WebSphere Application Server for z/OS
- Batch Update -

John Hutchinson
IBM

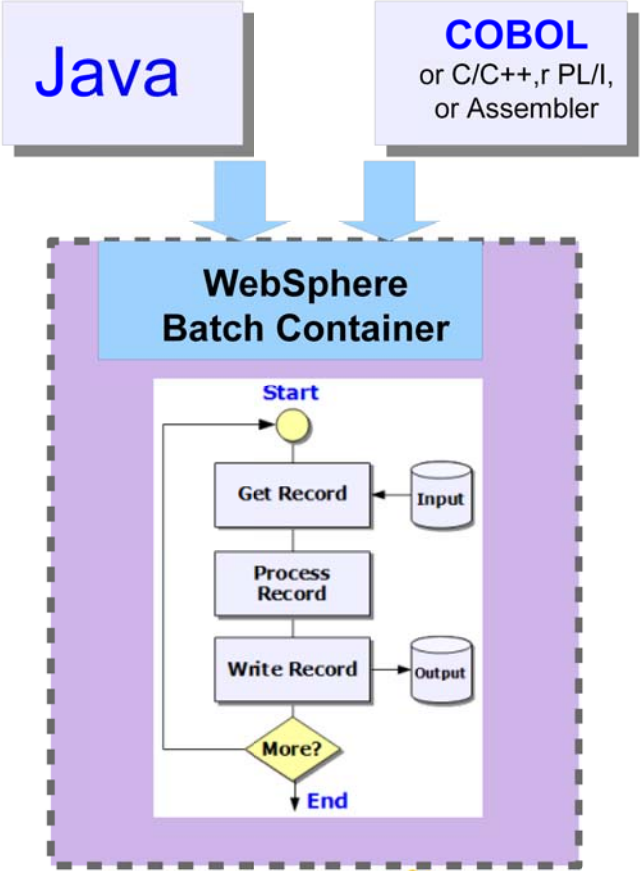
August, 2011 - Orlando
SHARE Session 09486

WebSphere Application Server Sessions

Day	Time	#	Title	Speaker	Room
Wednesday	3:00	9483	Using IBM's New Cross-Platform Installer on z/OS	Mierzejewski (Loos)	Oceanic 5
Thursday	8:00	9482	WAS Version 8 – Overview	Follis	Europe 2
Thursday	9:30	9486	WAS Version 8 – Batch Update	Hutchinson	Europe 2
Thursday	11:00	9485	WAS Version 8 – New z/OS Exploitation/Differentiation Features	Follis	Europe 2
Thursday	1:30	9484	WAS Version 8 – High Availability Enhancements	Follis	Europe 2
Thursday	3:00	9488	WAS - Back to Basics Part 1	Loos	Europe 2
Thursday	4:30	9489	WAS - Back to Basics Part 2	Stephen	Europe 2
Friday	8:00	9490	WAS for z/OS - Level 2 Update	Stephen	Europe 2
Friday	9:30	9487	WAS for z/OS – PotPourri	Follis, Hutchinson, Loos, Stephen, etc.	Europe 2

What's New with Batch (Compute Grid) on z/OS?

- *WebSphere Application Server Version 8 includes V7 Batch FeP*
  - Sub-set of function in XD Compute Grid
- *WebSphere XD Compute Grid Version 8 Enhancements:*
  - Configuration & Operational Improvements
  - Programming Framework *
  - Job Structure Enhancements *
  - Integration with JES Schedulers *
  - Parallel Job Management *
  - Job Classification & Control *
  - Job Usage Reporting *
  - Integration with CICS & COBOL *
- *Migration from Version 6.1.1*
- *Information Center & other Resources*



Java

COBOL
or C/C++,r PL/I,
or Assembler

WebSphere
Batch Container

Batch Feature Pack and XD Compute Grid

Here's a summary of the key features:

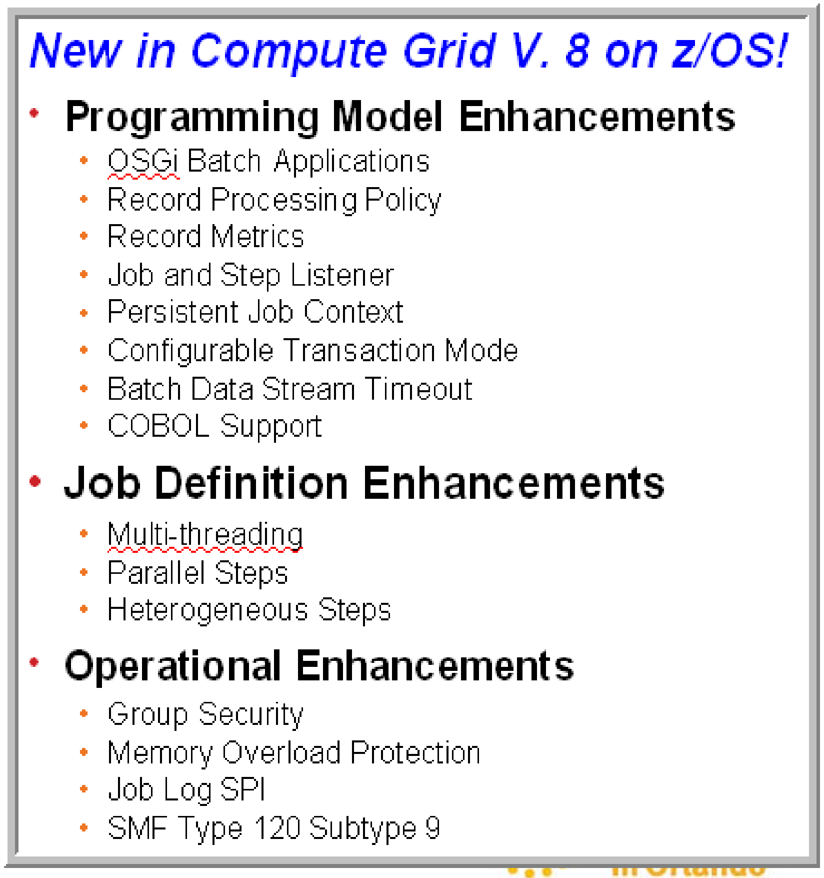
Feature Pack for Modern Batch (now part of WAS V.8)

Batch container environment
Job scheduler and dispatcher function
Declarative job control file (xJCL)
Development class libraries
Batch Data Stream (BDS)
Conditional multi-step job support
Ckpt processing leverages WAS trans. Mgr

WebSphere XD Compute Grid

Everything you see under "FP for Modern Batch" plus ...

Calendar & clock scheduling of jobs
Integration with ext. scheduler products
Usage reporting with SMF 120.20 & .9
WLM transaction classification *by job*
Application quiesce and update
Job submission pacing and throttling
Parallel job management & dispatching
Integration with COBOL and CICS

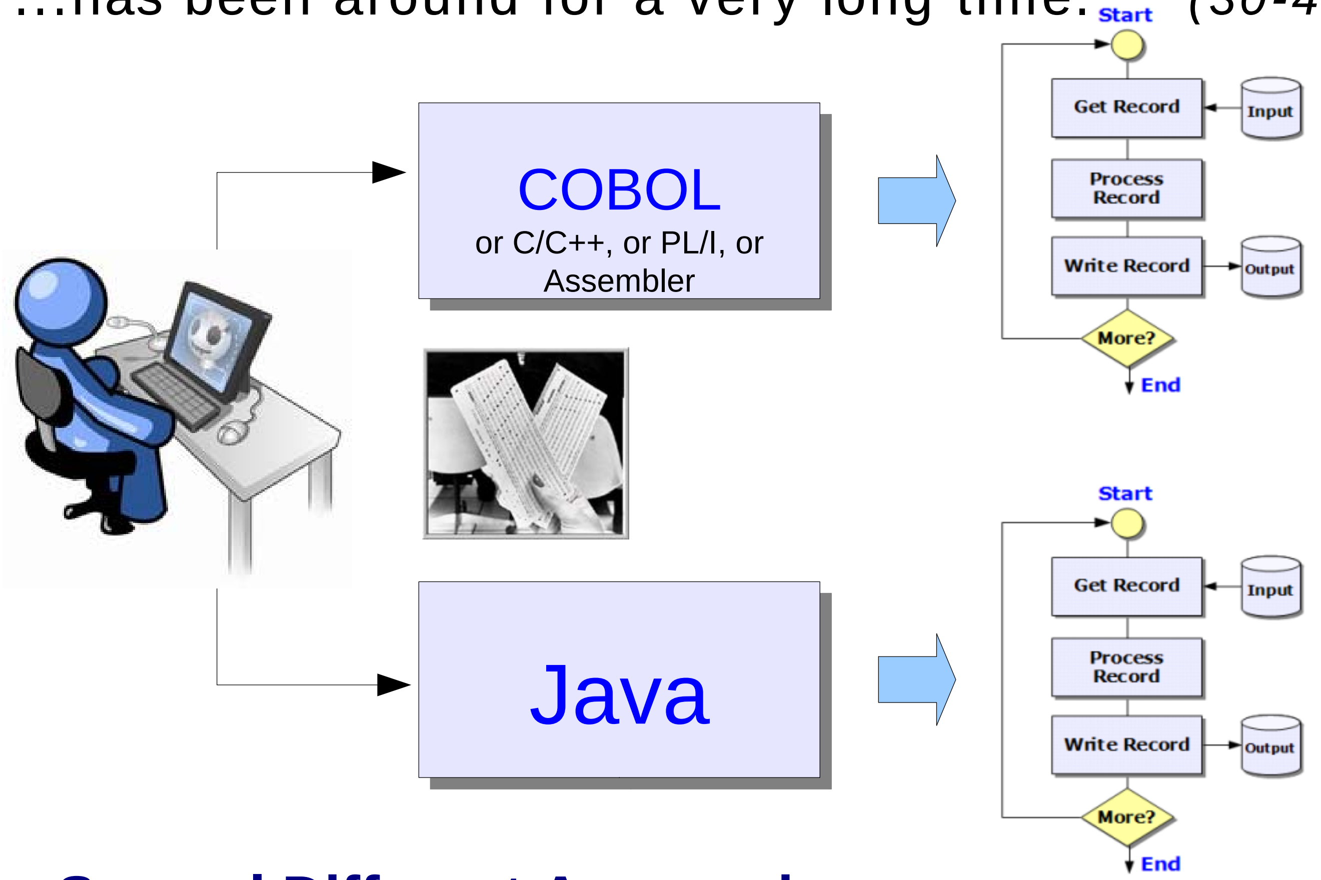


New in Compute Grid V. 8 on z/OS!

- **Programming Model Enhancements**
  - OSGi Batch Applications
  - Record Processing Policy
  - Record Metrics
  - Job and Step Listener
  - Persistent Job Context
  - Configurable Transaction Mode
  - Batch Data Stream Timeout
  - COBOL Support
- **Job Definition Enhancements**
  - Multi-threading
  - Parallel Steps
  - Heterogeneous Steps
- **Operational Enhancements**
  - Group Security
  - Memory Overload Protection
  - Job Log SPI
  - SMF Type 120 Subtype 9

Batch Processing ….

...has been around for a very long time. *(30-45 yrs?)*



COBOL
or C/C++, or PL/I, or Assembler

Java

Different programming languages ...

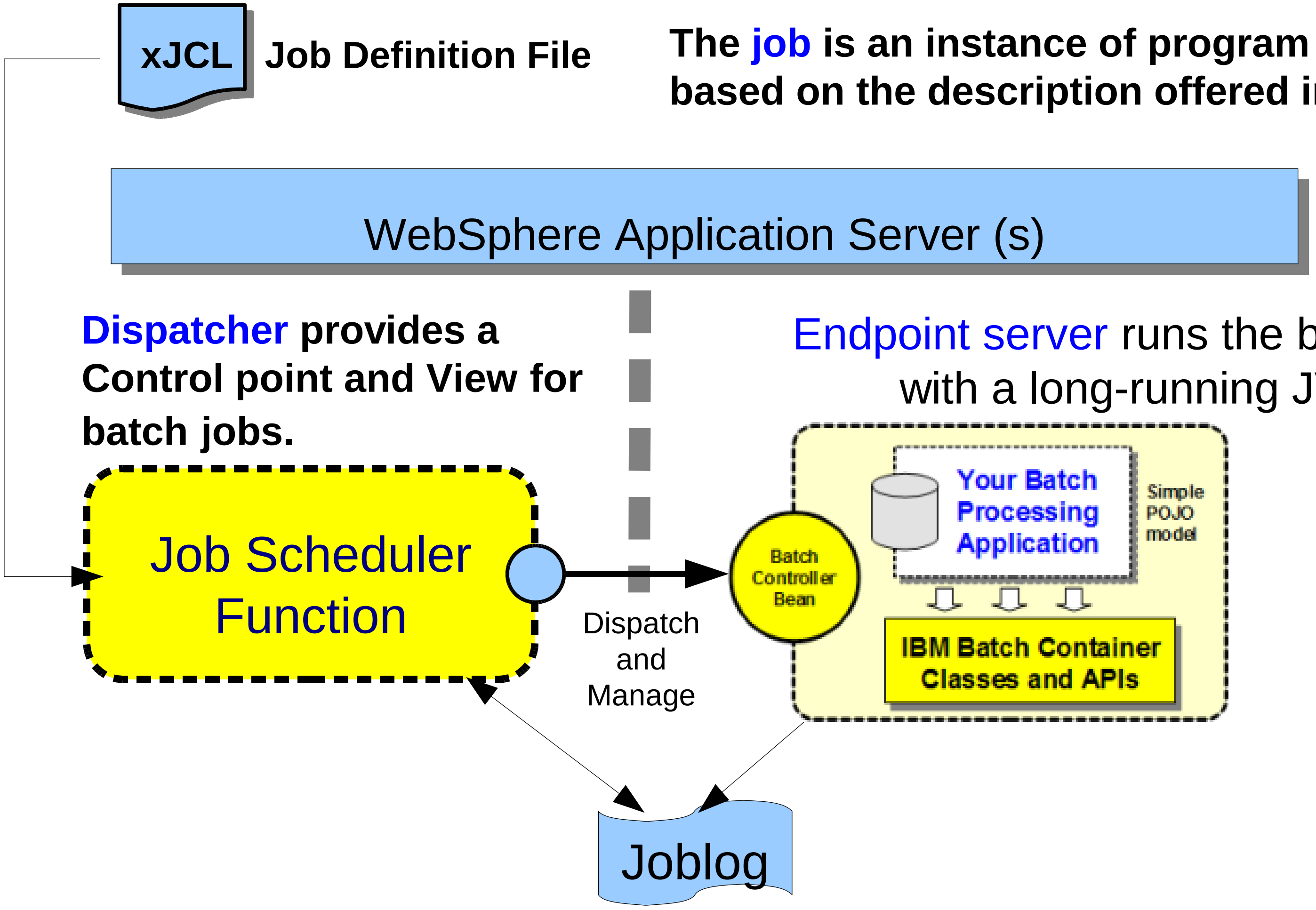
... similar business results.

