# WebSphere Application Server Sessions

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

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- CICS®
- DB2®
- GDPS®
- Geographically Dispersed Parallel Sysplex
- HiperSockets
- IBM®
- IBM eServer
- IBM logo®
- IMS
- On Demand Business logo

* Registered trademarks of IBM Corporation

Parallel Sysplex®
RACF®
System z®
WebSphere®
z/OS
zSeries®

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
Disclaimer

- The information contained in this documentation is provided for informational purposes only. While efforts were many to verify the completeness and accuracy of the information contained in this document, it is provided “as is” without warranty of any kind, express or implied.

- This information is based on IBM’s current product plans and strategy, which are subject to change without notice. IBM will not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation.

- Nothing contained in this documentation 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 the IBM software.

- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.

- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
WOLA 7.0.0.4 and 7.0.0.12
In The Beginning Was "Local Comm"

This has been around since the early days of WAS on z/OS. It's a way to bypass the TCP/IP stack for internal IIOP calls between servers on the same LPAR:

If an IIOP call is made and WAS z/OS sees it's on the same LPAR, then this is automatically done.

The Daemon server plays a key role in this; it owns the above-the-bar space used shared space and does the inter-address switching.

It's very fast with very low overhead.
The Motivation Behind WOLA

It started out as a way to allow program access into WAS for high transaction rate batch programs. Other solutions existed, but they all had limitations:

Inbound to WAS?
As more and more solutions are built based on Java EE, there is a growing desire to access them by batch, CICS and IMS programs

MQ or Web Services?
Both are very good technologies and have their role. But for very high throughput and low overhead, each has their drawbacks.

Something else was needed ... something very fast with as little overhead per exchange as possible.
Answer: Externalize the Local Comm Function

The Local Comm function was there. It just needed interface modules so external address spaces could access it:

WOLA was born
1. Existing Local Comm exploited
2. Externalization routines written
3. Programming APIs for external address spaces provided
4. Standard JCA adapter for the WAS server provided

Just Inbound? No!
This is a bi-directional technology. Outside into WAS, and WAS out to external address space
Concept #1 - WOLA is a Byte Array Pipe

As such it pays no attention to format or code page. That's why it's so fast.

The two sides of the exchange must have some awareness of each other so that the data can be in the proper format, layout and codepage.

This can be XML if you want. Or COBOL COPYBOOK format. Or whatever. WOLA doesn’t care.

There are Eclipse based wizards to help with COBOL COPYBOOKS for CICS programs.
Concept #2 - WOLA is Address Space to Address Space

This is a very low-level mechanism between address spaces:

To exchange with an application in a server, it is required to register to *that specific server*

WOLA does not provide general access to the WAS cell, it provides very low-level access to the specific server and applications in that server

Multiple registrations permitted, to the same server or different servers
WOLA and CICS, Transaction and Security

**WAS z/OS**

**Outbound**

- **Transaction = 2PC**
  - Requires WAS 7.0.0.12 or higher and CICS 4.1 or higher
  - **Security = ID on WAS thread**

- **Transaction = None**
  - **Security = No ID Propagation**

**CICS**

**Inbound**

- **Transaction = 2PC**
  - Available since original 7.0.0.4
  - **Security = Region ID or Application User ID**

- **Bypass Link Server for maximum performance**

See the WP101490 "Design and Planning Guide or the InfoCenter for the specific details of this
High Level Overview of IMS Support – all new with 7.0.0.12

- **Daemon**
  - CR

- **DMGR**
  - CR
  - SR

- **Node Agent**
  - CR

- **AppServer**
  - CR
  - SR

- **Batch Controller**
  - Batch DL/I Apps

- **Exchange**
  - Control Blocks
  - Owned by Daemon
  - 2GB Bar

- **WOLA over OTMA**

- **WOLA**
  - and/or
  - Batch Controller
  - Batch DL/I Apps
  - and/or
  - WOLA

- **Dependent Region**
  - Message Processing Program (MPP)
  - Batch Message Processing Program (BMP)
  - IMS Fast Path (IFP)

- **Controlling Region**

- **External Subsystem Attach Facility (ESAF)**

- **No WOLA enablement needed**

- **Write to the WOLA APIs**

- **Shielded**

- **SHARE in Orlando 2011**
WOLA and IMS, Transaction and Security

A summary picture:

WAS z/OS

Outbound

Transaction = None

Security = No ID propagation

Transaction = Sync On Return

Security = ID on IMS thread

WOLA APIs

Inbound

SyncLevel = None | Confirm

Security = ID on WAS thread

MPP and IFP, but not BMP

IMS

OTMA

WOLA APIs

Bypass OTMA for maximum performance

 IMS ESAF

WOLA APIs
Version 8
## Versions, Dates, and Service Levels...

