11442 CICS and Java: How the JVM Server transforms Java in CICS

Matthew Webster
matthew_webster@uk.ibm.com
CICS Transaction Server for z/OS V4.2

Events

• System Events
• Assured Events
• Lifecycle Management

Management

• Transaction Tracking
• Workload Management
• Password Phrases

Java

• 64-bit Applications
• Multithreaded Server
• OSGi Management

Scalability

• More Threadsafe
• Optimised Threadsafe
• 64-bit Exploitation

Connectivity

• Axis2 Web Services
• Web Services Offload
• HTTP & IP Extensions

new and enhanced capability across five major technology areas
Related CICS TS V4.2 Sessions

- **Monday**
  - 11420: CICS TS V4.R2 Technical Overview
  - 11454: CICS TS V4.R2 User Experience

- **Tuesday**
  - 11434: CICS Explorer - A System Programmer Perspective

- **Wednesday**
  - 11439: Event Processing: Insight into Your CICS Systems and Business
  - 11442: CICS and Java: How the JVM Server transforms Java in CICS

- **Thursday**
  - 11448: Core Foundations and Scalability
  - 11450: CICS WLM: Improved Throughput from Optimized WLM

- **Friday**
  - 11458: Modernizing CICS -- Hands-on Lab, Part 1 of 2
  - 11459: Modernizing CICS -- Hands-on Lab, Part 2 of 2
Java

- 64-bit Applications
- Multithreaded Server
- OSGi Management
Themes & Topics

- Brief History
- User Roles
- Platform & Tools
- Demonstration
- Future
BRIEF HISTORY
CICS TS V4.2 JVM Server

- **New 64-bit Java runtime environment**
  - Increases the number of JVMs that can be used in a single CICS region
  - Removes the constraints on heap storage for each JVM
  - Provides performance advantages when running on IBM zEnterprise 196 (z196) hardware

- **JVM server for user programs**
  - Run multiple Java transactions as threads in a single JVM server process
  - Simplified the setup and management of the CICS Java runtime environment
  - Improved portability for Java applications and tools deployed in CICS

- **OSGi bundles for development, deployment & management of Java applications**
  - Java applications packaged as one or more industry standard OSGi bundles and then deployed/installed as a CICS BUNDLE
  - Removes the need to load Java applications from a statically defined class path
  - Provision of application isolation and versioning, cross-package prerequisite checking, and simplified package redeployment
CICS TS V4.2 Pooled JVM

- **Enterprise Java Bean Support**
  - Enterprise Java Bean (EJB) 1.1 specification and the associated CICS EJB Server components was stabilized in CICS TS V4.1
  - Not available in JVM server
  - Encouraged to migrate EJB applications to be Java SE components and make them available through web services or the JEE Connector Architecture (JCA)

- **Java pool infrastructure**
  - Infrastructure was stabilized in CICS TS V4.1
  - Encouraged to migrate Java applications to OSGi bundles
  - See *Migrating applications using the CICS Explorer SDK* in the V4.2 Information Center
What is a JVM server…?

- a new CICS resource containing a **long-running** JVM.
- the strategic direction of Java in CICS
- a JVM that serves **multiple transactions concurrently**.
- a JVM in which applications/tasks run as OSGi bundles.
JVM server vs existing Java support?

### Single JVM - serves many tasks (reduced storage)
- Concurrent, multi-threaded, up to 256 threads per JVM server

### Pool of JVMs - each serves only a single task
- Java Program Isolation

### Differences
- **T8 (CICS key)**: J8 (CICS key), J9 (User key)
- **MAXTHRTDCBS (automatically calculated)**: up to max of 1024 per region
- **MAXJVMTCBS**, SIT parm
- More standard Server model (+ data-sharing)
- Difficult, convoluted to share data and state.
- Dynamic update and replace of modules
- JVMs must be restarted to effect changes
Attaching work to a JVMSERVER?
CICS Explorer SDK

- **CICS Explorer SDK V1.0**
  - Introduced with CICS TS V4.1
  - Plug-in development for CICS Explorer

- **CICS Explorer SDK V1.1**
  - Introduced with CICS TS V4.2
  - CICS Java development

- **CICS Explorer meets Eclipse IDE**
  - Integrated documentation, examples & wizards for simplified development & deployment
  - Support for migrating applications using conversion, wrapping or injection
  - Supports the development of Java applications for both a JVM server and pooled JVM
USER ROLES
Dave the Developer

- **Goals**
  - Dave wants to write code

- **Tools and Products**
  - Development tools (e.g. Eclipse, Rational Application Developer)
  - Web browser

- **Tasks**
  - Tasked with implementing individual application components according to predefined contract

- **Skills / Education**
  - Background in software development
  - Familiar with one or more languages/technologies e.g. Java.
Steve the Systems Programmer

- **Goals**
  - Work with a test and production environment which empowers the business user and is up 24/7.