Several Different Approaches...

- Standalone Java Program
  - JVM Launcher – JZOS
    - *WebSphere Java Batch Container*

Reviewing Compute Grid Servers . .

Consists of a Dispatcher & Endpoint server



The **job** is an instance of program invocation based on the description offered in the xJCL file.

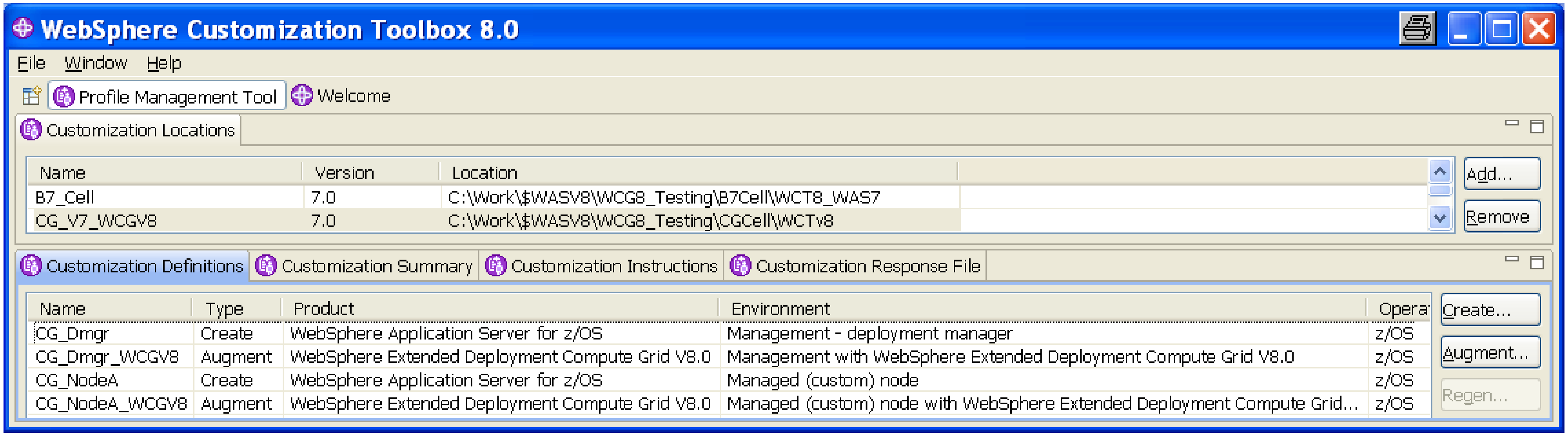
Dispatcher provides a Control point and View for batch jobs.

Endpoint server runs the batch jobs with a long-running JVM.

The Dispatcher & Endpoint may be in the Same server, Separate servers, or Clustered across many LPARs. Your choice. ☺

Configuration Improvements

- ***WAS Version 8***
  - Includes Modern Batch Feature Pack
  - Installation Manager support (Required)
  - WebSphere Customization Toolbox (WCT) V8 supports WCG Augmentation
- ***Use WCT V8 to create and augment a WAS V7 or V8 Cell with WCG V8***
  - *Deployment Manager & Empty Node*
  - *Augment with Compute Grid*



Name	Version	Location
B7_Cell	7.0	C:\Work\$WASV8\WCG8_Testing\B7Cell\WCT8_WAS7
CG_V7_WCGV8	7.0	C:\Work\$WASV8\WCG8_Testing\CGCell\WCTv8

Name	Type	Product	Environment	Opera
CG_Dmgr	Create	WebSphere Application Server for z/OS	Management - deployment manager	z/OS
CG_Dmgr_WCGV8	Augment	WebSphere Extended Deployment Compute Grid V8.0	Management with WebSphere Extended Deployment Compute Grid V8.0	z/OS
CG_NodeA	Create	WebSphere Application Server for z/OS	Managed (custom) node	z/OS
CG_NodeA_WCGV8	Augment	WebSphere Extended Deployment Compute Grid V8.0	Managed (custom) node with WebSphere Extended Deployment Compute Grid...	z/OS

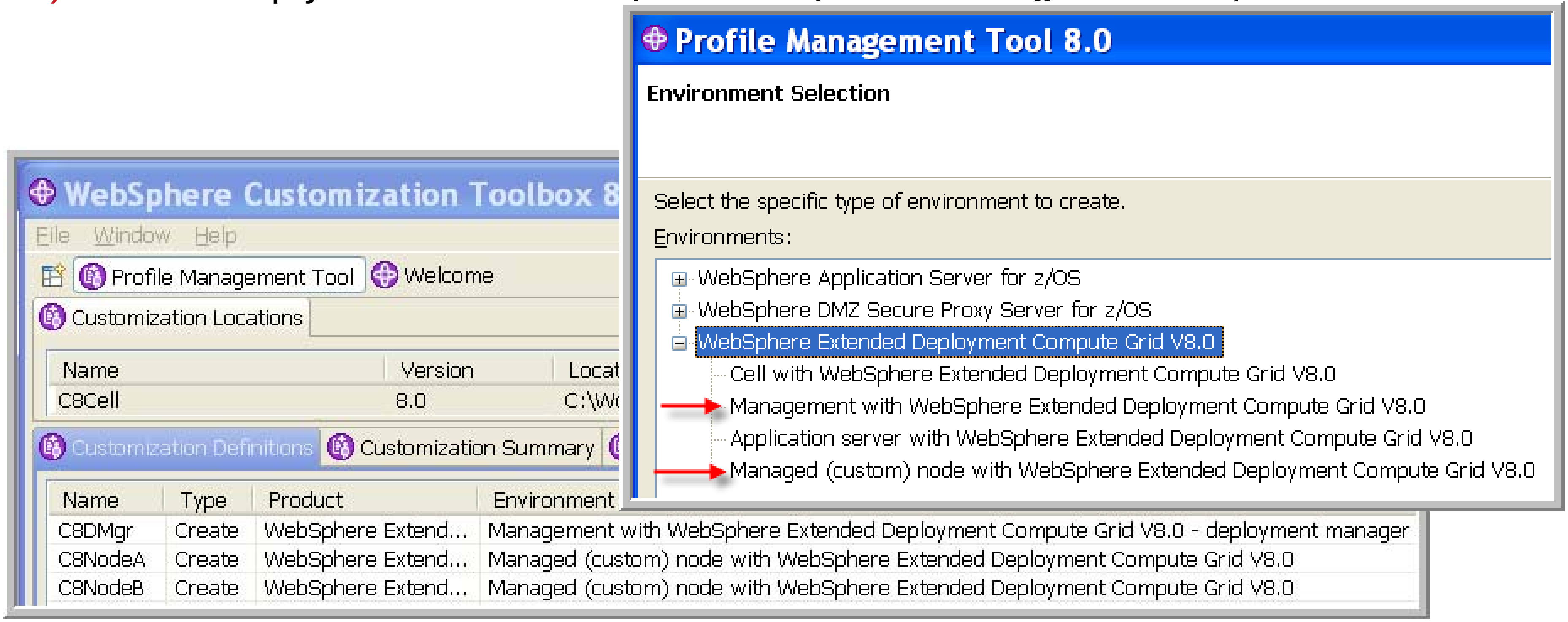
- ***Pre-configured in Compute Grid Version 8***
  - **Parallel Job Manager**
  - **PGCProxy** (used by **CICS CN11 SupportPac**)
  - **COBOL Container** and JAR files

Configuring WebSphere Compute Grid V8 on z/OS

ND Cell configuration Simplified with WCT Version 8:

1) Create a WebSphere V8 Deployment Manager Augmented with Compute Grid

2) Create Empty Nodes with Compute Grid (includes Augmentation)



Name	Version	Locat
C8Cell	8.0	C:\Wc

Name	Type	Product	Environment
C8DMgr	Create	WebSphere Extend...	Management with WebSphere Extended Deployment Compute Grid V8.0 - deployment manager
C8NodeA	Create	WebSphere Extend...	Managed (custom) node with WebSphere Extended Deployment Compute Grid V8.0
C8NodeB	Create	WebSphere Extend...	Managed (custom) node with WebSphere Extended Deployment Compute Grid V8.0

Profile Management Tool 8.0

Environment Selection

Select the specific type of environment to create.

Environments:

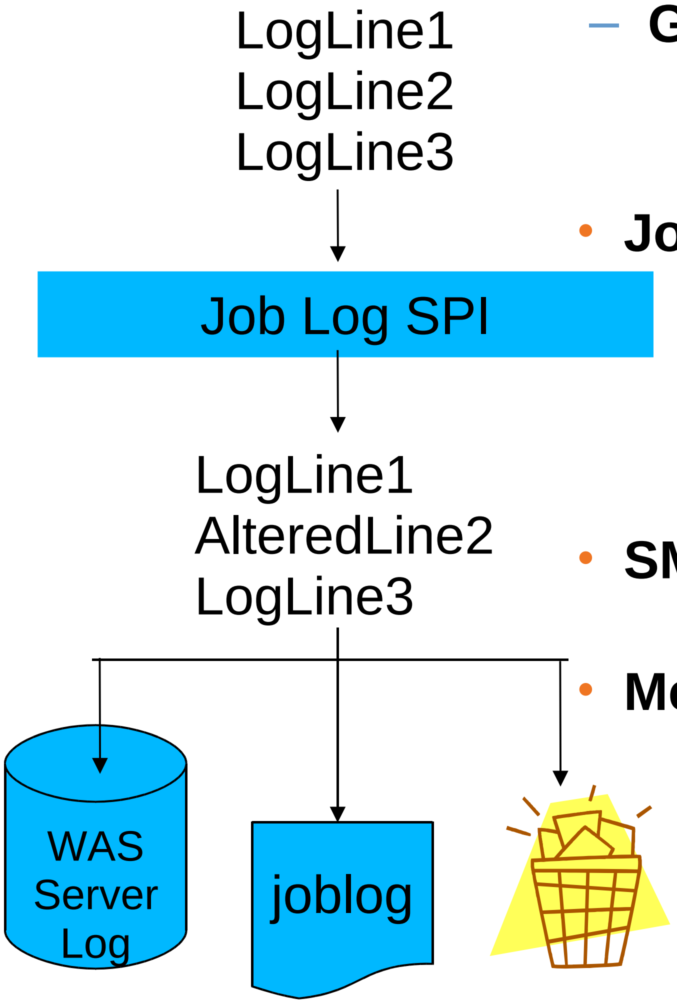
- WebSphere Application Server for z/OS
- WebSphere DMZ Secure Proxy Server for z/OS
- WebSphere Extended Deployment Compute Grid V8.0
  - Cell with WebSphere Extended Deployment Compute Grid V8.0
  - Management with WebSphere Extended Deployment Compute Grid V8.0
  - Application server with WebSphere Extended Deployment Compute Grid V8.0
  - Managed (custom) node with WebSphere Extended Deployment Compute Grid V8.0

Configure the Compute grid Dispatcher & Endpoint Servers:

3) Create Database & Data sources (DBA & ISC)

4) Configure Job Scheduler & Endpoint Server (ISC)

Operational Improvements



LogLine1
LogLine2
LogLine3

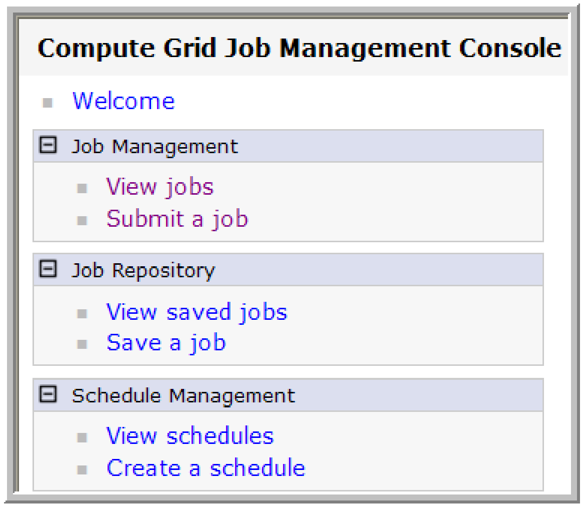
Job Log SPI

LogLine1
AlteredLine2
LogLine3

- **Group Level Security**
  - Control access to Jobs based on Group Membership (including the JMC)
- **JobLog SPI** (System Pgming Interface) **Controls**
  - Destination: Joblog or WAS Server log, or Both, or Neither (suppress)
  - Content: SPI can modify any job log line.
- **SMF Type 120 Subtype 9**
- **Memory Overload Protection**
  - Protects against over-scheduling jobs to an Endpoint, & Java OutOfMemory
  - Batch Container monitors job memory demand against available JVM heap
  - Automatic real time job memory estimation with declarative xJCL override

The Job Management Console

Browser-based view into the batch environment for Monitoring & Control:

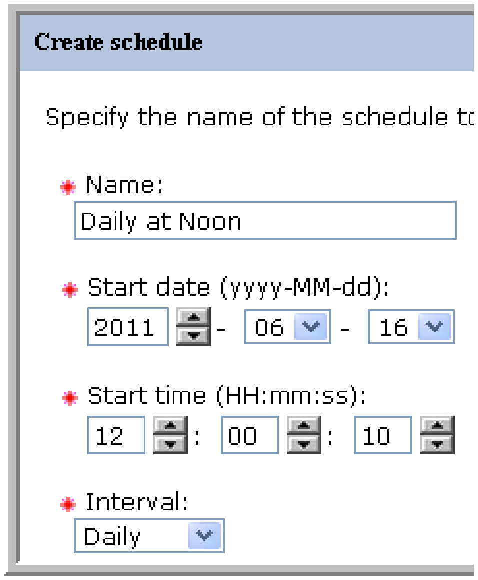


Compute Grid Job Management Console

- Welcome
- ⊟ Job Management
  - View jobs
  - Submit a job
- ⊟ Job Repository
  - View saved jobs
  - Save a job
- ⊟ Schedule Management
  - View schedules
  - Create a schedule

Browser

A web interface allows very simple access.