<table>
<thead>
<tr>
<th>Version</th>
<th>GA</th>
<th>End of Marketing</th>
<th>End of Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 7</td>
<td>9/26/2008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version 8</td>
<td>6/17/2011</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **End of Service announced for V6.1**
- **GA of Version 8**
- **Service level naming convention change:**
  - V6.1 and V7 – even numbers z/OS only, odd numbers common
  - V8 – all levels are common (no more z/OS only levels)
The Key Technical Messages

Further refinement of proven base
• Updated specifications (Java EE 6, EJB 3.1, Servlet 3.0, JSP 3.0, etc)
• Enhanced web services: JAXB, JAX-RE, JAX-WS
• Feature packs rolled in: SCA, XML, OSGi, JPA, Web 2.0, WOLA, Batch
• Continued focus on tightening and refining programming

Enhanced commonality across platforms
• Common installation process using Installation Manager
• Common high-performance logging function

Greater z/OS exploitation
• More granual settings for timeouts and tracing w/ dynamic control
• Resource routing for data source and connection factories
WAS V8: Enhanced, Not Retooled

Administrative Console Skills

Application Development Skills

WSADMIN Scripting Assets and Skills

z/OS Operator Skills

Server Runtime Construction Skills

Version 6.1
Version 7.0
Version 8
Rollup of Prior Feature Packs

SCA
XML
OSGi
JPA
Modern Batch
Web 2.0
Mobile
WOLA

WebSphere Application Server V7.0

Applied Feature Packs

WAS Administrator

Manually apply feature packs according to need

WebSphere Application Server V8.0

All the feature pack function is rolled into the WAS V8 product

Manual apply feature packs according to need
New Version = Updated/New Specs

Best place to understand the new specs is to go to the InfoCenter and search on rovr_specs

### Java EE 6 (JSR 316)
Continues trend towards increased function and a simpler development model

### Java Servlet 3.0 (JSR 315)
Enhancements to support modern web development

### EJB 3.1 (JSR 318)
Further simplifies development of EJBs

### JCA 1.6 (JSR 322)
Update specification architecture based on feedback from experts and users

---

**Specifications and API documentation**

The WebSphere® Application Server product supports various J2EE specifications and application programming interface (API) documents in previous product releases.

**Components**
- Any application type
- Web applications
- Portlet applications
- SIP applications
- EJB applications
- Client applications
- Web services
- Service Component Architecture
- Service integration
- Data access resources
- Messaging resources
- Mail, URL, and other Java EE resources
- Security
- Web Services Security
- Naming and directory
- Object Request Broker
- Transactions
- WebSphere extensions
- Administration

---

**Table 1: Supported specifications for any application type**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Version 8.0</th>
<th>Version 7.0</th>
<th>Version 6.1</th>
<th>Version 6.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java EE 6 (JSR 316) New</td>
<td>Java EE 6 New</td>
<td>Java EE 6 New</td>
<td>Java EE 1.4 New</td>
<td>Java EE 1.4 New</td>
</tr>
<tr>
<td>Java EE 5</td>
<td>Java EE 5</td>
<td>Java EE 1.3</td>
<td>Java EE 1.3</td>
<td>Java EE 1.3</td>
</tr>
<tr>
<td>J2EE 1.4</td>
<td>J2EE 1.4</td>
<td>J2EE 1.2</td>
<td>J2EE 1.2</td>
<td>J2EE 1.2</td>
</tr>
<tr>
<td>J2EE 1.3</td>
<td>J2EE 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ISO 8859 specifications

ISO 8859 applies to these versions.
Java, JVM, and Platform-Awareness

Version 8 still provides Java 6 (though called “6.0.1”). What’s new is the JVM inside the supplied Java, and the z196-awareness:

Application still “sees” Java 6, but benefits from the performance enhancements below

Java 6 Specification Definition

- Significant enhancements to JIT optimization technology
- New Balanced GC policy to reduce max pause times
- z196 exploitation of instructions and new Out-of-Order-Execution pipeline

Java Virtual Machine

z-Aware Native Layer

z196 processor has faster chips, larger and better cache, an Out-of-Order-Execution pipeline and additional instructions
Background on IBM Installation Manager

IBM Installation Manager is a product intended to install software, update software, and keep track of levels installed.