- **Tools and Products**
  - CICS Explorer
  - TSO/ISPF
  - OMEGAMON® for z/OS

- **Tasks**
  - Manage the incoming requests for new roles and new applications

- **Skills / Education**
  - z/OS knowledge
  - Trouble shooting
  - Programming (e.g., JCL, COBOL)
DEMONSTRATION
Demonstration: CICS Explorer & CICS Explorer SDK

- **Goals**
  - Run a Java application in CICS
  - Demonstrate different application development & systems management roles
  - End-to-end experience with CICS Explorer SDK

- **Dave the Developer**
  - Will create, deploy & test the application
  - Can troubleshoot application problems

- **Steve the System Programmer**
  - Will create the Java environment and install & manage the application
  - Can troubleshoot application dependency problems

- **Setup**
  - Anaheim: Mac OS X 10.7 + Java SE 6 (Update 26) + Eclipse 3.6.2 (Cocoa 64-bit) + CICS Explorer SDK V1.1
  - Hursley: CICS TS V4.2 (CICSplex)
  - Java and CICS SM perspectives in separate workbench windows
CICS Explorer & CICS Explorer SDK Demonstration

Java - com.ibm.cics.server.examples.hello/src/examples/hello/HelloCICSWorld.java - Eclipse SDK

```
package examples.hello;

import com.ibm.cics.server.ComAreaHolder;

public class HelloCICSWorld
{
    public static void main(ComAreaHolder CAH)
    {
        Task t = Task.getTask();
        if (t == null)
        {
            System.err.println("HelloCICSWorld example: Can't get Task");
        }
        else {
            t.out.println("Hello from a Java CICS application");
        }
    }
}
```

This class provides a set of methods and variables that correspond to a CICS task.

Since CICS TS version: 13
Since package version: 10
1. Configure and install JVM server
2. Create and install (sample) CICS resources
3. Create CICS bundle

Steve the Systems Programmer
Edit supplied DFHOSGI profile and “File > Save As…”
Edit supplied DFHOSGI profile and “File > Save As…”

```bash
# JAVA_HOME specifies the location of the Java directory.
#JAVA_HOME=/usr/lpp/java/J6.0/
JAVA_HOME=/java/java601_bit64_GA/J6.0.1_64
#
# Set the current working directory. If this environment variable is
# set, a change to the specified directory is issued before the JVM
# is initialized, and the STDIN, STDOUT and STDERR streams are
# allocated to this directory.
#
# If you do not specify this option, the current working directory is
# left unchanged and the STDIN, STDOUT and STDERR streams are allocated
# to the /tmp directory.
#
#WORK_DIR=.
WORK_DIR=/u/webster/CICSEXPI
```
Edit supplied DFHOSGI profile and “File > Save As…”
Use “Show SIT Parameters” to find JVM Profile Directory