Create schedule

Specify the name of the schedule t

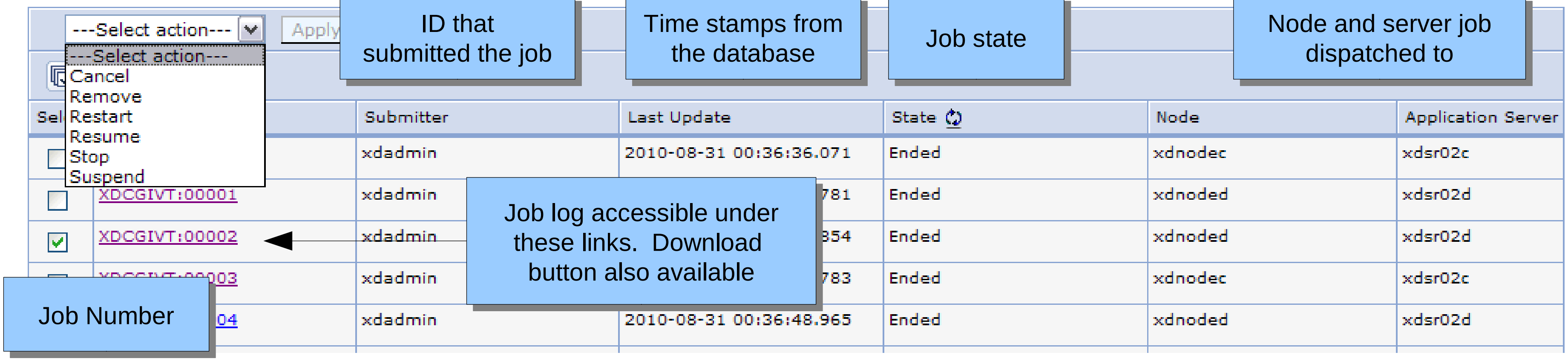
* Name:
Daily at Noon

* Start date (yyyy-MM-dd):
2011 - 06 - 16

* Start time (HH:mm:ss):
12 : 00 : 10

* Interval:
Daily

Actions against select jobs



---Select action---
Apply
---Select action---
Cancel
Remove
Restart
Resume
Stop
Suspend
ID that submitted the job
Time stamps from the database
Job state
Node and server job dispatched to
Job log accessible under these links. Download button also available
Job Number

Sel		Submitter	Last Update	State	Node	Application Server
□		xdadmin	2010-08-31 00:36:36.071	Ended	xdnodec	xdsr02c
□	XDCGIVT:00001	xdadmin	781	Ended	xdnoded	xdsr02d
☑	XDCGIVT:00002	xdadmin	354	Ended	xdnoded	xdsr02d
□	XDCGIVT:00003	xdadmin	783	Ended	xdnodec	xdsr02c
	04	xdadmin	2010-08-31 00:36:48.965	Ended	xdnoded	xdsr02d

Command Line, Web Services, IIOP and JMX interfaces as well

The Job Scheduler Interfaces

The previous chart tended to focus on the web interface, which is certainly the easiest to use. But others are present and offer great value:

Browser

A web interface allows very simple access.

Command Line

Automation through shell script programming.

Web Service

Expose without requiring access to the JMC

RMI

Expose to EJB clients

JMX

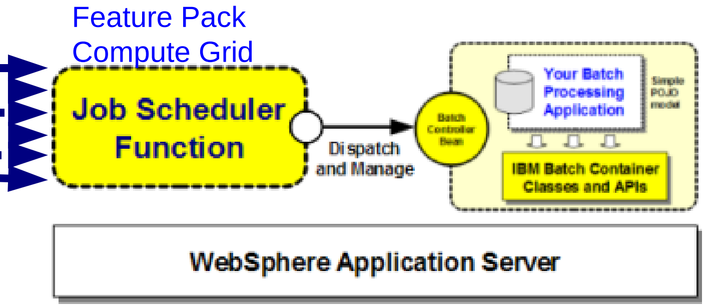
Expose to Java JMX client

Tivoli. software

Or others

MDB (Compute Grid only)

Used to integrate with enterprise schedulers

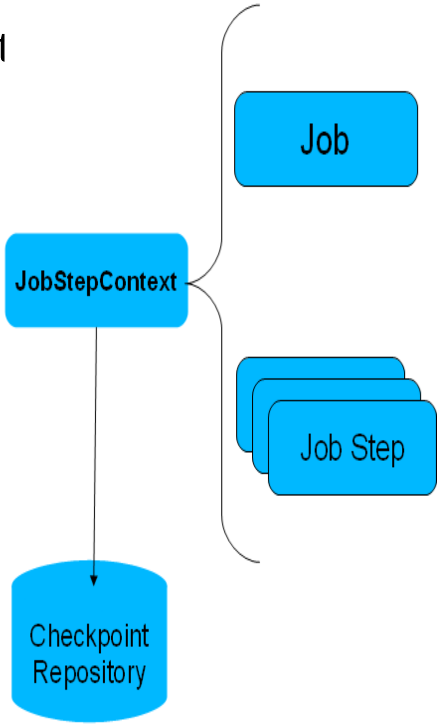


A wide variety of access methods

Blend to meet your business needs

Programming Model Enhancements in WCG V8

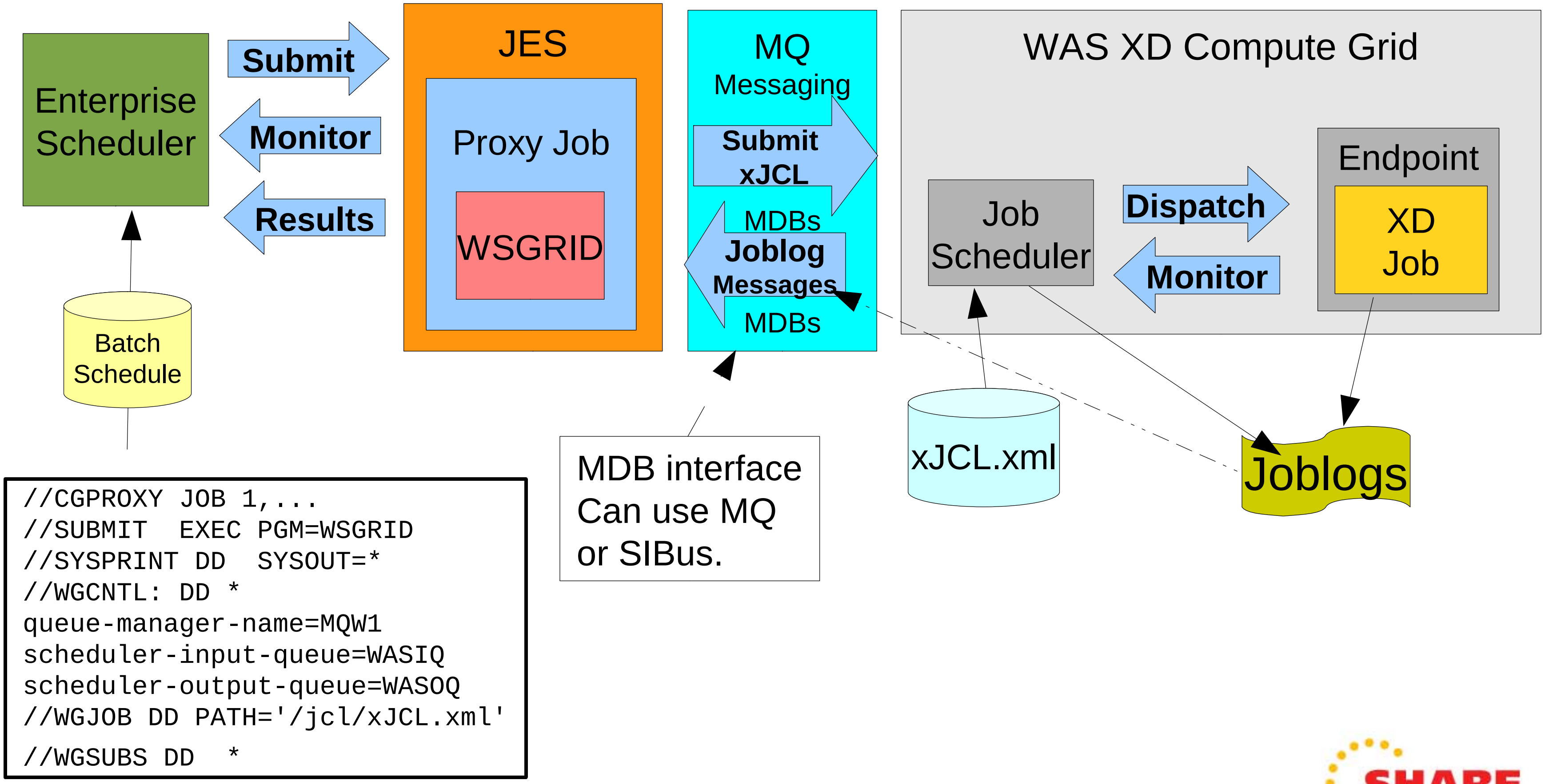
- **Persistent JobStepContext object**
  - Exists for life of job; Step-specific context reset at each job step.
  - New persistent user data object stored across checkpoint/restart
- **Job and Step Listener**
  - Notification of Job/Step Start/End thru JobStepContext object.
- **Configurable Transaction Mode**
  - Select job step transaction mode: Local or Global
- **Record Processing Policy**
  - Skip bad records, Retry, or Stop Job Processing controls
- **Batch Data Stream Timeout** – Configurable by Job Step
  - Some BatchDataStreams need Short timeouts, others Long.
- **Record Metrics**
  - Skipped record count, Retry count,
  - Records/Second, Processing time
  - Written to Joblog
  - Available to batch application thru JobStepContext object.
- **OSGi Batch Applications**
  - Deploy batch applications as OSGi bundles
- **COBOL Container Support**



Integrate with JES Schedulers

Traditional batch jobs or Schedulers can extend their reach to Compute Grid with the “WSGRID” utility running as a “Proxy” for an XD batch job:

- Proxy job stays active & receives joblog messages in SYSOUT file until XD job ends.
- WSGRID utility notifies the Scheduler of XD job Return Code.

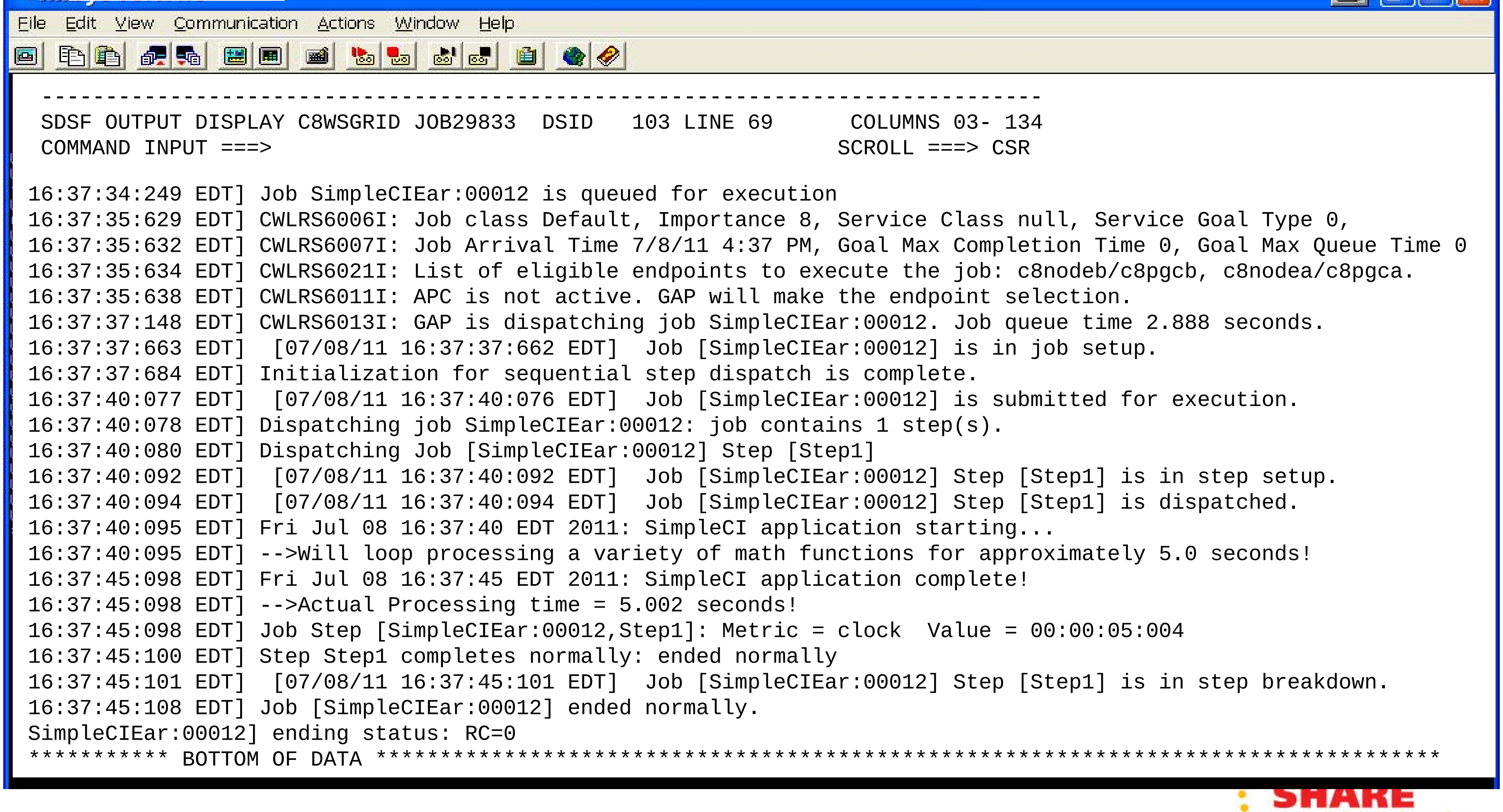


MDB interface Can use MQ or SIBus.

```
//CGPROXY JOB 1,...
//SUBMIT  EXEC PGM=WSGRID
//SYSPRINT DD  SYSOUT=*
//WGCNTL: DD *
queue-manager-name=MQW1
scheduler-input-queue=WASIQ
scheduler-output-queue=WASOQ
//WGJOB DD PATH='/jcl/xJCL.xml'
//WGSUBS DD  *
```