Introduced in 2006, it now has over 120 IBM products using it

Things it can do:
- Install software
- Update with fix packs
- Modify features and functions
- Rollback features and functions
- Uninstall software

Most commonly thought of as a workstation tool, it does have a z/OS command line component

For WebSphere Application Server z/OS V8, IM is used for two things:

1. Creating/maintaining the hlq.SBBOHFS file system on z/OS
2. Installing/maintaining the WCT V8 tool on your workstation
Key Concept: IM “Repository”

Think of a repository as the place where Installation Manager goes to get the software files and any updates to installed software:

- IBM Software Repository Maintained at an Internet-Accessible site
- I’m keeping track of what’s been installed, the levels and the features selected
- No physical media

Greatly simplifies updates and feature additions and removals

Details not shown: how identity/security is provided, where repository is located, steps to install software
The “WebSphere Customization Tool” (WCT) is what’s used to create the customized jobs that build your runtime. IM installs that for WCT V8:

**Process:**

- Install IM on your workstation (no-charge software; download, double-click and take most defaults)
- Point IM to IBM repository where WCT V8 is available (WCT is no-charge as well)
- Use WCT much like you did for Version 7:
  - Plan runtime with spreadsheet (updated for V8)
  - Generate customized jobs and upload to z/OS system
  - Submit execute jobs
  - Start the runtime
- Use IM to updated WCT with any extensions for feature packs or stack products
IM and z/OS

This is entirely new for WAS z/OS V8 … the use of command line IM on z/OS to create and maintain the hlq.SBBOHFS file product file system:

Installation Manager z/OS Command Line

- Initial creation
- Updates

```
SERVICE.hlq.SBBOHFS
/SERVICE/usr/lpp/zWebSphere/V8R0

hlq.SBBOHFS
/usr/lpp/zWebSphere/V8R0
```

WAS z/OS V8 Repository

This is what we need to focus on!

WAS z/OS Runtime

Symlinks to the files in hlq.SBBOHFS

Essentially same process as before
Installing the Installation Manager Itself

It’s delivered as an SMP/E package. The result is a file system that contains the IM code. You then run simple job to create runtime instance:

SMP/E installable Installation Manager
(Comes with WAS z/OS V8)

SMP/E Process

GIN2INST job creates a function instance of the Installation Manager

SMP/E Administrator

/SERVICE/usr/lpp/InstallationManager/V1R4

SMP/E managed copy of the IM installation image

Product Repository

Initial WAS V8 Install
Fixpacks, Feature Packs, Stack Products

Focus on this next!

Installation Manager z/OS
Command Line

/InstallationManager/V1R4

IM managed copy of the WAS z/OS V8 hlq.SBBOHFS product file system
The WAS V8 Product Repository

We have a two-part story here: one for the initial product delivery, then the opportunity to use IBM’s hosted repository for fixpacks/feature packs.

Initial Acquisition of WAS z/OS V8 from IBM

Fixpack and Feature Pack Updates

- SMP/E installable
- Local Repository
- SMP/E Process
- CSI
- Local Repository (Mounted File System)

- Internet
- IBM-hosted Repository
- /InstallationManager/V1R4
- Installation Manager z/OS Command Line
- SERVICE.hlq.SBBOHFS
- /SERVICE/usr/lpp/zWebSphere/V8R0
Summary of Installation Manager

- IBM’s key software installation and update management product
- Used extensively by other IBM products (i.e. Rational tools)
- Now used for WAS z/OS V8
- SMP/E still plays a role for now:
  - Installation of IM itself
  - Installation of the initial local repository
- Fixpacks and updates may be then drawn from IBM hosted repository
  - IM generates UCLIN you may then apply to keep SMP/E information current
- Feature Packs binaries are added to WAS z/OS product file system, not hung off to side and symlinked to as it’s done today
- Stack products may be installed through IM
- Add hot updates without need for ++APAR
- Back-out updates easily using IM

IM is a powerful product and is the direction IBM is headed. Some learning curve involved but we are confident you will become proficient quickly and will come to understand the value of IM into the future
High Performance Extensible Logging (HPEL)
Very High Level Overview of HPEL

This is an **optional** mechanism to format traces and logs into a WAS binary format. A utility is then used to offload to a viewable text file:

Configurable on a Server-by-Server Basis

WebSphere Application Server Version 8

All Supported Platforms

- Java Trace
- Java Logs
- System.out
- System.err

More efficient use of space, faster write operations

WAS-specific binary format log file

Write to memory buffer, then file

Controls to dictate size limits, what to do when limit reached, how to trim files, start new files, etc.