![SIT Parameters Window](image)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Install</td>
<td>AUTO</td>
</tr>
<tr>
<td>CICS SVC</td>
<td>240</td>
</tr>
<tr>
<td>CPSM Connection</td>
<td>LMAS</td>
</tr>
<tr>
<td>GM Text</td>
<td>‘WELCOME TO EXPLORER LMAS CICS 670, REGION IYCWEGG1’</td>
</tr>
<tr>
<td>SIT Suffix</td>
<td>PE</td>
</tr>
<tr>
<td>SRBSVC</td>
<td>241</td>
</tr>
<tr>
<td>Start</td>
<td>INITIAL</td>
</tr>
<tr>
<td>SYSIDNT</td>
<td>EGG1</td>
</tr>
<tr>
<td>USS Home</td>
<td>/cics/cics670</td>
</tr>
<tr>
<td>CSD</td>
<td></td>
</tr>
<tr>
<td>Auto Install Group Lists</td>
<td>(C42E67)</td>
</tr>
<tr>
<td>CSD Concurrent Requests</td>
<td>6</td>
</tr>
<tr>
<td>CSD Disposition</td>
<td>SHR</td>
</tr>
<tr>
<td>CSD Forward Recovery Log</td>
<td>1</td>
</tr>
<tr>
<td>CSD Journal ID</td>
<td>1</td>
</tr>
<tr>
<td>CSD Recovery</td>
<td>ALL</td>
</tr>
<tr>
<td>CSD RLS</td>
<td>YES</td>
</tr>
<tr>
<td>Dump</td>
<td></td>
</tr>
<tr>
<td>System Dump Max</td>
<td>5</td>
</tr>
<tr>
<td>File Control</td>
<td></td>
</tr>
<tr>
<td>RLS</td>
<td>YES</td>
</tr>
<tr>
<td>Java</td>
<td></td>
</tr>
<tr>
<td>JVM Profile Directory</td>
<td>/u/mqtest/jvmprofs/IYCWEGG1</td>
</tr>
<tr>
<td>Network</td>
<td></td>
</tr>
<tr>
<td>Application ID</td>
<td>IYCWEGG1</td>
</tr>
<tr>
<td>Intersystem Communication</td>
<td>YES</td>
</tr>
<tr>
<td>IRC Start</td>
<td>YES</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>YES</td>
</tr>
</tbody>
</table>
INSTALL sample DFH$JVM JVM server
INSTALL sample DFH$JVM JVM server
JVM server DFH$JVMS ENABLED
Check IYCWEGG1.DFH$JVMS.dfhjvmtrc for any problems
List of system and “middleware” OSGi bundles installed

```
```

0 org.eclipse.osgi 3.6.1.R36x_v20100806 ACTIVE
1 org.eclipse.osgi.services 3.2.100.v20100503 ACTIVE
2 org.eclipse.equinox.log 1.2.100.v20100503 ACTIVE
3 com.ibm.cics.server 1.300.0 ACTIVE
4 com.ibm.cics.ras 1.300.0 ACTIVE
5 com.ibm.ras 1.300.0 ACTIVE
6 com.ibm.record 1.0.0 ACTIVE
7 com.ibm.cics.domains 1.300.0 ACTIVE
8 com.ibm.cics.osgi 1.0.0 ACTIVE
9 com.ibm.cics.osgi.impl 1.0.0 ACTIVE
Lots of useful information about JVM server

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td></td>
</tr>
<tr>
<td>Basedefinever</td>
<td>0</td>
</tr>
<tr>
<td>CICS Release</td>
<td>E670</td>
</tr>
<tr>
<td>Enable Status</td>
<td>ENABLING</td>
</tr>
<tr>
<td>JVM Created (GMT)</td>
<td></td>
</tr>
<tr>
<td>JVM Created (Local time)</td>
<td></td>
</tr>
<tr>
<td>JVM Profile</td>
<td>DFHOSGI</td>
</tr>
<tr>
<td>LERUNOPTS</td>
<td>DFHAXRO</td>
</tr>
<tr>
<td>Name</td>
<td>DFHSJVM5</td>
</tr>
<tr>
<td>PID</td>
<td>0</td>
</tr>
<tr>
<td>Region</td>
<td>YCWEGL1</td>
</tr>
<tr>
<td>Use Count</td>
<td>0</td>
</tr>
<tr>
<td>JVM Heap</td>
<td></td>
</tr>
<tr>
<td>GC Events (Major)</td>
<td>0</td>
</tr>
<tr>
<td>GC Events (Minor)</td>
<td>0</td>
</tr>
<tr>
<td>GC Heap Freed (Major)</td>
<td>0</td>
</tr>
<tr>
<td>GC Heap Freed (Minor)</td>
<td>0</td>
</tr>
<tr>
<td>GC Min Heap</td>
<td>0</td>
</tr>
<tr>
<td>GC Policy</td>
<td></td>
</tr>
<tr>
<td>GC Time (Major)</td>
<td>0</td>
</tr>
<tr>
<td>GC Time (Minor)</td>
<td>0</td>
</tr>
<tr>
<td>Heap</td>
<td>0</td>
</tr>
<tr>
<td>Init Heap</td>
<td>0</td>
</tr>
<tr>
<td>Max Heap</td>
<td>0</td>
</tr>
<tr>
<td>Peak Heap</td>
<td>0</td>
</tr>
<tr>
<td>Resource Signature</td>
<td></td>
</tr>
<tr>
<td>Threads</td>
<td></td>
</tr>
<tr>
<td>Max Threads</td>
<td>15</td>
</tr>
<tr>
<td>Peak Threads</td>
<td>0</td>
</tr>
<tr>
<td>Peak Waiting Threads</td>
<td>0</td>
</tr>
<tr>
<td>Sys Thread Peak Wait</td>
<td>0</td>
</tr>
<tr>
<td>Sys Thread Usage</td>
<td>8</td>
</tr>
<tr>
<td>Sys Thread Waits</td>
<td>0</td>
</tr>
<tr>
<td>Sys Thread Wait Time</td>
<td>0</td>
</tr>
<tr>
<td>Threads</td>
<td>0</td>
</tr>
<tr>
<td>Thread Waits</td>
<td>0</td>
</tr>
<tr>
<td>Thread Wait Time</td>
<td>0</td>
</tr>
<tr>
<td>Waiting Threads</td>
<td>0</td>
</tr>
<tr>
<td>Waiting Threads (Sys Thread)</td>
<td>0</td>
</tr>
</tbody>
</table>
1. Consult *CICS Java Developer Guide*
2. Set Target Platform
3. Create (example) application
4. Deploy application as CICS bundle