Joblog & Return Code returned to JES Poxy job

Use SDSF or other products to view results:

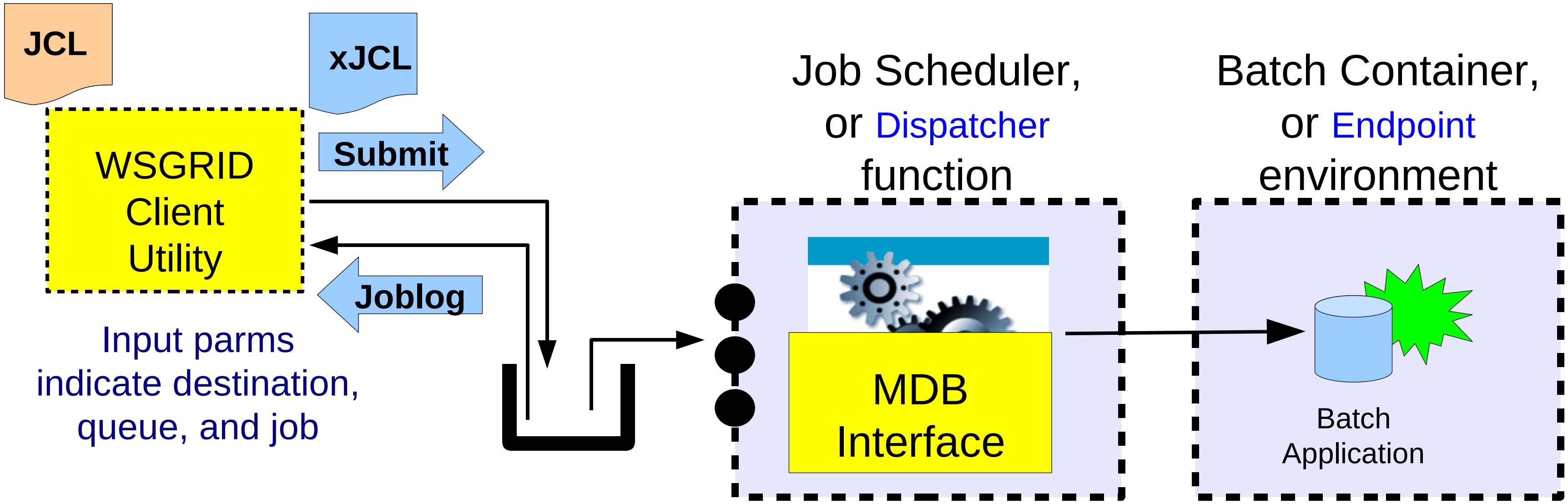


```
 ------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY C8WSGRID JOB29833  DSID   103 LINE 69      COLUMNS 03- 134
 COMMAND INPUT ===>                                            SCROLL ===> CSR

16:37:34:249 EDT] Job SimpleCIEar:00012 is queued for execution
16:37:35:629 EDT] CWLRS6006I: Job class Default, Importance 8, Service Class null, Service Goal Type 0,
16:37:35:632 EDT] CWLRS6007I: Job Arrival Time 7/8/11 4:37 PM, Goal Max Completion Time 0, Goal Max Queue Time 0
16:37:35:634 EDT] CWLRS6021I: List of eligible endpoints to execute the job: c8nodeb/c8pgcb, c8nodea/c8pgca.
16:37:35:638 EDT] CWLRS6011I: APC is not active. GAP will make the endpoint selection.
16:37:37:148 EDT] CWLRS6013I: GAP is dispatching job SimpleCIEar:00012. Job queue time 2.888 seconds.
16:37:37:663 EDT]  [07/08/11 16:37:37:662 EDT]  Job [SimpleCIEar:00012] is in job setup.
16:37:37:684 EDT] Initialization for sequential step dispatch is complete.
16:37:40:077 EDT]  [07/08/11 16:37:40:076 EDT]  Job [SimpleCIEar:00012] is submitted for execution.
16:37:40:078 EDT] Dispatching job SimpleCIEar:00012: job contains 1 step(s).
16:37:40:080 EDT] Dispatching Job [SimpleCIEar:00012] Step [Step1]
16:37:40:092 EDT]  [07/08/11 16:37:40:092 EDT]  Job [SimpleCIEar:00012] Step [Step1] is in step setup.
16:37:40:094 EDT]  [07/08/11 16:37:40:094 EDT]  Job [SimpleCIEar:00012] Step [Step1] is dispatched.
16:37:40:095 EDT] Fri Jul 08 16:37:40 EDT 2011: SimpleCI application starting...
16:37:40:095 EDT] -->Will loop processing a variety of math functions for approximately 5.0 seconds!
16:37:45:098 EDT] Fri Jul 08 16:37:45 EDT 2011: SimpleCI application complete!
16:37:45:098 EDT] -->Actual Processing time = 5.002 seconds!
16:37:45:098 EDT] Job Step [SimpleCIEar:00012,Step1]: Metric = clock  Value = 00:00:05:004
16:37:45:100 EDT] Step Step1 completes normally: ended normally
16:37:45:101 EDT]  [07/08/11 16:37:45:101 EDT]  Job [SimpleCIEar:00012] Step [Step1] is in step breakdown.
16:37:45:108 EDT] Job [SimpleCIEar:00012] ended normally.
SimpleCIEar:00012] ending status: RC=0
*********** BOTTOM OF DATA ***********************************************************************************
```

How to Integrate with JES Batch Jobs

MDB interface to the dispatcher using MQ or imbedded messaging



JCL

xJCL

WSGRID Client Utility

Submit

Joblog

Input parms indicate destination, queue, and job

Job Scheduler, or Dispatcher function

Batch Container, or Endpoint environment

MDB Interface

Batch Application

Configuration simplified with Compute Grid V.8 ....

1. Define WebSphere MQ input & output queues.
2. Configure runtime variables to access MQ libraries.
3. **`installWSGridMQ.py`** script
   - Sets up JMS Connection Factories, Queues & ListenerPort
   - Installs WSGRID system application.
4. Create WSGRID load module in an executable library.

Simple!

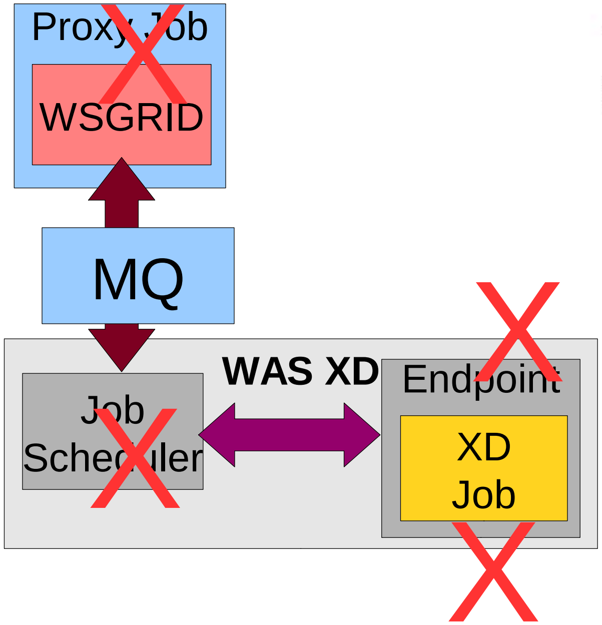
Failure Scenarios

What happens if . . .

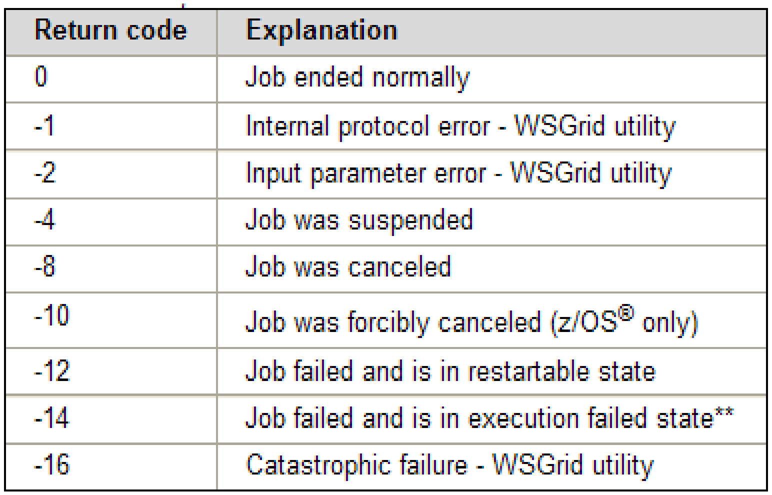
- **Proxy job fails (or canceled)**
  - XD job is canceled.
- **XD job fails (or canceled)**
  - Proxy job fails
- **Endpoint server fails**
  - XD job fails
  - Proxy job will timeout
- **Scheduler server fails**
  - Proxy job fails
  - XD job fails
- **MQ Fails**
  - Proxy job fails,
  - XD job is canceled.

In all cases . . .

- XD job may be restarted.



WSGRID Return Codes

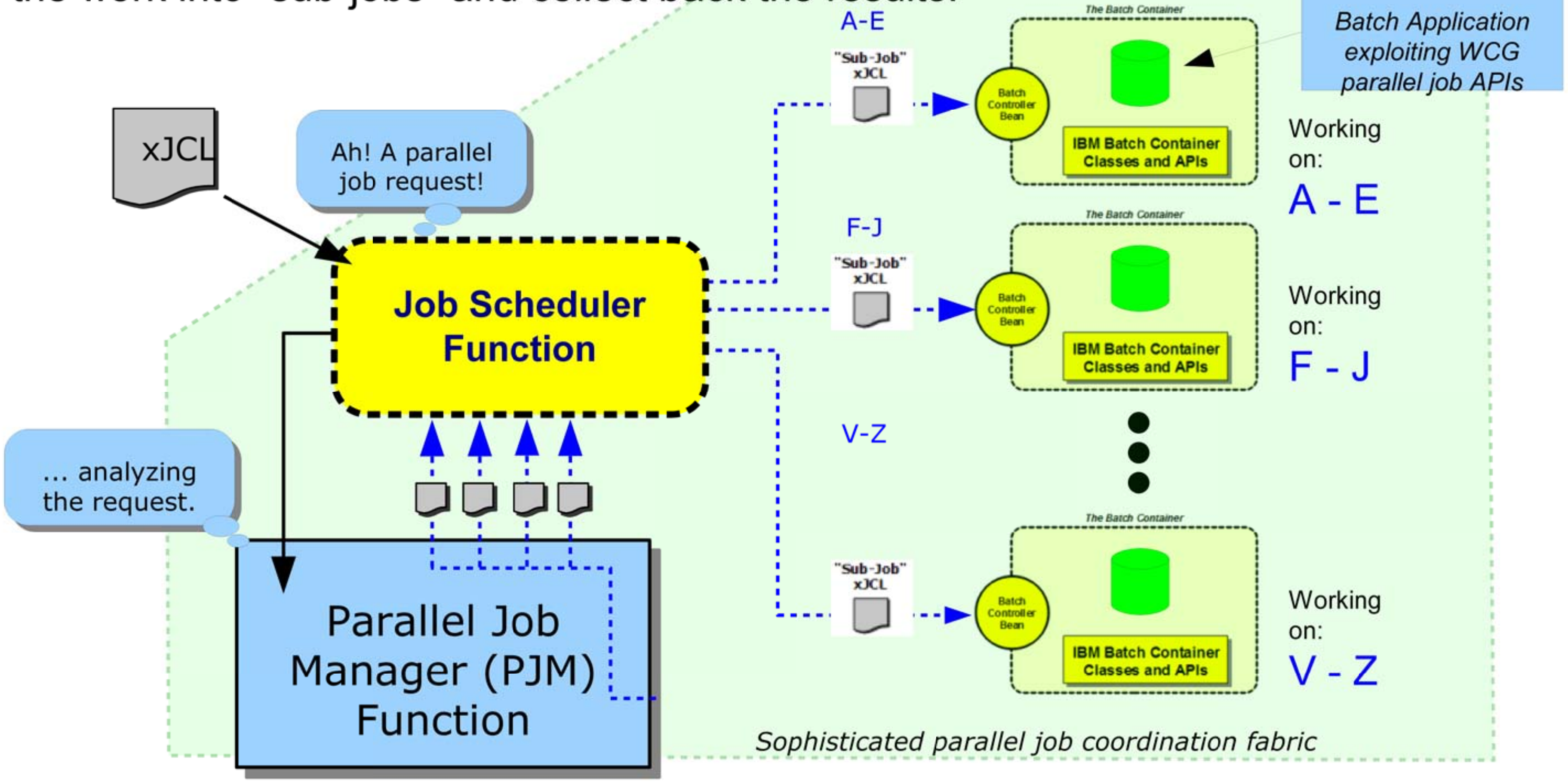


Return code	Explanation
0	Job ended normally
-1	Internal protocol error - WSGrid utility
-2	Input parameter error - WSGrid utility
-4	Job was suspended
-8	Job was canceled
-10	Job was forcibly canceled (z/OS® only)
-12	Job failed and is in restartable state
-14	Job failed and is in execution failed state**
-16	Catastrophic failure - WSGrid utility

Parallel Job Manager (PJM)

Batch processing often lends itself to running the work in parallel.

