WebSphere Application Server Versions: What's Different?

For V9.0 and earlier

Agenda

- **v9, v8.5, v8.0 and v7.0 Feature Highlights**
- High level view of Migration changes
- Modernizing to Liberty
- Changes per version
  - Changes in v9.0
  - Changes in v8.5
  - Changes in v8.0
  - Changes in v7.0
  - Changes in v6.1
  - Changes in v6.0

WAS V9: 2Q16 Key Announcements

New June 2016

Create new cloud apps

WAS V9 delivers robust & modern developer environment for speed & enterprise scalability

- Cloud-First Java app platform - Speed development with composable runtime and microservices architecture;
  - Full **Java EE7** certification for **both WAS Traditional and Liberty Profiles**;
  - Leverage **Spring and Spring Boot** frameworks within applications;
  - Quick start cloud native Java apps with **Liberty app accelerator** and **Game-On exemplar** – seamless deploy to IBM Bluemix
  - Leverage end-to-end **DevOps Toolchain** and **Garage Method** (best practices)
  - Portability of apps with Docker support; Deploy to IBM Container Service, Docker Data Center, and other container services
- Seamlessly manage Java and Node.js apps and APIs through common management interface

Revision Date 06/2016

WAS V9: 2Q16 Key Announcements (cont.)

New June 2016

Optimize costs for app infra

Move apps anywhere, in any way, to increase speed and optimize costs

- Create a borderless environment with easy app portability regardless of architectural environment (Cloud container services, Docker, VMWare)
- Exploit WAS ND intelligent management for workload optimization and placement
- Move apps to the IBM public cloud with choice of pre-configured environ, flexible PayGo pricing models
- WAS Liberty app deployment to OpenShift and Pivotal Cloud Foundry running in IBM SoftLayer, Amazon AWS, Microsoft Azure, and IBM Bluemix
- WAS on Bluemix- Single Tenant : New option to deploy on Cloud
- Ease of use enhancements for caching to optimize user experience

WAS V9: 2Q16 Key Announcements (cont.)

New June 2016

Connect existing apps & data

Connect to cloud services to integrate new capabilities, improve time to mkt & lower costs

- Seamlessly connect on-premises apps to cloud services like Watson, Cloudant, dashDB. API Connect , Log Analytics (beta) - take advantage of latest technologies and extend the value of existing Java apps
- Optimize use of APIs for exposing and better monetizing traditional apps
- Leverage PayGo models and eliminate risk and complexity of managing these new services
- Leverage “API Connect Essentials” now included in WAS editions

WebSphere Application Server

New June 2016

The cornerstone of your cloud strategy

30%+ TCO
WAS on Cloud Bluemix vs On-premises

122% ROI
Liberty vs Open Source

$325K annual infrastructure savings
by year 3 vs. Open Source

Industry Leading Security
- Open ID Connect
- Secure Engineering Accreditation O-TTPS

30% better performance
with Java 8

Intelligent Management
45% less hardware
60% admin savings
45% less software
90% fewer outages

CREATE

Developer focused to speed delivery pipeline

- Lightweight composable runtime - perfect for microservices
- Full integration with any DevOps toolchain for continuous delivery
- Java EE7 market leadership and support for Open Source

CONNECT

Easy cloud connections for new & existing apps

- Create, expose and connect APIs
- Re-use existing apps and connect to “on or off” premises
- Deploy anywhere - on premises, in cloud or hybrid

OPTIMIZE

Smart management of the mission critical

- Leading edge cloud & mobile security
- Enterprise Management of Java & Node.js
- High availability: auto-scaling, dynamic routing, health management, diagnostics

Revision Date 06/2016

WAS V8.5 Delivers

Unparalleled Application Development and Management Environment, Rich User Experiences…Faster

Developer Experience

Fast, flexible, and simplified application development

- Liberty Profile
- Expanded Tooling and WAS Tooling Bundles
- OSGI programming model enhancements
- EJB support in OSGi apps
- JDK7 Support
- Migration toolkit
- Web 2.0 & Mobile Toolkit; IBM Worklight Integration
- SCA OASIS programming model

Application Resiliency

Intelligent Management & Enhanced Resiliency

- Application Edition Management
- Application Server Health Management
- Dynamic Clustering
- New Intelligent Routing capabilities
- Messaging infrastructure resiliency
- Memory leak detection & protection in WAS

Operations and Control

Improved Operations, Security, Control & Integration

- Selectable JDK
- WebSphere Batch enhancements
- Admin Security Audit
- OSGi Blueprint security improvements
- Cross Component Trace (XCT)
- Enhanced IBM Support Assistant
- Better log and trace filtering

