Web Apps using Liberty Profile Technology in CICS

Ian J Mitchell
IBM System Z Middleware CTO

ianj_mitchell@uk.ibm.com

Session 15563, Monday 4th August 2014
Abstract

CICS TS V5.1 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

• Java Update for CICS TS
• What is Liberty?
• Liberty in CICS
• Deeper look at Liberty Technology in CICS
• Future Direction and Summary
Java Update

- Java 7 (64-bit) JVMServer
- Equinox 3.7 as the OSGi framework.
  - Implements the OSGi R4.3 specification
- WAS Liberty Profile 8.5.0 based Web Container
- IBM CICS SDK for WebSphere Application Server Liberty profile v5.1
- Eclipse 3.6.2
Introduction to the CICS Java Web Container based on WAS Liberty technology
What's Liberty?
WebSphere Application Server: 15 Years of Leadership and Trusted Delivery

WAS V6
WAS V6.1
WAS V7 Feature Packs
WAS V7 & V6.1 Feature Packs
WAS V7
WAS V8 Beta, GA
Web 2.0 & Mobile FEP
WAS v8.5 Alpha, Beta
Migration Toolkit Refresh
WAS Tooling Bundles
WAS v8.5  Beta 2
(including lightweight Liberty profile)
Intelligent Management
Java SE 7
Java SE 7
WAS HV
WAS HV Refresh
Migration Toolkit Refresh
WAS HV
WAS EC2 AMI
WAS V8 Alpha, Beta
WAS V7 Feature Packs
(OSGi, JPA 2.0, Modern Batch, CEA, Dynamic Scripting, WOLA IMS)
WAS HV Refresh
Migration Toolkit Refresh
WAS V8.5 Beta 2
WAS V8.5 Feature Packs
WAS V8.5 Feature Packs
(XML, CEA, SCA)
SAML & WOLA
WAS HV
WAS EC2 AMI
J2EE1.4
WAS V6.1
Feature
WAS V6.1
WAS V6
2005
2006
2007
2008
2009
2010
2011
2012
Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
If this is tWAS...

...this is Liberty (WAS)

...so is this

...or even this!
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
Configuration by Exception

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

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

Features control what's available in the runtime.

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

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

Includes can be used to implement an extensible configuration model.

References can be used in multiple elements to point to and share a common definition.
“CICS TS V5.1 offers a fast and lightweight Java web container, providing developers with the rich features of the Java Servlet and JavaServer Pages (JSP) specifications, and fast local access to your existing CICS applications and data. Built on WebSphere Application Server Liberty 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. “
FAST. LIGHTWEIGHT. LOCAL.
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

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
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
Benefits for CICS

- Provides “off the shelf” Web-server capabilities (JSPs and Servlets)
- Potential to re-use even more WebSphere technology in CICS.
- 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.
Nought to Web-App
Create a JVM server resource in Explorer, CEDA, or CPSM.
Configure the JVMProfile

- Copy the sample DFHWLP
- Check JAVA_HOME is correct.
- Uncomment the WLP_SERVER_HTTP_PORT and choose a unique port number.
- Point your JVM server definition at the new JVMProfile
Enable the JVM server
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 pmz6470srl-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 ] CWWKZ0058I: Monitoring dropins for applications.

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

Install Eclipse 4.2.2 (Juno) preferably JEE version, but Classic will suffice.
• Windows
• Linux

Install IBM CICS SDK for WebSphere Application Server Liberty profile v5.1

http://www.ibm.com/support/docview.wss?rs=1083&uid=swg24033579

Add CICS SDK to Eclipse.

- Via Installation Manager (IM)
- Via Eclipse “Help->Add New Software” (P2)
- Direct download
Create a Dynamic Web Project, or choose one of the Examples
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

    // obtain the input values from the request
    String tsq = request.getParameter("tsq");

    System.out.println("TsQ INFO is: " + tsq);

    TSQ tsqQ = new TSQ();
    tsqQ.setName(tsq);

    int length = 0;
    try {
        length = tsqQ.readItem(1, new ItemHolder());
    } catch (Exception e) {
        e.printStackTrace();
    }

    String name = "<name>" + tsqQ.getName() + "</name>";
    String type = "<type>" + tsqQ.getType().toString() + "</type>";

    String lenStr = "<length>" + length + "</length>";

    //System.out.println("TsQ SYSID is: " + tsqQ.getSysId());

    response.getOutputStream().write("<info>").getBytes();

    response.getOutputStream().write(name.getBytes());
    response.getOutputStream().write(type.getBytes());
    response.getOutputStream().write(sysID.getBytes());
    response.getOutputStream().write(lenStr.getBytes());
}
Export the CICS bundle project
Pick a zFS location for the CICS bundle project

![Export to z/OS UNIX File System dialog box]

- **Bundle project:** `com.ibm.cics.server.example.wlp.tsq.bundle`
- **Connection:** `winmvs2c`
- **Parent Directory:** `/u/ivanh/bundles/`
- **Bundle Directory:** `/u/ivanh/bundles/com.ibm.cics.server.example.wlp.tsq.bundle_1.0.0`
  - Options: Clear existing contents of Bundle directory

Finish dialog box to complete.
Create a CICS bundle definition to control the life-cycle of the Application

![New Bundle Definition window](image-url)
Install the CICS bundle definition
Run the application!