- WebSphere Compute Grid facilitates this with function to cut up the work into “sub-jobs” and collect back the results:



xJCL
Ah! A parallel job request!
Job Scheduler Function
... analyzing the request.
Parallel Job Manager (PJM) Function
A-E
"Sub-Job" xJCL
F-J
"Sub-Job" xJCL
V-Z
"Sub-Job" xJCL
The Batch Container
Batch Controller Bean
IBM Batch Container Classes and APIs
The Batch Container
Batch Controller Bean
IBM Batch Container Classes and APIs
The Batch Container
Batch Controller Bean
IBM Batch Container Classes and APIs
Batch Application exploiting WCG parallel job APIs
Working on: A - E
Working on: F - J
Working on: V - Z
Sophisticated parallel job coordination fabric

Configuring the Parallel Job Manager

How the PJM in V8 differs from previous versions

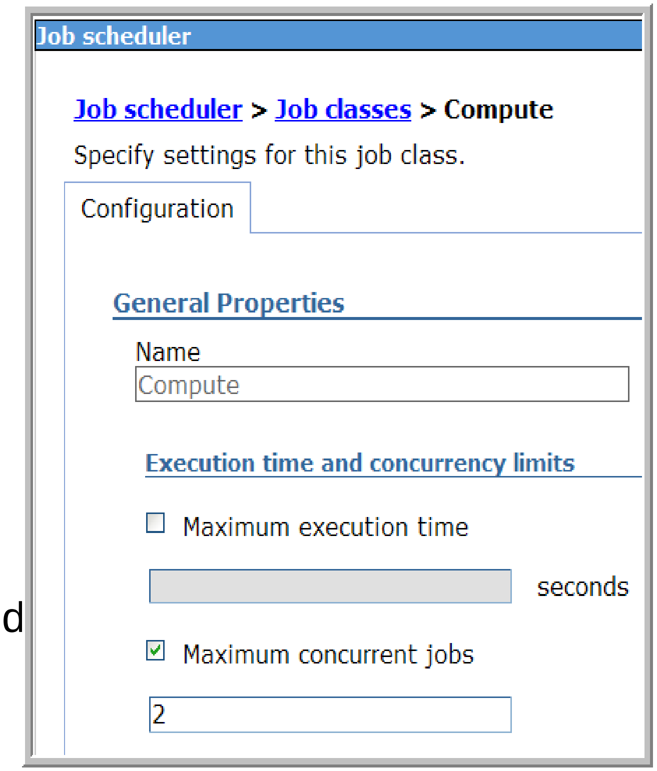
- Parallel job manager integrated into the batch container.
  - Not a separate system application as before.
  - No need to install and configure the PJM, or separate DB2 tables.
  - No shared library required - PJM APIs in batch utility JAR.
- The contents of the xd.spi.properties file are now part of the xJCL.
  - No **`xd.spi.properties`** file required.
- Only a single xJCL file is required.
  - Combines the top-level job xJCL with subordinate jobs.
- PJM applications built for CG V6 can run as is on WCGv8.
  - Migrate a WCGv6 PJM application to WCGv8:
  - Add the API implementation classes to the application EAR.
  - Reauthor xJCL as described in V8 InfoCenter.

"Job Class" Controls for Batch Jobs

- **Job Class** specified in xJCL:

```
<?xml version="1.0"
<job name="SimpleCI"
 class="Compute"
```

- **Job Class can be used to limit:**
  - **MaxExecutionTime** – Before Jobs are canceled
  - **MaxConcurrentJob** – # Allowed to run concurrently
  - **MaxClassSpace** – Joblogs exceeding this size are Purged
  - **MaxFileAge** – Joblogs older than this are Purged
  - **MaxJob** – Max. # of Jobs (Oldest are Purged)
  - **MaxJobAge** – Jobs older than this are Purged from Output Queue



Job scheduler

Job scheduler > Job classes > Compute

Specify settings for this job class.

Configuration

General Properties

Name: Compute

Execution time and concurrency limits

□ Maximum execution time

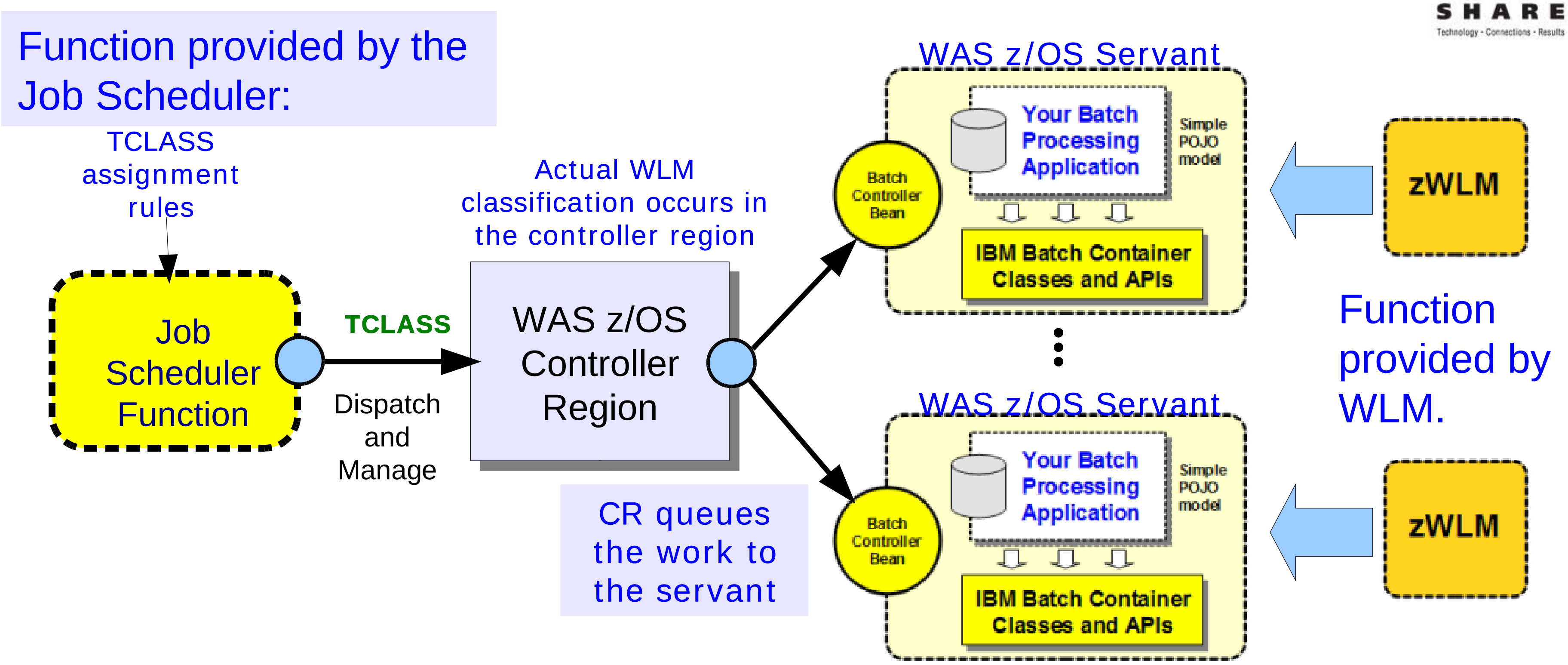
seconds

☑ Maximum concurrent jobs

2

- **Job Class** can also be used to assign a Transaction Class (next foil....)

Classifying Batch Jobs with Compute Grid


Function provided by the Job Scheduler:
TCLASS assignment rules
Job Scheduler Function
TCLASS
Dispatch and Manage
Actual WLM classification occurs in the controller region
WAS z/OS Controller Region
CR queues the work to the servant
WAS z/OS Servant
Batch Controller Bean
Your Batch Processing Application
Simple POJO model
IBM Batch Container Classes and APIs
WAS z/OS Servant
Batch Controller Bean
Your Batch Processing Application
Simple POJO model
IBM Batch Container Classes and APIs
zWLM
zWLM
Function provided by WLM.

Classify batch jobs according to your service goals.

- Unique to z/OS platform.

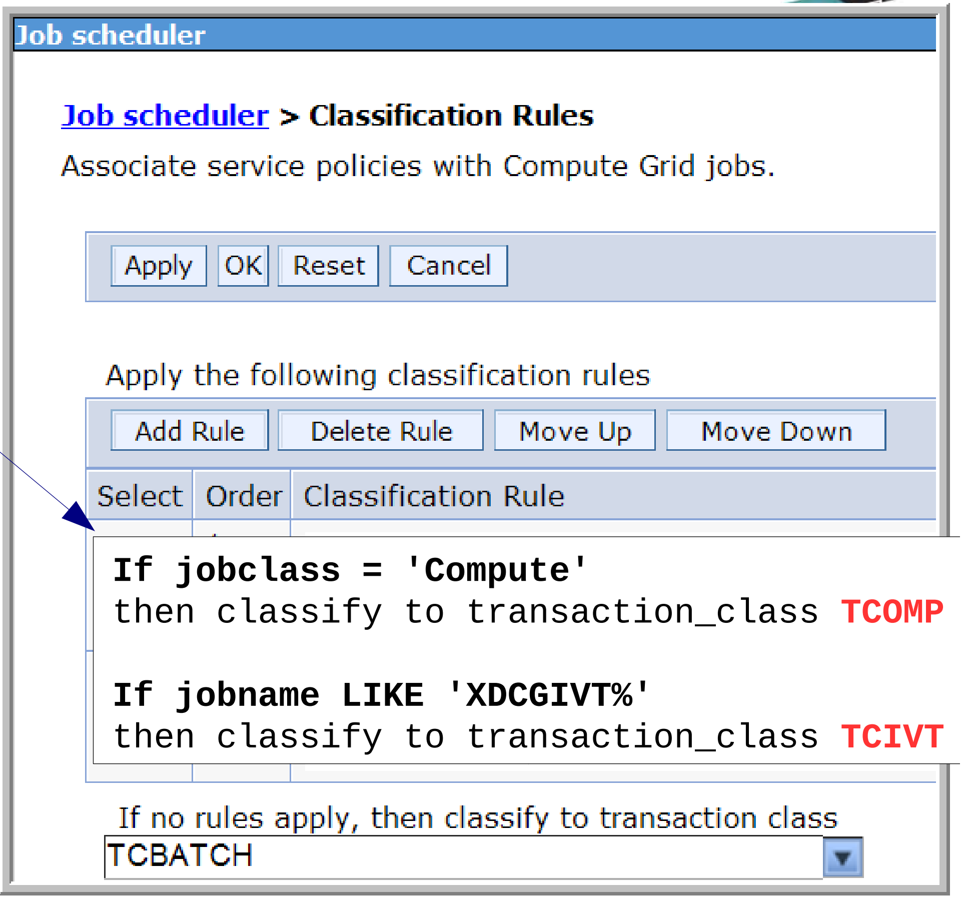
Classifying Batch Jobs:

ISC:Settings: Job Scheduler > Classification Rules

- **Transaction Class** assigned based on:
- Submitter Identity or Group
- **Job name** or **Job class**
- Application name or Application type
- Platform or Time
- Else, Default class = **TCBATCH**

WLM Classification Rules:

- CB Rules assign **Service Class** based on:
  - Generic Server (Cluster) name (CN)
  - **Transaction class (TC)**
    - assign by Job Scheduler > Classification Rules
-



Job scheduler > Classification Rules

Associate service policies with Compute Grid jobs.

Apply the following classification rules

```
If jobclass = 'Compute'
then classify to transaction_class TCOMP

If jobname LIKE 'XDCGIVT%'
then classify to transaction_class TCIVT
```

If no rules apply, then classify to transaction class

TCBATCH

```
   Workload Manager CB Classification Rules:
  ----Qualifier-------
-------Class--------
    Type     Name     Start    Service    Report
                       DEF:  CBCLASS    RWASDEF
 1  CN       Z8*      ___      CBFAST     Z8BAT
 2     TC       TCOMP    ___   CBSLOW     Z8BATCI
 2     TC       TCIVT    ___   CBCLASS    Z8XDIVT
 2     TC       TCBATCH  ___   CBCLASS    Z8BATCH
```

2 Classification Mechanisms for Batch Jobs:

(1) “Job Class” can be specified by a job.