Log Viewer Utility

Common across all platforms

Use whatever view/edit tool you prefer

ASCII readable file
For z/OS: Output We Have Today

This is known as “Basic” in Version 8. On the next chart we’ll see where HPEL affects the picture.

Someone unfamiliar with System z had to navigate around z/OS-specific log and trace activity to see what they needed to see.

This portion could be routed to JES or USS/MVS output.
For z/OS: With HPEL in Effect for Server

And here’s what it looks like when you put HPEL into effect:

- Java Trace
- Java Logs
- System.out
- System.err
- zNative Trace
- cout
- cerr
- zNative Message

These now contain z-specific content of interest to system programmers

Better leveraging and focusing of existing skills in the organization

Log and trace format consistent across all platforms
Data Resource Routing
High Level Overview of this Function

This function allows you to configure an alternative connection factory so in the event of a data resource loss the defined backup can be used:

- **DB2 z/OS**
- **IMS** if you’re at a relatively recent level of maintenance
- **Not yet CICS** if using CTG functionality

Key is ability of WAS to detect loss of backend
Further z/OS Exploitation

Some of what was shown on prior page is common across platforms. WAS z/OS V8 adds an additional layer:

Imagine the backup DB2 is also lost, or you’ve not configured for an alternative connection factory. What then?

Three further configurable options:

1. Have WAS issue a Write to Operator (WTO) and use that to take further system automation action

2. Have WAS issue PAUSELISTENERS, thus turning off input ports to server, allowing front-end routers to detect and go elsewhere

3. Have WAS determine which applications are affected and stop just those applications, leaving other applications up and running
WOLA Variation on This New Function

WOLA participates in this as well in that a backup registered external address space now be used in the event the primary is lost:

For example, imagine two CICS regions on the same LPAR

WOLA is by definition “same LPAR,” and this gives you a degree of availability by allowing routing to secondary registered external address space
Granular RAS
RAS Function Control Down to Request

This new function leverages existing classification file to extend scope of certain RAS functions down to the request level:

Not just HTTP … any classifiable input type (IIOP, MDB, etc.)

Additional XML values permitted in the classification file

These control behavior when classification identification is made by this function

MODIFY command allows you to dynamically enable a new file, or dynamically revert back to previous

Enhances granularity. Previously down to server level; now to request.
### WAS 8 Security Default Changes

<table>
<thead>
<tr>
<th>Feature</th>
<th>WAS 8</th>
<th>Prior release</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMM Active Directory for User look up for performance</td>
<td>(ObjectCategory=User)</td>
<td>(ObjectClass=Person)</td>
</tr>
<tr>
<td>VMM Active Directory and Adam for Group membership lookup for performance</td>
<td>&quot;memberOf&quot; attribute with &quot;Direct&quot; scope</td>
<td>none</td>
</tr>
<tr>
<td>VMM LDAP attribute search cache dist policy</td>
<td>None</td>
<td>Push</td>
</tr>
<tr>
<td>EJB/CSIv2 transport</td>
<td>SSL-required</td>
<td>Supported</td>
</tr>
<tr>
<td>Session security integrity</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>&quot;Use available authentication data when an unprotected URI&quot;</td>
<td>Checked</td>
<td>Uncheck</td>
</tr>
<tr>
<td>HttpOnly</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Generated Certificate Key Length</td>
<td>2048</td>
<td>1024</td>
</tr>
<tr>
<td>z/OS java.security file location</td>
<td>/WAS_HOME/properties</td>
<td>jre/lib/security</td>
</tr>
<tr>
<td>DataPower Certificated</td>
<td>Now need to import</td>
<td></td>
</tr>
</tbody>
</table>
Companion Videos on YouTube

<table>
<thead>
<tr>
<th>Title</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>WASZOSV8 - Intro to WAS V8 - Part 1 of 4</td>
<td>4m:50s</td>
</tr>
<tr>
<td>WASZOSV8 - Intro to WAS V8 - Part 2 of 4</td>
<td>6m:30s</td>
</tr>
<tr>
<td>WASZOSV8 - Intro to WAS V8 - Part 3 of 4</td>
<td>8m:54s</td>
</tr>
<tr>
<td>WASZOSV8 - Intro to WAS V8 - Part 4 of 4</td>
<td>18m:00s</td>
</tr>
</tbody>
</table>

