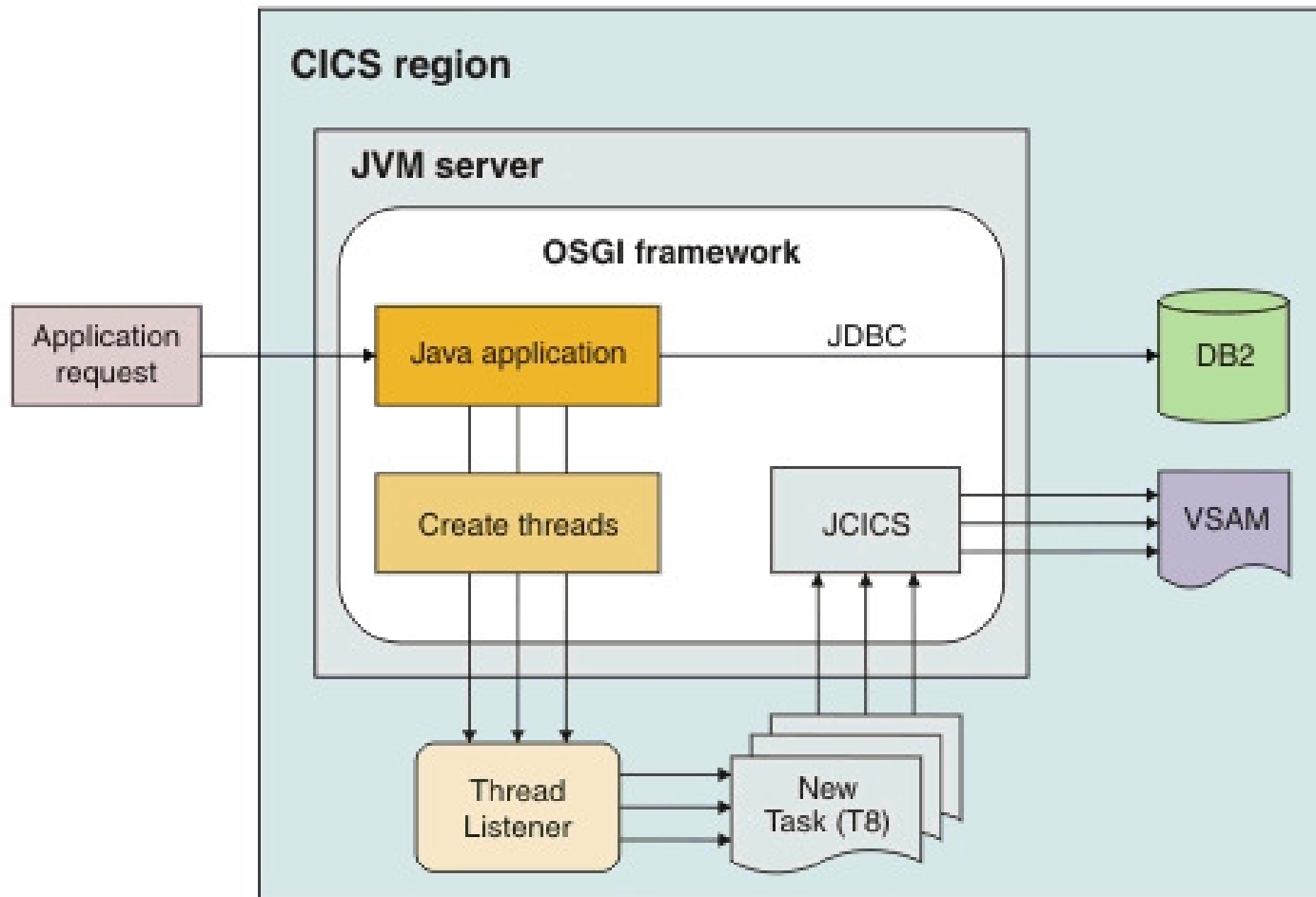
```
File Edit Source Refactor Navigate Search Project Run Window Help
HelloWorld.java TsqInfo.java index.jsp
Team My W Feed
All Project and Team Areas (60 of 91 areas selected)
Repository Connections
CICS Development and Service Environment [jazz1
CICS RFE Project [jazz104.hursley.ibm.com]
CICS TS for zOS [jazz104.hursley.ibm.com]
COBOL Container for WAS [jazz104.hursley.ibm.com]
Eclipse Tooling Development [jazz104.hursley.ibm.com]
SCM CICS TS for zOS [jazz104.hursley.ibm.com]
Debug
Favorites
Feeds
My Repository Workspaces
My Team Areas
Work Item History
Project Explore Remote System
WebContent
images
META-INF
WEB-INF
index.jsp
response.jsp
splat.jsp
HostConnectProjectFiles
org.example.mysql
org.example.mysql.ui
RTC-LP
Tivoli CDM
Chang Tag Cl Work I Probl Builds Repo Antz Histor Prope Remo Progr Pendi Navig Cons Team Search
Search for ID or Text CICS TS for zOS Writable Smart Insert 53 : 13 <No Current Work>
```

```
22  */
23  public TsqInfo() {
24      super();
25      // TODO Auto-generated constructor stub
26  }
27
28  /**
29   * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
30   */
31  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32
33      // obtain the input values from the request
34      String tsq = request.getParameter("tsq");
35
36      System.out.println("Tsq INFO is: " + tsq);
37
38      // create a tsq
39      Tsq tsqQ = new Tsq();
40      tsqQ.setName(tsq);
41
42      int length = 0;
43      try{
44          length = tsqQ.readItem(1, new ItemHolder());
45      } catch(Exception e){
46          e.printStackTrace();
47      }
48
49      String name = "<name>" + tsqQ.getName() + "</name>";
50      String type = "<type>" + tsqQ.getType().toString() + "</type>";
51      //String sysID = "<sysid>" + tsqQ.getSysId() + "</sysid>";
52      String lenStr = "<length>" + length + "</length>";
53      |
54      //System.out.println("Tsq SYSID is: " + tsqQ.getSysId());
55
56
57      response.getOutputStream().write("<info>".getBytes());
58
59      response.getOutputStream().write(name.getBytes());
60      response.getOutputStream().write(type.getBytes());
61      //response.getOutputStream().write(sysID.getBytes());
62      response.getOutputStream().write(lenStr.getBytes());
63
```