Job Scheduler dispatches jobs to Endpoint Servers, based on:

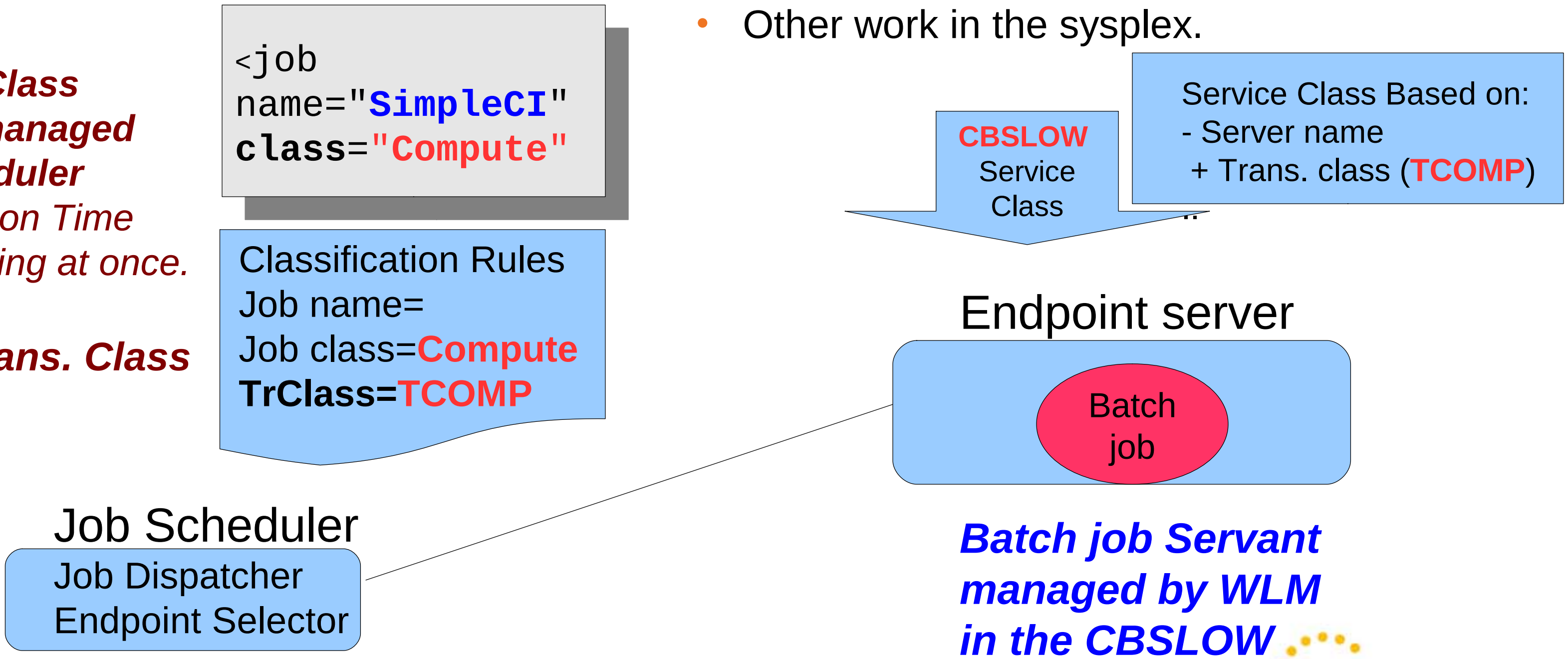
- Availability of Servers to accept new work.
- Number of jobs running within the maximum threshold for the “**Job Class**”
- “Maximum Execution Time” can also be assigned based on the **Job Class**.

(2) z/OS Workload Manager (WLM) dispatches CPU cycles & I/O to jobs according to:

- **Service class goals** assigned to prioritize the:
  - “Importance” and
  - Response Time objectives or “Velocity” goals.
- Service Class assigned based on Cluster Name & **Transaction Class** assigned by the Scheduler.
- Dynamically adjusted based on:
  - Availability of CPU (& other) resources
  - Other work in the sysplex.

Example:
Jobs in Job Class
“Compute” managed
by Job Scheduler
- Max. Execution Time
- Max. # Running at once.

Assigned Trans. Class



```
<job
name="SimpleCI"
class="Compute"
```

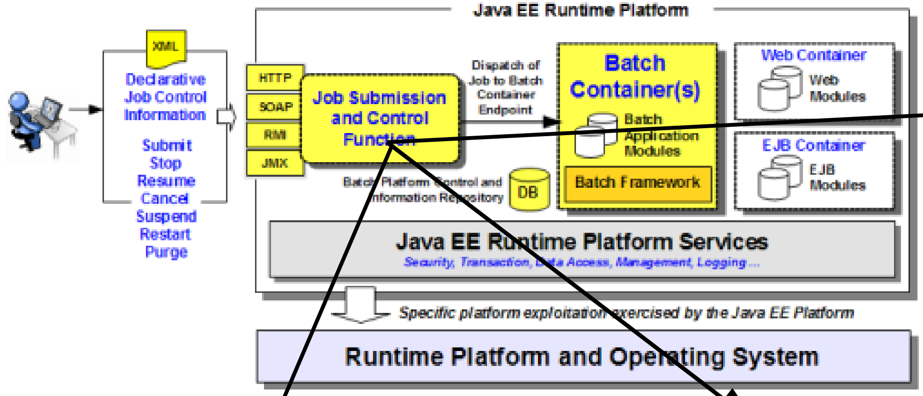
Classification Rules
Job name=
Job class=**Compute**
TrClass=TCOMP

Service Class Based on:
- Server name
+ Trans. class (**TCOMP**)

Batch job Servant managed by WLM in the CBSLOW service class

Compute Grid Job Usage Recording

Job Usage Accounting with SMF records and DB2 JOBUSAGE tables:


Java EE Runtime Platform
XML
Declarative Job Control Information
Submit
Stop
Resume
Cancel
Suspend
Restart
Purge
HTTP
SOAP
RMI
JMX
Job Submission and Control Function
Dispatch of Job to Batch Container Endpoint
Batch Container(s)
Batch Application Modules
Batch Framework
Batch Platform Control and Information Repository
DB
Web Container
Web Modules
EJB Container
EJB Modules
Java EE Runtime Platform Services
Security, Transaction, Data Access, Management, Logging ...
Specific platform exploitation exercised by the Java EE Platform
Runtime Platform and Operating System

DB2 JOBUSAGE records:

- Job identifier
- Job submitter
- Final Job state
- Server
- Node
- Accounting info.
- Job start time
- Last update time
- General CPU usage

SMF 120.20 record contents:

- Job identifier
- Job submitter
- Final Job state
- Server
- Node
- Accounting information
- Job start time
- Last update time
- General CPU usage
- zAAP or zIIP CPU use

SMF 120.9 UserData Section (WAS V.8 only)

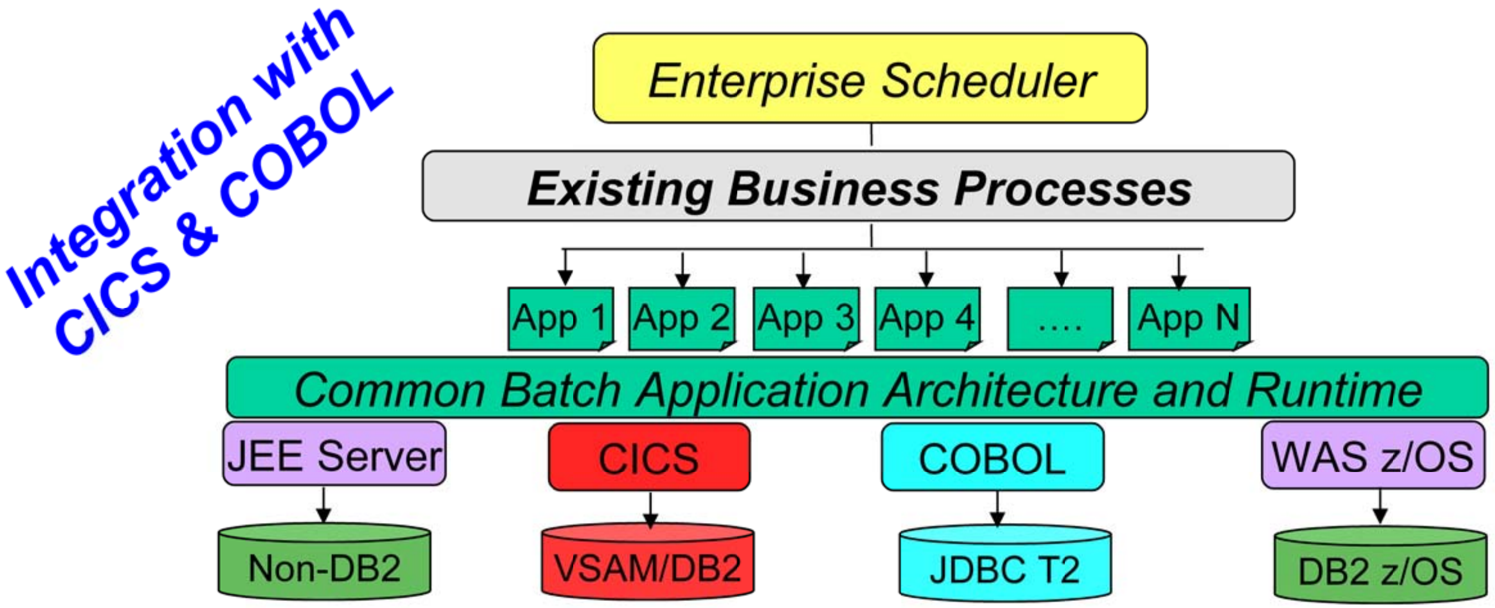
- Job identifier
- Job submitter
- Accounting info

Record Metrics added to Joblogs....

- Skipped record count, Retry count,
- Records/Second, Processing time
- Written to Joblog & Available in JobStepContext object.

Generate reports & determine usage for Charge-back & Capacity Planning.

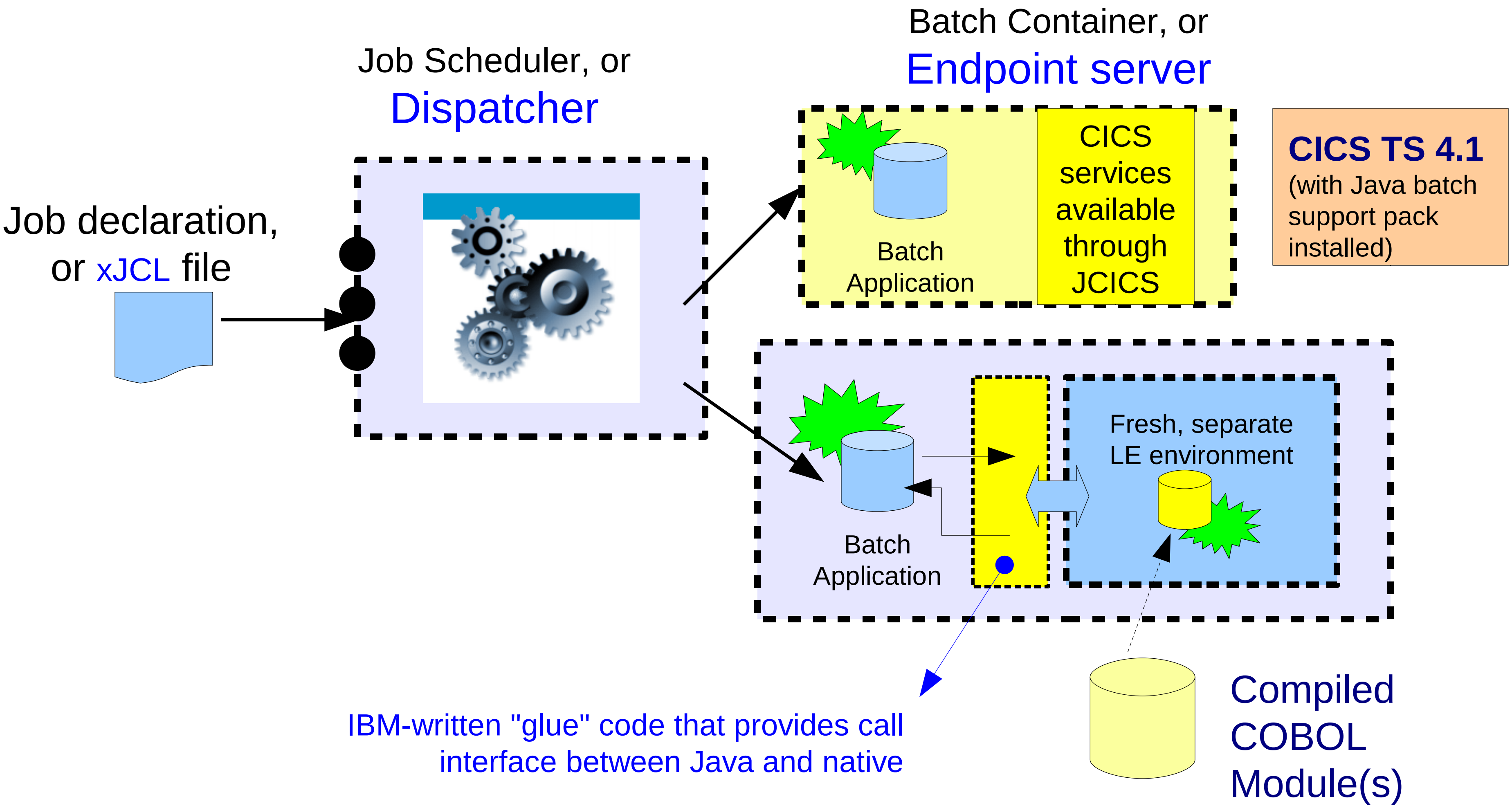
The Batch Vision


Integration with CICS & COBOL
Enterprise Scheduler
Existing Business Processes
App 1
App 2
App 3
App 4
….
App N
Common Batch Application Architecture and Runtime
JEE Server
CICS
COBOL
WAS z/OS
Non-DB2
VSAM/DB2
JDBC T2
DB2 z/OS

1. Batch Containers should run ***everywhere***
2. ***Portable Batch applications*** across platforms and J2EE vendors
3. Location of the data dictates the placement of the batch application
4. Centrally managed by your enterprise scheduler
5. Integrating with existing: Disaster Recovery, Auditing, Logging, Archiving

Java Batch + CICS, or COBOL? Yes ...

New Batch container for CICS and COBOL interoperability function allows Java batch programs to call CICS and COBOL directly ...