Dave the Application Developer
Empty workspace, empty CICS bundle
Developing applications using the CICS Explorer SDK

You can use the SDK to develop a Java application to run in any supported release of CICS. Different releases of CICS support different versions of Java, and the JCICS API has also been extended in later releases to support additional features of CICS. To avoid using the wrong classes, the SDK provides a feature to set up a target platform. You can define which release of CICS you are developing for and the SDK automatically hides the Java classes that you cannot use.

See the CICS Java Developer Guide in the SDK help for full details on how you can perform each of the following steps to develop and deploy applications.

**Procedure**

1. Set up a target platform for your Java development. The target platform ensures you use only the Java classes that are appropriate for the target release of CICS in your application development.

2. Create a plug-in project for your Java application development.

3. Develop your Java application using best practices. If you are new to developing Java applications for CICS, you can use the JCICS examples provided with the CICS Explorer™ SDK to get started. To use JCICS in a Java application, you must import the com.ibm.cics.server package.

4. Deploy your Java application in a CICS bundle to zFS. CICS bundles can contain one or more OSGi bundles and are the unit of deployment for your application in CICS. If you are running the Java application in a JVM server, you must know the name of the JVMSERVER resource in which you want to deploy the application.
Set Target Platform

Preferences

Target Platform

Add, edit and remove target definitions. The active target definition will be used as the target platform which workspace plug-ins will be compiled and tested against. New definitions are stored locally, but they can be moved to a project in the workspace and shared with others.

Target definitions:
- Running Platform (Active)

Locations:

Reload... Add... Edit... Remove... Share...

Restore Defaults Apply Cancel OK
Set Target Platform

**New Target Definition**

**Target Definition**

Create a new target definition.

 Initialize the target definition with:

- Nothing: Start with an empty target definition
- Default: Default target for the running platform
- Current Target: Copy settings from the current target platform
- Template: CICS TS V4.2 Runtime
JCICS version 1.300 (V4.2)
Java 1.6

New Target Definition

Target Content
Edit the name, description, and plug-ins contained in a target.

Name: CICS TS V4.2 Runtime

Target Environment
Specify the target environment. If left blank, the default environment variables from the host (running) platform will be used.

Operating System:
Windowing System:
Architecture:
Locale:

Java Runtime Environment
Specify the JRE or execution environment for this target. Selecting a named JRE will change the workspace default JRE setting.

- Default JRE
- JRE name: JVM Contents (MacOS X Default)
- Execution Environment: JavaSE-1.6
Select the new target
Add the examples (Hello, JCICS & Web) to the workspace
Add OSGi bundle projects
Add CICS bundle project
Errors in the Problems view for missing OSGi bundles
Deploy application as CICS bundle
Select CICS bundle directory
Transfer metadata files
Build, locally export & transfer OSGi bundle JAR files
Handoff CICS bundle to Steve
What is an OSGi Bundle?

HelloWorld

ClassA

ClassB
What is an OSGi Bundle?

HelloWorld
ClassA
ClassB
What is an OSGi Bundle?

**MANIFEST.MF**

Manifest-Version: 1.0
Created-By: IBM Corporation
What is an OSGi Bundle?