Hybrid Threads



Attaching hybrid threads – CJS� and CJSА





Benefits of Hybrid Threads

- Each 'Invocation' (Servlet Request) on a Hybrid Thread **is also** a CICS Transaction (Has a Tranid, Task Context etc).
- This gives you
 - A single common Transaction (UOW) and CICS Managed JDBC
 - Which can cross between Java and Cobol
 - Full JCICS API Access
 - In particular, LINK and access to VSAM
 - WLM (CICS WLM, Performance Classes etc).
 - Monitoring / Statistics
 - CICS Transaction Tracking / Association Data

CICS Security with Liberty profile



- Servlets run under default transaction CJSA with CICS default userid
 - Can override transaction (and userid) using URIMAP
 - Can use different transactions to authorize different users
- **SEC=YES** turns Security ON in Liberty JVM server
- Basic-auth only (http or https) – Client certificates not yet supported
- **Client Application:** Web.xml needs <security_constraint> to run with Security
- **Liberty:** Server.xml will be updated by CICS automatically
 - <application-bnd>
- Role based Security not currently supported



URIMAP enhancements



- URIMAP provides CICS authorization via Transaction Security

- URIMAP allows context switch to a 'user' transaction
 - Transaction Security (URL mapped to transaction)
 - Monitoring and audit purposes
 - “Transaction class” support



Agenda

- What is the WebSphere Application Server Liberty Profile?
- Liberty Profile on z/OS
- Liberty Web Container in CICS TS
- Step-by-step running a Web app in CICS
- Further details of Liberty in CICS
- **Liberty Features in CICS**
- Summary and Q&A



Liberty Features (as of WAS 8.5.0)

- Bean validation
- Blueprint
- Java Database Connectivity (JDBC)
- Java Management Extensions (JMX)
- Java Persistence API (JPA)
- JavaServer Faces (JSF)
- JavaServer Pages (JSP)
- Secure Sockets Layer (SSL)
- Security, supported by either the basic user registry or a Lightweight Directory Access Protocol (LDAP) user registry
- Servlet
- Web application bundle (WAB)
- Web security
- zOS Security
- zOS Transactions



Liberty Features (for CICS TS V5.1)

- Bean validation
- Blueprint
- **Java Database Connectivity (JDBC)**
- Java Management Extensions (JMX)
- Java Persistence API (JPA)
- JavaServer Faces (JSF)
- **JavaServer Pages (JSP)**
- **Secure Sockets Layer (SSL)**
- Security, supported by either the basic user registry or a Lightweight Directory Access Protocol (LDAP) user registry
- **Servlet**
- Web application bundle (WAB)
- **Web security**
- zOS Security – **CICS Security**
- zOS Transactions – **CICS Transactions**



Summary

- WebSphere Application Server Liberty Profile
- Liberty Profile on z/OS
- Liberty Web Container in CICS TS
- Step-by-step running a Web app in CICS
- Further details of Liberty in CICS
- Liberty Features in CICS
- **Summary and Q&A**

Any Further Questions ?



I ♥ CICS

Complete your sessions evaluation online at SHARE.org/SFEval