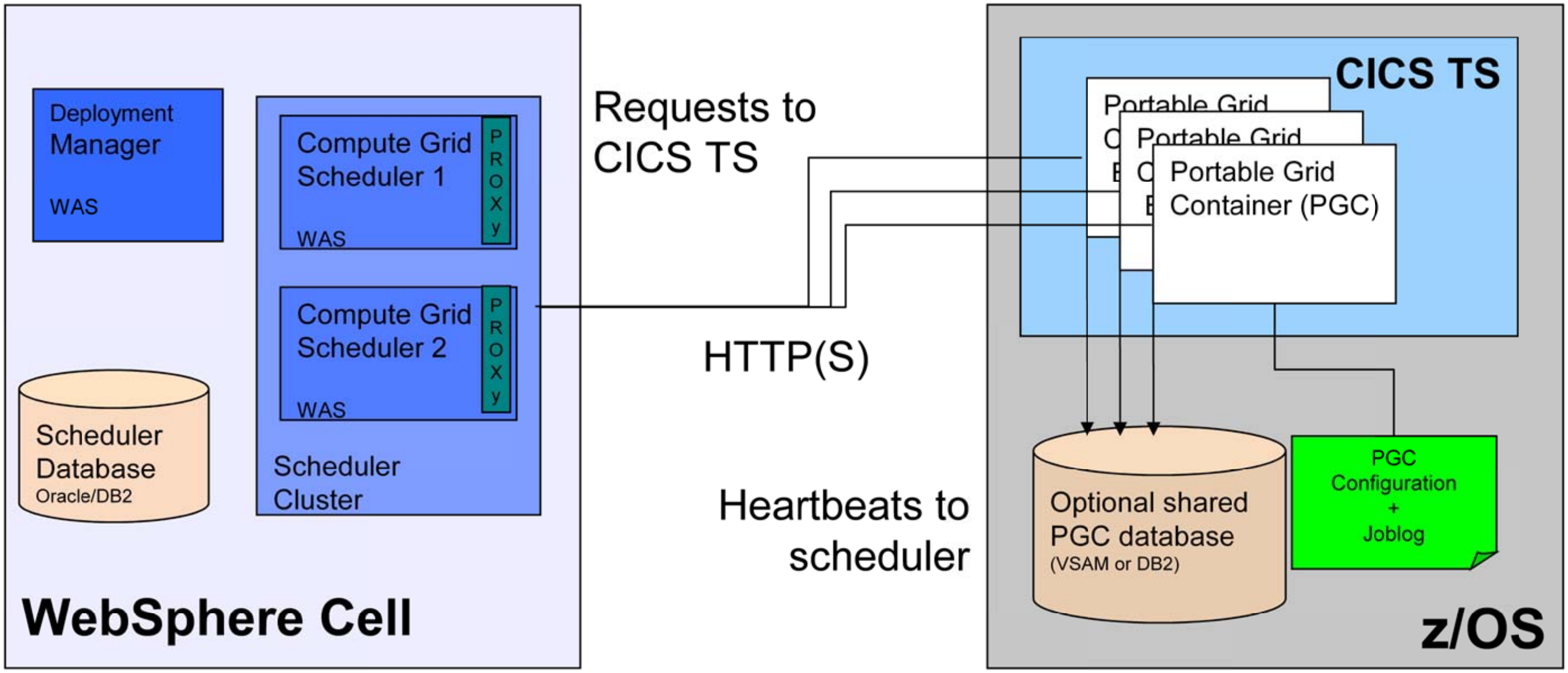


Re-use CICS & COBOL assets with a Java batch job.

Compute Grid & CICS w/ SupportPac CN11

Batch job in Compute Grid sends HTTP request to CICS to start the transaction program.

CICS matches it against the path in CN11URIresource to handle the request.



Deployment Manager

WAS

Compute Grid Scheduler 1

WAS

PROXY

Compute Grid Scheduler 2

WAS

PROXY

Scheduler Cluster

Scheduler Database

Oracle/DB2

WebSphere Cell

Requests to CICS TS

HTTP(S)

Heartbeats to scheduler

CICS TS

Portable Grid

Portable Grid

Portable Grid Container (PGC)

Optional shared PGC database

(VSAM or DB2)

PGC Configuration + Joblog

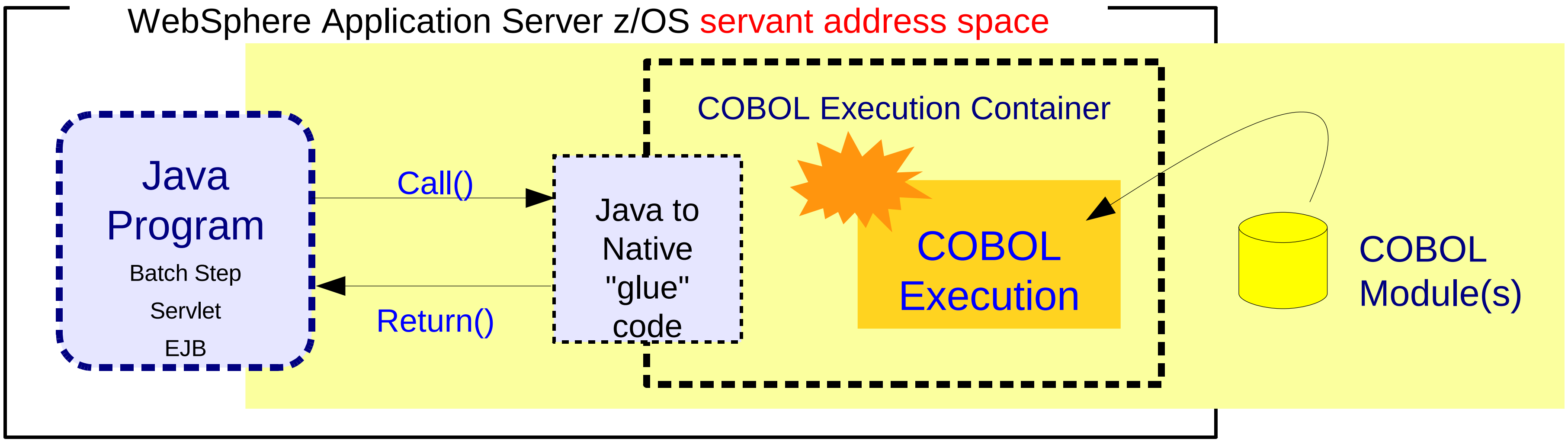
z/OS

Integration with CICS “CN11” SupportPac

- **SupportPac enables Compute Grid to schedule jobsteps in CICS**
  - Provides Java interfaces to process input and output files in VSAM or DB2
  - Increased availability for CICS during batch processing, Automatic checkpoints, Restarts
  - CICS TS V. 4.1 required
- **Compute Grid provides**
  - General job dispatching, management, execute control, monitoring
  - Higher throughput: Process jobs in parallel across multiple CICS regions
  - Locking of data: Updates are synchronised at Checkpoints for I/O resources
  - Failure/Recovery scenarios
    - If batch jobstep fails, Rollback updates, Restore last checkpoint and Retry jobstep.
- **Configuration:**
  - Configure supporting CICS, DB2 and VSAM resources (Samplejobs provided)
  - Customize **`endpoint-config.xml`** & **`CN11PROF`** describing CICS to WCG
  - Initialize the CICS-to-Compute Grid Connection
    - Run the **“`CN11`”** CICS transaction to register the SupportPac samples with WCG.
  - **CN11SampleJCL.xml** sample job runs a CICS transaction that updates a VSAM file.

The Compute Grid ***COBOL Container***

"COBOL Container" provides the JNI services:


WebSphere Application Server z/OS servant address space
Java Program
Batch Step
Servlet
EJB
Call()
Return()
Java to Native "glue" code
COBOL Execution Container
COBOL Execution
COBOL Module(s)

Important Points:

- Create and destroy COBOL container multiple times in the servant address space
- COBOL container's LE enclave is separate from the address space's LE enclave (clean environment)
- JDBC T2 connection can be shared between Java and the COBOL program (maintaining transactional context using RRSAF)

Essentials of the New COBOL Support

Included as part of WebSphere Compute Grid V8.
- Compatible with WAS z/OS V.7 or V.8

The server must run in 31-bit mode since COBOL programs are 31-bit. And run with a workload profile of ISOLATE (to insure OUTDD back from COBOL works with DISPLAY)

Compiler and Link Edit Options:

- Must be a Dynamic Link Library (DLL)
- Must specify OUTDD(WCGILOUT) so output may go back to Java batch

Major pieces of this:

- Runtime support (a few JAR and native files) shipped with WCG V.8
- Development tooling support (JAR files)
- Call Stub Generator utility
- Your Java code, that calls the COBOL module.

Basic components ...

Enabling the function...

Provide WAS server access to COBOL DLLs -- `LIBPATH & STEPLIB`

Create the container

```
 ILContainer ilc = ILContainerFactory.getFactory().create();
```

Name the COBOL module and procedure within the module, and where you pass parameters.

Create the procedure

```
ILProcedure ilp = ILProcedureFactory.getFactory().create(...);
```

Call Stub Generator makes doing this much easier.

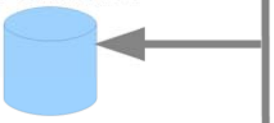
Invoke the procedure

```
 x = ilc.invokeProcedure(ilp)
```

Compile the COBOL modules and put DLL in USS or PDSE

Compiled DLL

In USS or in PDSE



```
IDENTIFICATION DIVISION.
PROGRAM-ID. ADDER RECURSIVE.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
DATA DIVISION.
LINKAGE SECTION.
01 adder-input.
  05 int-a             PIC 9(8).
  05 int-b             PIC 9(8).
01 adder-return.
  05 int-c             PIC 9(8).
 :
```

The "procedure" is the value provided for PROGRAM-ID.

The DLL module name is also part of the invocation procedure

Call Stub Generator ...

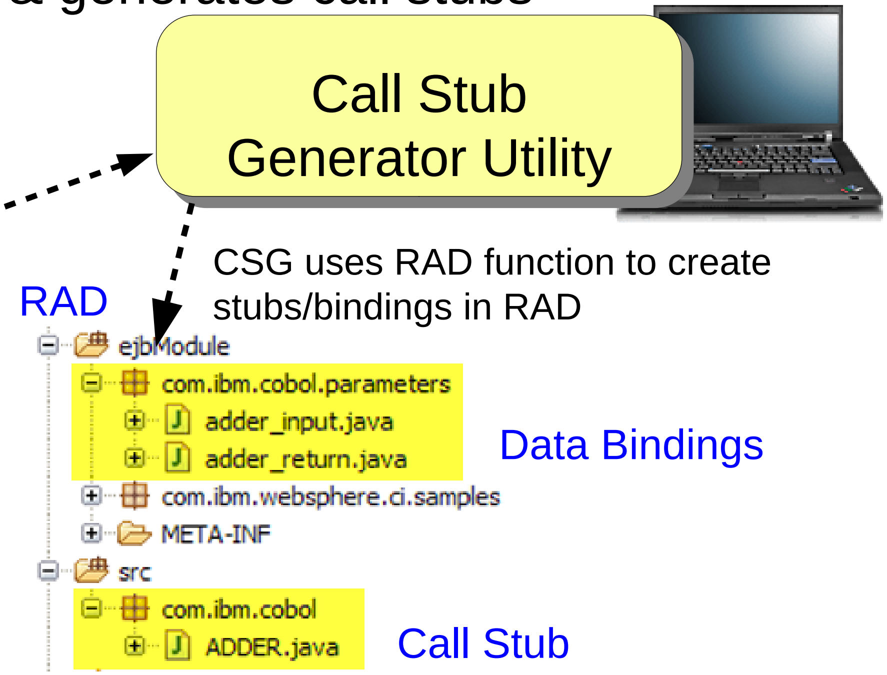
The Call Stub Generator (CSG)

Java utility inspects the COBOL source & generates call stubs & data bindings:

COBOL Source

```
  :
LINKAGE SECTION.
01 adder-input.
  05 int-a                   PIC 9(8).
  05 int-b                   PIC 9(8).
01 adder-return.
  05 int-c                   PIC 9(8).

PROCEDURE DIVISION USING adder-input
                         RETURNING adder-return.
  :
```



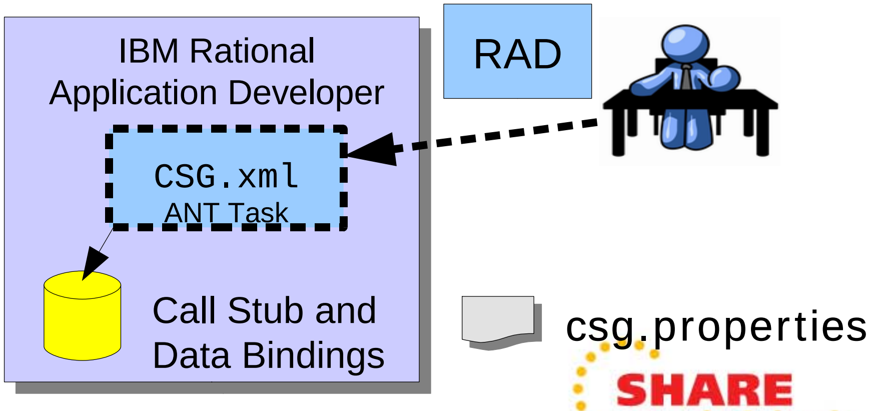
CSG uses RAD function to create stubs/bindings in RAD

Two ways to invoke the CSG:

Command line & ANT task in RAD:

Command Prompt

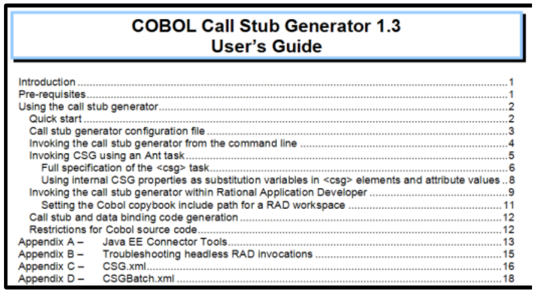
```
java -jar \
   lib/COBOLCallStubGenerator.jar \
   testcases/adder.cbl \
   -configFile csg.properties \
   -callStubPackage com.ibm.mycobol
```