**MANIFEST.MF**

Manifest-Version: 1.0
Created-By: IBM Corporation
Bundle-SymbolicName: com.ibm.cics.server.examples.hello
Bundle-Version: 1.0.0
Bundle-RequiredExecutionEnvironment: J2SE-1.6
Import-Package: com.ibm.cics.server;version="4.2.0"
Export-Package: examples.hello
What is an OSGi Bundle?

**MANIFEST.MF**

```
Manifest-Version: 1.0
Created-By: IBM Corporation
Bundle-SymbolicName: com.ibm.cics.server.examples.hello
Bundle-Version: 1.0.0
Bundle-RequiredExecutionEnvironment: J2SE-1.6
Import-Package: com.ibm.cics.server;version="4.2.0"
Export-Package: examples.hello
```
What is an OSGi Bundle?

MANIFEST.MF

Manifest-Version: 1.0
Created-By: IBM Corporation
Bundle-SymbolicName: com.ibm.cics.server.examples.hello
Bundle-Version: 1.0.0
Bundle-RequiredExecutionEnvironment: J2SE-1.6
Import-Package: com.ibm.cics.server;version="4.2.0"
Export-Package: examples.hello
1. Install application as CICS bundle
2. Validate OSGi bundles and services
3. Ensure TCP/IP service and URI Map available
4. Run the application

Steve the Systems Programmer
Handoff CICS bundle from Dave
Handoff CICS bundle from Dave

Properties:

- Basescope: /u/webster/orlando/com.ibm.cics.server.examples
- Bundle Directory: OSGIMAW1
- CSDGroup: CICS bundle containing OSGi sample bundles
- Description: DFHSOSGB
- Name: ENABLED
- Status: N/A
- Userdata 1: N/A
- Userdata 2: N/A
- Userdata 3: N/A
- Version: 0

Error Log:

Fetching children of webster

© 2012 IBM Corporation
Install application as CICS bundle
CNX0211I Context: IYCWEGG1. Resource: BUNDLE. 1 records collected at 08-Aug-2011 20:04:19

<table>
<thead>
<tr>
<th>Region</th>
<th>Name</th>
<th>Status</th>
<th>Install Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>IYCWEGG1</td>
<td>DFHSOSGB</td>
<td>✔ ENABLED</td>
<td>08-Aug-2011 19:36:46</td>
</tr>
</tbody>
</table>
Check CICS bundle and bundle parts

### Bundle Parts

**Region** | **Name** | **Status** | **Install Time**
---|---|---|---
IYCWEGG1 | DFHSOSGB | ENABLED | 08–Aug–2011 19:36:46

**Region** | **Bundle** | **Bundle Part** | **Enable Status** | **Meta Data File** | **Part Class** | **Part Type**
---|---|---|---|---|---|---
IYCWEGG1 | DFHSOSGB | hello | ENABLED | hello.osgibundle | DEFINITION | http://www.ibm.com/xml
IYCWEGG1 | DFHSOSGB | jcics | ENABLED | jcics.osgibundle | DEFINITION | http://www.ibm.com/xml
IYCWEGG1 | DFHSOSGB | web | ENABLED | web.osgibundle | DEFINITION | http://www.ibm.com/xml
## OSGi bundle for Web example

<table>
<thead>
<tr>
<th>JVM Server</th>
<th>Symbolic Name</th>
<th>Version</th>
<th>State</th>
<th>Bundle</th>
<th>Bundle Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFH$JVMS</td>
<td>com.ibm.cics.server.examples.hello</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>hello</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>com.ibm.cics.server.examples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>com.ibm.cics.server.examples.web</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>web</td>
</tr>
</tbody>
</table>
### OSGi service for Web example

#### Table 1: OSGi Bundles

<table>
<thead>
<tr>
<th>JVM Server</th>
<th>Symbolic Name</th>
<th>Version</th>
<th>State</th>
<th>Bundle</th>
<th>Bundle Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFH$JVMS</td>
<td>com.ibm.cics.serverexamples.hello</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>hello</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>com.ibm.cics.serverexamples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
</tbody>
</table>