Search on key string WASZOSV8 to find these videos
<table>
<thead>
<tr>
<th>Part</th>
<th>Title and URL</th>
<th>Duration</th>
<th>Abstract</th>
</tr>
</thead>
</table>
| 1    | WP101490 - WOLA - Essentials of WOLA  
http://www.youtube.com/watch?v=bpUiJ-wp0qA | 11:59    | Provides an overview of the framework and concepts of WOLA |
| 2    | WP101490 - WOLA - CICS  
http://www.youtube.com/watch?v=d3WWNWiQMJI | 14:21    | Provides an overview of how WOLA is implemented in CICS |
| 3    | WP101490 - WOLA - IMS  
http://www.youtube.com/watch?v=BDBQ1qbsRqo | 7:07     | Provides an overview of how WOLA is implemented in IMS |
| 4a   | WP101490 - WOLA - Native APIs Part 1 of 2  
http://www.youtube.com/watch?v=lNyg6tMUKXc | 10:39    | First of two parts on the WOLA native APIs |
| 4b   | WP101490 - WOLA - Native APIs Part 2 of 2  
http://www.youtube.com/watch?v=7wNESRJ91XY | 14:07    | Second of two parts on the WOLA native APIs |
| 5    | WP101490 - WOLA - Java Considerations  
http://www.youtube.com/watch?v=oVFSY_F6ceY | 14:45    | Overview of the Java programming considerations for WOLA |

Videos about WOLA
## WebSphere Compute Grid and Modern Batch

<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Duration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WP101783 - IBM Modern Batch - Part 1 of 4</td>
<td>8:44</td>
<td>Overview of &quot;Batch Modernization&quot; and comparisons of technologies</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.youtube.com/watch?v=3eeUTp1IlIA">http://www.youtube.com/watch?v=3eeUTp1IlIA</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>WP101783 - IBM Modern Batch - Part 2 of 4</td>
<td>14:22</td>
<td>Overview of the IBM architectural approaches to modern batch</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.youtube.com/watch?v=kLtHsKeh1Ok">http://www.youtube.com/watch?v=kLtHsKeh1Ok</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>WP101783 - IBM Modern Batch - Part 3 of 4</td>
<td>9:02</td>
<td>Demonstration of job submission and execution</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.youtube.com/watch?v=jcyhaNtp4mI">http://www.youtube.com/watch?v=jcyhaNtp4mI</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>WP101783 - IBM Modern Batch - Part 4 of 4</td>
<td>12:55</td>
<td>Demonstration of scheduler integration</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.youtube.com/watch?v=IqqOiT119r4">http://www.youtube.com/watch?v=IqqOiT119r4</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>WP101783 - WCG/TWS Integration - Part 1 of 2</td>
<td>11:03</td>
<td>Powerpoint screenshow of concepts related to integration of schedulers with WCG.</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.youtube.com/watch?v=g_mLklap8bM">http://www.youtube.com/watch?v=g_mLklap8bM</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>WP101783 - WCG/TWS Integration - Part 2 of 2</td>
<td>8:51</td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.youtube.com/watch?v=NLuZYXUXWX0c">http://www.youtube.com/watch?v=NLuZYXUXWX0c</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATSWAS101 - The Essentials - Part 1 of 4</td>
<td>9:00</td>
<td>Background of what &quot;Application Server&quot; is.</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------</td>
<td>------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>ATSWAS101 - The Essentials - Part 2 of 4</td>
<td>8:27</td>
<td>Overview of Java and the JVM</td>
</tr>
<tr>
<td></td>
<td>ATSWAS101 - The Essentials - Part 3 of 4</td>
<td>11:26</td>
<td>Overview of WebSphere Application Server</td>
</tr>
<tr>
<td></td>
<td>ATSWAS101 - The Essentials - Part 4 of 4</td>
<td>5:09</td>
<td>Very quick peek at WAS z/OS; more to come.</td>
</tr>
<tr>
<td></td>
<td>ATSWAS201 - The Essentials - Part 1 of 3</td>
<td>7:59</td>
<td>Background on platform exploitation in general</td>
</tr>
<tr>
<td></td>
<td>ATSWAS201 - The Essentials - Part 2 of 3</td>
<td>14:22</td>
<td>A look at the split JVM model and some of the other ways WAS z/OS takes advantage of z/OS.</td>
</tr>
<tr>
<td></td>
<td>ATSWAS2101 - The Essentials - Part 3 of 3</td>
<td>12:10</td>
<td>A tour of an actual WAS z/OS ND cell.</td>
</tr>
</tbody>
</table>