IBM Rational Application Developer
RAD
CSG.xml
ANT Task
Call Stub and Data Bindings
csg.properties

Documentation for COBOL Container

Call Stub Generator User's Guide

- Reference for file properties & details on setup and usage

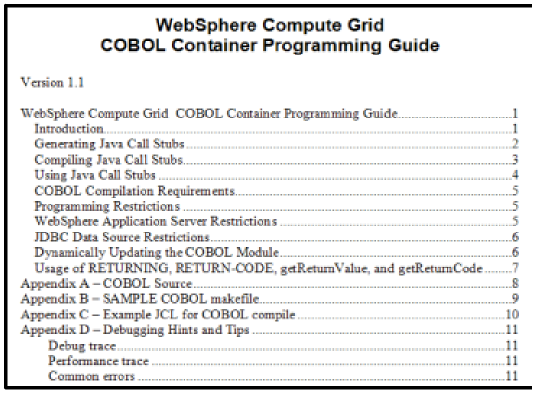


COBOL Call Stub Generator 1.3
User's Guide

Introduction 1
Pre-requisites 1
Using the call stub generator 2
Quick start 2
Call stub generator configuration file 3
Invoking the call stub generator from the command line 4
Invoking CSG using an Ant task 5
Full specification of the <csg> task 6
Using internal CSG properties as substitution variables in <csg> elements and attribute values .. 8
Invoking the call stub generator within Rational Application Developer 9
Setting the Cobol copybook include path for a RAD workspace 11
Call stub and data binding code generation 12
Restrictions for Cobol source code 12
Appendix A – Java EE Connector Tools 13
Appendix B – Troubleshooting headless RAD invocations 15
Appendix C – CSG.xml 16
Appendix D – CSGBatch.xml 18

COBOL Container Programming Guide

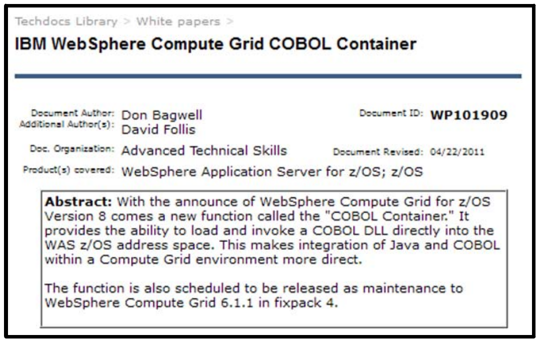
- Detail on programming to the call stubs and bindings



WebSphere Compute Grid
COBOL Container Programming Guide

Version 1.1

WebSphere Compute Grid COBOL Container Programming Guide 1
Introduction 1
Generating Java Call Stubs 2
Compiling Java Call Stubs 3
Using Java Call Stubs 4
COBOL Compilation Requirements 5
Programming Restrictions 5
WebSphere Application Server Restrictions 5
JDBC Data Source Restrictions 6
Dynamically Updating the COBOL Module 6
Usage of RETURNING, RETURN-CODE, getReturnValue, and getReturnCode 7
Appendix A – COBOL Source 8
Appendix B – SAMPLE COBOL makefile 9
Appendix C – Example JCL for COBOL compile 10
Appendix D – Debugging Hints and Tips 11
Debug trace 11
Performance trace 11
Common errors 11



Techdocs Library > White papers >

IBM WebSphere Compute Grid COBOL Container

Document Author: Don Bagwell
Additional Author(s): David Follis
Document ID: WP101909
Doc. Organization: Advanced Technical Skills
Document Revised: 04/22/2011
Product(s) covered: WebSphere Application Server for z/OS; z/OS

Abstract: With the announce of WebSphere Compute Grid for z/OS Version 8 comes a new function called the "COBOL Container." It provides the ability to load and invoke a COBOL DLL directly into the WAS z/OS address space. This makes integration of Java and COBOL within a Compute Grid environment more direct.

The function is also scheduled to be released as maintenance to WebSphere Compute Grid 6.1.1 in fixpack 4.

WP101909 Techdoc

"WebSphere Compute Grid COBOL Container"

- Technical Executive Color Flyer
- Architectural Comparison Document
- Contains the two documents shown above
- Available on www.ibm.com/support/techdocs

Migration from Compute Grid V 6.1.1 or FP

Notes:

- ***WCG V. 6.1 not supported on WAS V. 8.0.***
- ***WCG V. 8.0 not supported on WAS V. 6.1.***

Migrate the nodes in the following order:

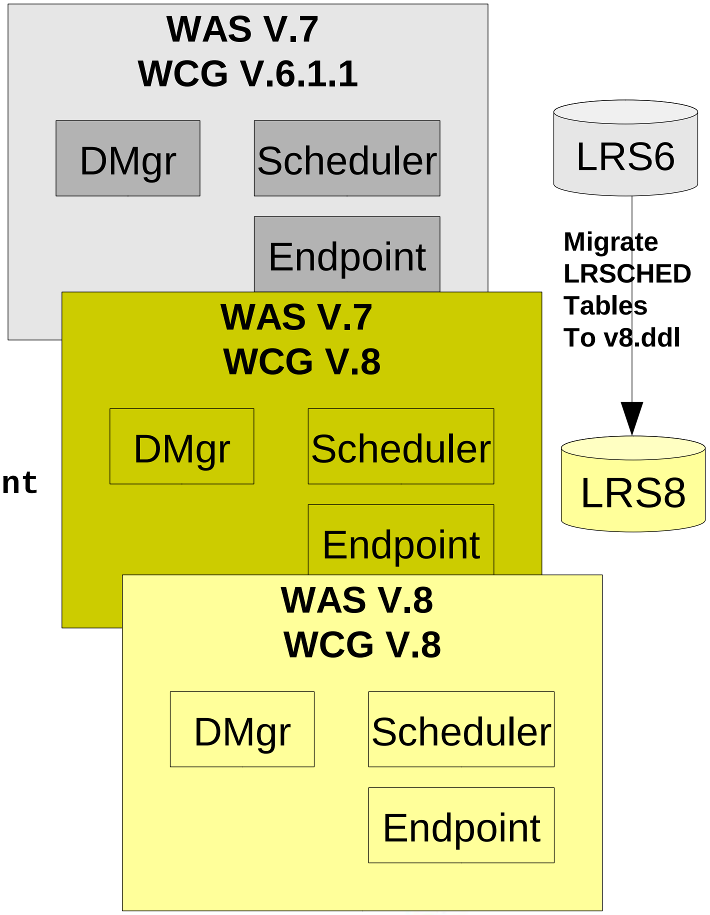
A) Deployment manager.
B) Migrate the databases.
C) Schedulers and endpoints nodes one at a time.

Migrate the Deployment Manager:

1. Run the backup script **`migrateWCGConfigTo8.py --backup`**
2. Stop the deployment manager.
3. Unaugment Dmgr: **`manageprofiles.sh -unagument`**
4. Uninstall XD Compute Grid V. 6.1.1.3 or the Batch FeP
5. Install XD Compute Grid V. 8.0. **(Installation Manager)**
6. Augment the deployment manager. **`manageprofiles.sh -agument`**
7. Migrate the Database(s) **`MigrateLRSCHEDTablesTov8.ddl`**
8. Start the deployment manager.
9. Run the restore script **`migrateWCGConfigTo8.py --restore`**

Migrate the Scheduler and Endpoint nodes:

10. Stop the server and node.
11. Unaugment the profile. **`manageprofiles.sh -unagument`**
12. Uninstall XD Compute Grid V. 6.1.1.3 or the Batch FeP
13. Install XD Compute Grid V. 8.0. **(Installation Manager)**
14. Augment the scheduler node. **`manageprofiles.sh -agument`**
15. Run the restore script **`migrateWCGConfigTo8.py --restore`**
16. Start the server.



WAS V.7
WCG V.6.1.1
DMgr
Scheduler
Endpoint

LRS6

Migrate LRSCHED Tables To v8.ddl

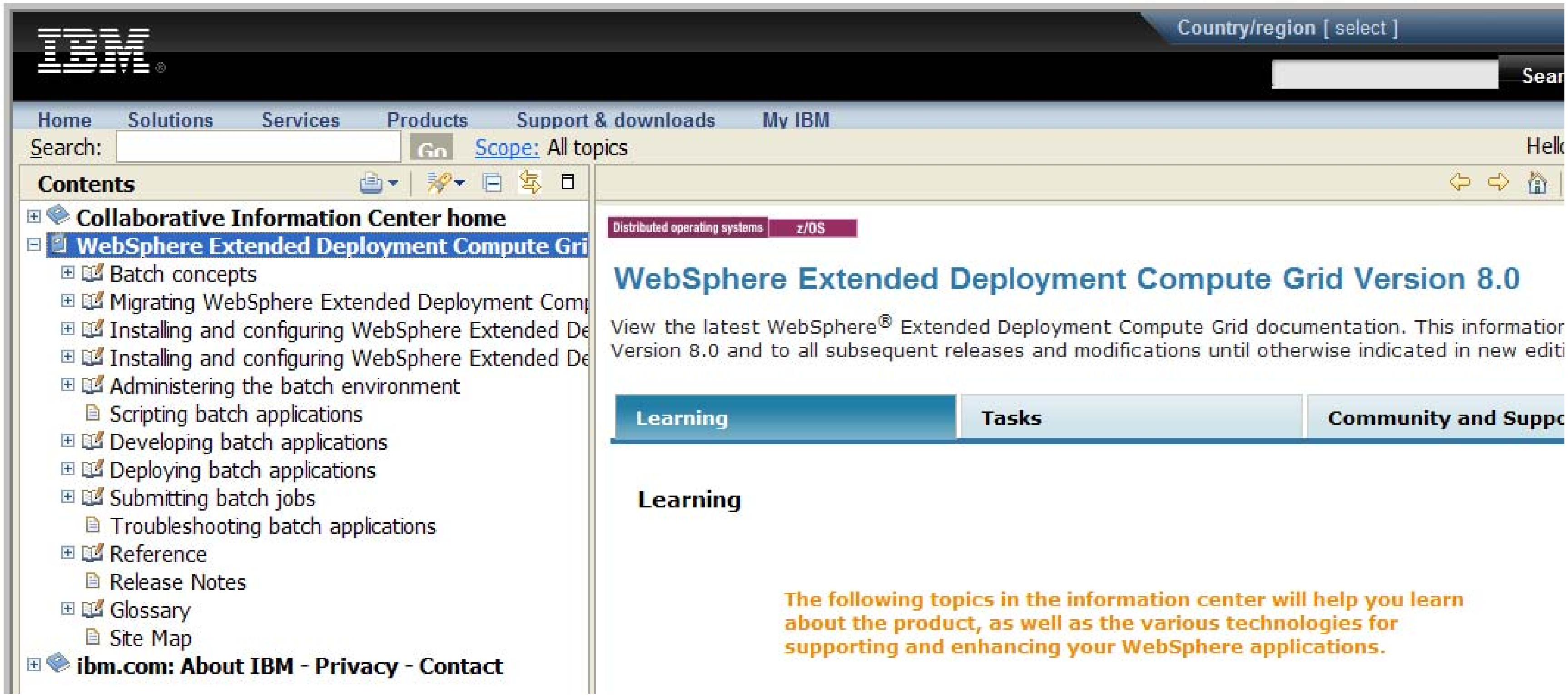
LRS8

WAS V.7
WCG V.8
DMgr
Scheduler
Endpoint

WAS V.8
WCG V.8
DMgr
Scheduler
Endpoint

Information Center & other Resources

http://publib.boulder.ibm.com/infocenter/wasinfo/cgwas80/index.jsp?topic=/com.ibm.websphere.cgwas.doc/info/ae/ae/welcome_cg80.html



- Collaborative Information Center home
- WebSphere Extended Deployment Compute Gri
  - Batch concepts
  - Migrating WebSphere Extended Deployment Comp
  - Installing and configuring WebSphere Extended De
  - Installing and configuring WebSphere Extended De
  - Administering the batch environment
  - Scripting batch applications
  - Developing batch applications
  - Deploying batch applications
  - Submitting batch jobs
  - Troubleshooting batch applications
  - Reference
  - Release Notes
  - Glossary
  - Site Map
- ibm.com: About IBM - Privacy - Contact

WebSphere Extended Deployment Compute Grid Version 8.0

View the latest WebSphere® Extended Deployment Compute Grid documentation. This informatior Version 8.0 and to all subsequent releases and modifications until otherwise indicated in new editi

Learning

The following topics in the information center will help you learn about the product, as well as the various technologies for supporting and enhancing your WebSphere applications.

Library page: http://www-01.ibm.com/software/webservers/appserv/extend/computegrid/library/
Download docs: http://www.ibm.com/support/docview.wss?uid=swg27021566&wv=1
Techdocs: PRS4644, PRS4467, WP101783 & WP101909
PRS4686 - WAS z/OS Version 8 Configuration Spreadsheets
WP101936 - Migrating to WebSphere XD Compute Grid v8 on z/OS

WebSphere Application Server Sessions

Day	Time	#	Title	Speaker	Room
Wednesday	3:00	9483	Using IBM's New Cross-Platform Installer on z/OS	Mierzejewski	Oceanic 5
Thursday	8:00	9482	WAS Version 8 – Overview	Follis	Europe 2
Thursday	9:30	9486	WAS Version 8 – Batch Update	Hutchinson	Europe 2
Thursday	11:00	9485	WAS Version 8 – New z/OS Exploitation/Differentiation Features	Follis	Europe 2
Thursday	1:30	9484	WAS Version 8 – High Availability Enhancements	Follis	Europe 2
Thursday	3:00	9488	WAS - Back to Basics Part 1	Loos	Europe 2
Thursday	4:30	9489	WAS - Back to Basics Part 2	Stephen	Europe 2
Friday	8:00	9490	WAS for z/OS - Level 2 Update	Stephen	Europe 2
Friday	9:30	9487	WAS for z/OS – PotPourri	Follis, Hutchinson, Loos, Stephen, etc.	Europe 2