[AUDIT ] CWWKT0016I: Web application available (default_host):

WebSphere Application Server Liberty Profile TSQ Interface

Use this application to manage your TSQs. This interface allows you to:
- Write to a TSQ
- Browse a TSQ
- Delete a TSQ

TSQ: __________________________ Name: ________ Type: ________
Record: ________________________ Items: ________

[Write TSQ] [Browse TSQ] [Delete TSQ]
Putting it all together

Eclipse with Liberty Tools

- Deploy
- Enable
- Use

CICS

JVM server

Liberty

CICS Resources

zFS

zOS

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
The Technology
Principles

As little customization as we can get away with.

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.
Specifications and Standards

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
Hybrid Threads

CICS

LE enclave

JVM

Liberty – Web Feature

HttpListener

(CICS)ExecutorService.execute

(Runnable)

Worker thread

JDBC

Link to COBOL

Etc.

Task

Same Task Context

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
'Standard' CICS Listener Pattern

CICS

URIMAP

Thread Dispatcher
(task generator)

XMAT

task

change-mode

Open TCB

Worker thread

CICS Web Domain

HttpListener

Drive Appropriate Thread Dispatcher

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
JCICS – ExecutorService

- Thread.start equivalent (from Java concurrency package)
- A standard Java pattern for dispatching runnable code to threads.
- CICS provides “CICSExecutorService” - to create CICS capable threads.
- CICSExecutorService registered with OSGi registry, can be obtained and used by 'vendor' products and applications.
- A convenience method provided called “CICSExecutorService.runAsCICS()”
- Liberty requests an ExecutorService from the OSGi service registry. When running in CICS JVM server, it is given the CICSExecutorService which produces JCICS enabled threads for Liberty to run servlets on.
Benefits of Hybrid Threads

- Each 'Invocation' (think 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
Servlets run under default transaction CJSA with CICS.

SEC=YES turns Security ON.

Basic-auth only (http or https) – Client cert 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 supported.
URIMAP enhancements for Liberty

- URIMAP provides CICS authorisation via Transaction Security

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

HttpResponse

401 - not auth
403 – not allowed
500 – error
503 - Disabled
OK

Web.xml

<security_constraint>

WebApp

Liberty

JVM server

CICS

URI MAP

Tran

SEC=YES

Web Client

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
Part 4 – Summary and Future
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)
- JAX-RS
- 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 GA)

- Bean validation
- Blueprint – via Service Stream
- Java Database Connectivity (JDBC)
- Java Management Extensions (JMX)
- Java Persistence API (JPA)
- JavaServer Faces (JSF)
- JavaServer Pages (JSP)
- JAX-RS, JSON – via Service Stream
- 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) – via Service Stream
- Web security
CICS TS V5.2 beta – Liberty runtime extensions

Liberty 8.5.5
Integrated and
optimized for CICS
Web workloads

IBM Java SDK V7.1

JTA
Java global transactions

JDBC
JNDI naming
DB2 DataSource

JEE Roles
RACF keyrings
LTPA single signon
Form based security
SSL client authentication
Trust Association Interceptors

Security

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
CICS Liberty Roadmap - 2014

CICS TS V5.1
- Liberty 8.5.0
- Servlet/jsp
- Explorer SDK for Web

V5.1 APAR PM85279
- JAX-RS, JSON

V5.1 APAR PM80214
- Liberty 8.5.0.1
- EBA support

V5.1 APAR PM91667
- Liberty 8.5.5

CICS TS V5.2 open beta
- Liberty 8.5.5.1
- JTA
- JDBC
- zosSecurity, appSecurity
- jndi,

WAS 8.5.0.1
- Spring 2012

WAS 8.5.0.2
- Spring 2013

WAS 8.5.5
- Liberty core
- Liberty base
- EJB, JMS, clusterring, jax-ws

WAS 8.5.5.1
- April 2013

WAS 8.5.5.2
- Oct 2013

WAS Liberty Vnext alpha
- JCA 1.6
- Web sockets
- EJB 3.2

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval.
WAS/Liberty feature set

New in Liberty 8.5.5
CICS/Liberty feature set - CICS TS V5.2 Open Beta

Feature Manager
HTTP Transport
Application Manager

zosSecurity
CICS TS V5.2 beta
CICS TS V5.1

cicsts:security-1.0
cicsts:jdbc-1.0
jndi
beanvalidation
appSecurity
jdbc

zosSecurity
CICS TS V5.2 beta
CICS TS V5.1

cicsts:security-1.0
cicsts:jdbc-1.0
jndi
beanvalidation
appSecurity
jdbc

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
Java Transaction API (JTA)

JTA provides coordination of updates across CICS UOW and a remote XA Resources:
- JDBC type 4 database driver
JTA – UserTransaction

InitialContext ctx = new InitialContext();
UserTransaction tran =
    (UserTransaction)ctx.lookup("java:comp/UserTransaction");