Revision Date 06/2016

WAS v8.5.5 Delivers

***Fit for Purpose Servers** enabling unmatched combination of Application Server runtime and development experience, from the highly resilient to the lightweight and nimble*

Developer Experience

Fast, flexible, and simplified application development

- New lightweight **WebSphere Application Server Liberty Core** edition
- Liberty Profile
  - Java EE 6 Web Profile
    - EJB Lite, CDI, Managed Beans
  - Web Services
  - JMS, MDB
  - NoSQL DB (MongoDB Client)
  - Support for WebSphere Web Cache (DynaCache)
  - Add custom and third party Liberty features
- Asynchronous work management
- Enhanced developer tools
- Supported WAS and WDT on developer desktops

Application Resiliency

Intelligent Management and Enhanced Resiliency

- WAS ND and WAS z/OS full profile enhancements in v8.5 **
  - App Edition Mgmt
  - App Server Health Mgmt
  - Dynamic Clustering
  - Intelligent Routing
  - Messaging resiliency
  - Enterprise Java Batch
  - Memory leak protection
- Liberty Profile collective administration
- Liberty Profile clustering
- Intelligent management in WebSphere web server

Operational Excellence

Improved Operations, Security, Control & Integration

- Liberty profile
  - Security enhancements
  - Problem determination
  - Monitoring
- Service Mapping
- SIP improvements
- Improved Load Balancer for IPV4 and IPV6
- Liberty profile packaging and install enhancements
- WebSphere Extreme Scale integration
- Performance enhancements

**Available since WAS 8.5

Revision Date 06/2016

WebSphere Application Server v8.0

Speed Delivery of Applications & Services

- Open Source to Enterprise
- Free WAS for Developers
- Self Service Development Environments
- Faster Edit-Compile-Debug
- Programming Models
  - Java EE 6
  - Web 2.0 & Mobile
  - OSGi Applications
  - SCA
  - Java Batch
  - XML
  - SIP & CEA
  - Dynamic Scripting
- Integrated Tooling
- Application Adapters

Operational Efficiency & Reliability

- High Performance
- Transactional Strength
- Scalability & HA
- Install & Maintenance
- Problem Determination
- Platform & Environment Flexibility
- Flexible Pricing Models
- Feature Packs

Security & Control

- Administrative Productivity
- OSGi Application Agility
- Security
- Migration

IBM WebSphere Application Server Version 7.0

Simplification for Developers

- New and enhanced Standards: JDK 6.0, Java EE 5 certification, EJB3, Web Services, SIP, Portlet
- Web 2.0 Feature Pack
- New Rational Application Developer Support

Intelligent Management

- New Flexible Management: Job Manager, Administrative Agent
- New WebSphere Business Level Applications (BLAs)
- New Centralized Installation Manager

High Performance Foundation for SOA

- Multi-Cell Support
- Application investment protection
- Performance leadership
- New Runtime Provisioning
- New and Enhanced Security Features
- New Consolidated WebSphere and DataPower administration

Innovation that Matters

Feature Pack for Web 2.0
Feature Pack for SCA
Feature Pack for Communications Enabled Applications
Feature Pack for XML
Feature Pack for OSGI and JPA
Feature Pack for Dynamic Scripting
Feature pack for Modern Batch

Revision Date 06/2016

Agenda

- v9, v8.5, v8.0 and v7.0 Feature Highlights
- **High level view of Migration changes**
- Modernizing to Liberty
- Changes per version
  - Changes in v9.0
  - Changes in v8.5
  - Changes in v8.0
  - Changes in v7.0
  - Changes in v6.1
  - Changes in v6.0

Migration impacts (worst case scenario)

Updated June 2016

Potential Impact areas	v6.0	v6.1	v7.0	v8.0	v8.5	V9.0
Java Runtime	n/a	6	2	n/a	0/20/19	19
JEE - JSP	8	n/a	1	1	0	0
JEE - Servlet	5	n/a	0	2	0	9
JEE - Other	3	n/a	5	7	1	50
WAS Specific	1	6	0	4	0	3
3rd party packages	2	0	1	0	0	3
Development total	**19**	**12**	**9**	**13**	**0/20/19**	**64**
Administrative script	4	3	2	0	0	1
WAS directory structure	1	1	0	0	0	1
Other administrative	5	2	6	7	0	1
Total administrative	**10**	**6**	**8**	**7**	**0**	**3**
Total potential impact areas	**29**	**18**	**17**	**20**	**0/20/19**	**67**