#### Table 2: OSGi Services

<table>
<thead>
<tr>
<th>JVM Server</th>
<th>Service Name</th>
<th>OSGi Bundle</th>
<th>Version</th>
<th>Service Status</th>
<th>Bundle</th>
<th>Bundle Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFH$JVMS</td>
<td>examples.hello.HelloCICSWorld</td>
<td>com.ibm.cics.serverexamples.hello</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>hello</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.hello.HelloWorld</td>
<td>com.ibm.cics.serverexamples.hello</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>hello</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.ProgramControl.ClassOne</td>
<td>com.ibm.cics.serverexamples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.ProgramControl.ClassTwo</td>
<td>com.ibm.cics.serverexamples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.ProgramControl.ClassThree</td>
<td>com.ibm.cics.serverexamples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.ProgramControl.ClassFour</td>
<td>com.ibm.cics.serverexamples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.TDQ.ClassOne</td>
<td>com.ibm.cics.serverexamples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.TSQ.ClassOne</td>
<td>com.ibm.cics.serverexamples.jcics</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>jcics</td>
</tr>
<tr>
<td>DFH$JVMS</td>
<td>examples.Web.Sample1</td>
<td>com.ibm.cics.serverexamples.web</td>
<td>1.0.0</td>
<td>ACTIVE</td>
<td>DFH$OSGB</td>
<td>web</td>
</tr>
</tbody>
</table>
### CICS PROGRAM for Web example

<table>
<thead>
<tr>
<th>Region</th>
<th>Name</th>
<th>Status</th>
<th>Use Count</th>
<th>Concurrent Use Count</th>
<th>Language</th>
<th>JVM Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>IYCWEGG1</td>
<td>DFJSJHE1</td>
<td>ENABLED</td>
<td>0</td>
<td>0</td>
<td>JAVA</td>
<td>examples.hello.HelloWorld</td>
</tr>
<tr>
<td>IYCWEGG1</td>
<td>DFJSJHE2</td>
<td>ENABLED</td>
<td>0</td>
<td>0</td>
<td>JAVA</td>
<td>examples.hello.HelloCICSWorl</td>
</tr>
<tr>
<td>IYCWEGG1</td>
<td>DFJSJWB1</td>
<td>ENABLED</td>
<td>0</td>
<td>0</td>
<td>JAVA</td>
<td>examples.Web.Sample1</td>
</tr>
</tbody>
</table>
Check IYCWEgg1.DFH$JVMS.dfhjvmtrc for any problems
Open the Internal Web Browser to test DFJ$JWB1
Enter Web example URL

http://cicsexp1host.hursley.ibm.com:27070/DFJSJWB1/
Success!

Web Sample1

Inbound Client Request Information:
Method: GET
Version: HTTP/1.1
Path: /DFJSWB1/
Request Type: HTTPYES
Query String: null

HTTP headers:
“CICS TS V5.1 open beta”

- **CICS Application Packaging**
  - New CICS bundle resources: LIBRARY, PROGRAM, TRANSACTION, URIMAP
  - Dependency management

- **CICS Application Lifecycle**
  - Simple Deploy-INSTALL-ENABLE-DISABLE-DISCARD life-cycle for multi-bundle, multi-language applications

- **WAS Liberty profile technology**
  - Java Servlets
  - JavaServer Pages (JSP)
  - Co-location of presentation and business applications
QUESTIONS
Summary

- **Platform**
  - Highly scalable runtime
  - Application lifecycle and versioning

- **Tools**
  - System programmer manages Java workloads using strategic interfaces consistent with other modern CICS workloads
  - Application developer leverages existing skills, tools and processes
  - Natural handoff between roles

- **Future**
  - “Auto-install” for CICS Java programs
  - Fully packaged Java and Java/traditional language applications
  - Liberty
Related CICS Java Sessions

**Monday**
11420: CICS TS V4.R2 Technical Overview
11423: CICS TS V4.R2 Planning Your CICS Upgrade

**Tuesday**
11434: CICS Explorer - A System Programmer Perspective

**Wednesday**
11441: Managing CICS Resources in a Unix File System: Best Practices
11442: CICS and Java: How the JVM Server transforms Java in CICS

**Thursday**
11449: CICS Question Box and Pot Luck

**Friday**
11458: Modernizing CICS -- Hands-on Lab, Part 1 of 2
11459: Modernizing CICS -- Hands-on Lab, Part 2 of 2
More Information

- **IBM CICS Explorer**

- **SHARE (Past Conferences)**
  - 8514: CICS Explorer Update
  - 8265: CICS JVM Server
  - 8272: Best Practices for CICS Systems Management
    [http://share.confex.com/share/116/webprogram/Session8272.html](http://share.confex.com/share/116/webprogram/Session8272.html)

- **Podcasts**

- **Blog**
  - [http://masterterminal.wordpress.com/](http://masterterminal.wordpress.com/)

- **developerWorks Community**

- **“CICS TS V5.1 open beta”**