DataSource ds = (DataSource)ctx.lookup("jdbc/SomeDB");
Connection con = ds.getConnection();

// Start the User Transaction
tran.begin();

// Perform updates to CICS resources via JCICS API
// Access DataSource

if (allOk) {
    // Commit updates on both systems
    tran.commit();
} else {
    // Backout updates on both systems
    tran.rollback();
}
JTA commit() with CICS updates and additional RM

- User appln
  - UserTransaction.begin()
    - POST_BEGIN
    - enlistResource()
  - JCICS update TSQ
  - JDBC

- JTM
  - UOWEventListenerTransaction
  - CICSXAResource
  - OT domain

- RM domain
  - T4
  - JDBC

- UserTransaction.commit()
  - XA_end
  - XA_prepare
  - XA_prepare
  - XA_commit
  - XA_commit
CICS TS V5.2 – Liberty Security

- RACF keyrings for SSL
- AppSecurity
  - Authentication
    - Basic authentication
    - Form logon
    - TAI/ JAAS
    - SSL client authentication
    - Customer user registry
  - Authorisation
    - JEE roles
    - EJB roles
    - CICS Transaction/Resource security
    - SyncToOS_thread – USS security
Liberty JVM server security

- WLP Angel process used to access authorized services
  - SAF password authentication
  - SAF role authorization (EJBROLEs)

- All WLP application security options supported in CICS TS V5.2 Open Beta
  - LTPA/SSO provides optimized performance for authentication
  - TAI/JAAS modules can be used to customize authentication
Liberty JVM server - URIMAP

- Liberty 'threads' run under default transaction 'CJSA'

- URIMAP extended for type 'JVMSERVER'
  - Allows setting of a User transaction other than CJSA
    - Integrates with CICS Transaction Security
    - Used for monitoring and audit purposes
    - Transaction class (TCLASS) support for scheduling

- CICS Task userid determined as follows:
  - Protected URI, use USERID from Http request (basic-auth header)
  - Unprotected URI, use USERID value from URIMAP
  - No URIMAP, or no USERID in URIMAP, use CICS DFLTUSER
CICS Security with Liberty Profile

- V5.1 - Security credentials provided by basic-auth (http or https) or URIMAP.
- V5.2 Open Beta – Form logon, SSL client cert mapping,

CICS transaction security pre-requisites:
- SEC=YES (triggers JVM server to add 'CICS' UserRegistry to Liberty)
- Optional use of URIMAP to switch to 'user transaction'
- <application-bnd> in server.xml must be present (CICS bundle deployed apps always get this added when SEC=YES)
- <security-constraint> in Client application “web.xml” must be present (Explorer SDK provides examples/template)
Application:
Web.xml – Example <security-constraint>

```xml
<security-constraint>
  <display-name>examples.web_SecurityConstraint</display-name>
  <web-resource-collection>
    <web-resource-name>examples.web</web-resource-name>
    <description>Protection area for examples.web</description>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description>All Authenticated users of examples.web</description>
    <role-name>cicsAllAuthenticated</role-name>
  </auth-constraint>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Must match application-bnd on Server

Best practice to force HTTPS
Server.xml – Example <application-bnd>

```xml
<server>
    <!-- Include file for CICS bundle installed applications -->
    <application id="examples.web" name="examples.web" type="war"
                 location="${server.config.dir}/installedApps/examples.web.war">
        <application-bnd>
            <security-role name="cicsAllAuthenticated">
                <special-subject type="ALL_AUTHENTICATED_USERS"/>
            </security-role>
        </application-bnd>
    </application>
</server>
```

Must match Web.xml role
Where are we going?

- Equivalence with WebSphere Liberty Profile feature set
  - Focus on API – JAXWS, JMS, JCA, EJB….
  - Qos (JMX, dynacache)

- Program linkage to Liberty services
  - LINK interface to EBAs in Liberty OSGi framework

- Java development and debugging
  - Improved testing/debugging with JCICS
  - JCICS modernisation

- JVM server
  - Scalability
  - Logging
  - Monitoring

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
Summary of Key Benefits

**Local. Lightweight. Fast.** Web Applications run locally in CICS with direct access to CICS data and resources. No adapters, no converters, same address space.

**Modular design.** Architected in a modular way using OSGi, the server only enables and starts the features required by the applications and configuration. If you're not using a feature, it won't start in your server runtime.

**Standard tools for developers.** Familiar, industry standard tools with Eclipse and Dynamic Web Projects. CICS Explorer SDK enhances the deployment experience.

**Dynamic runtime.** Features can be added to the server dynamically, using the OSGi framework, while the server is running, with zero downtime and server restarts. Similarly server and application config can be updated without the need to restart.

**Portable.** Presentation logic in Servlets, business logic in OSGi bundles. Servlets are portable across runtimes. Bundles provide componentization.

**Eclipse based tools.** The eclipse tools for the Liberty Profile are small and very well integrated with the Liberty Profile environment.

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
Thank You

Merci

Danke

Gracias!

Obrigado

Bedankt

多謝

多謝

감사합니다