Note: V8.5 supports Java 6, 7 and 8. Java7 and Java 8 introduces a number of behavior changes. “0” represents Java6 and “20” is for Java7. Not all breaking changes will impact all applications

Revision Date 06/2016

JEE specification change analysis (1 of 3)

traditional WebSphere

Updated June 2016

	v6.0	v6.1	v7.0	v8.0	v8.5			V9.0
Java Runtime	1.4.2	1.5	1.6	1.6	1.6	1.7	1.8	1.8
WebApps								
Servlet	2.4	2.4	2.5	3.0				3.1
JSF	1.0	1.1	1.2	2.0				2.2
JSP	2.0	2.0	2.1	2.2				2.3
EL								3.0
Portlet		1.0	2.0	2.0				2.0
SIP		1.1(FeP)	1.1	1.1				1.1
Enterprise Apps								
Bean validation				1.0				1.1
CDI				1.0				1.2
EJB	2.1	2.1/3.0(FeP)	3.0	3.1				3.2
JDBC	3.0	3.0	4.0	4.0				4.1
JMS	1.1	1.1	1.1	1.1				2.0
JPA		1.0(FeP)	2.0(FeP)	2.0				2.1
WebSocket								1.1
Java Web Start	1.4.2	1.4.2	1.4.2	1.4.2				1.6

Revision Date 06/2016

Red indicates significant, Yellow indicate some breakage, Gray indicates deprecation.

JEE specification change analysis (2 of 3)

traditional WebSphere

Updated June 2016

	v6.0	v6.1	v7.0	v8.0/v8.5	V9.0
Web Services					
JAXB		2.0	2.1	2.2	2.2
JAXP	1.2	1.3	1.4	1.4	1.4
JAXR	1.0	1.0	1.0	1.0	1.0
JAX-RPC	1.1	1.1	1.1	1.1	1.1
JAX-RS				1.1	2.0
JAX-WS		2.0	2.1	2.2	2.2
RAMP		1.0	1.0	1.0	1.0
SOAP	1.1	1.2	1.2	1.2	1.2
SAAJ	1.2	1.3	1.3	1.3	1.3
MTOM	n/a	1.0	1.0	1.0	1.0
StAX	n/a	1.0	1.0	1.0	1.0
UDDI	3.0	3.0	3.0	3.0	3.0
W3C XML	1.0	1.0	1.0	1.0	1.0
WS-Addressing	n/a	1.0	1.0	1.0	1.0
WS-AT	1.0	1.0	1.2	1.2	1.2
WS-BA	n/a	1.0	1.2	1.2	1.2
WSDL	1.1	1.1	1.1	1.1	1.1
WS-I Basic Profile	1.1	1.1	1.1	1.1	2.0

Revision Date 06/2016

Red indicates significant, Yellow indicate some breakage, Gray indicates deprecation.

JEE specification change analysis (3 of 3)

traditional WebSphere

Updated June 2016

	v6.0	v6.1	v7.0	v8.0/v8.5	V9.0
Web Services…					
WS-I Attachments	1.0	1.0	1.0	1.0	2.0
WS-Notification		1.3	1.3	1.3	1.3
WS-Policy			1.5	1.5	1.5
WS-ReliableMessaging		1.1	1.1	1.1	1.1
WSRF		1.2	1.2	1.2	1.2
XOP		1.0	1.0	1.0	1.0
ServiceComponentArch					
SCA*			1.0	1.0	Removed
SDO			2.1.1	2.1.1	2.1.1
Data Access					
JCA	1.5	1.5	1.5	1.6	1.7
JMS	1.1	1.1	1.1	1.1	2.0
Mail, URL and Other					
JavaMail	1.3	1.3	1.4	1.4	1.5
URL	1.4.2	1.4.2	1.4.2	1.4.2	1.4.2
JAF	1.0.2	1.0.2	1.1	1.1	1.1

Agenda

- v9, v8.5, v8.0 and v7.0 Feature Highlights
- High level view of Migration changes
- **Modernizing to Liberty**
- Changes per version
  - Changes in v9.0
  - Changes in v8.5
  - Changes in v8.0
  - Changes in v7.0
  - Changes in v6.1
  - Changes in v6.0

Modernizing to Liberty

Eliminate Future Migration Costs

New Liberty features and configuration, augment rather than replace

Zero migration for unchanged apps on WAS Liberty, regardless of Java EE version

Liberty Zero Migration

Config

- Write once, run forever

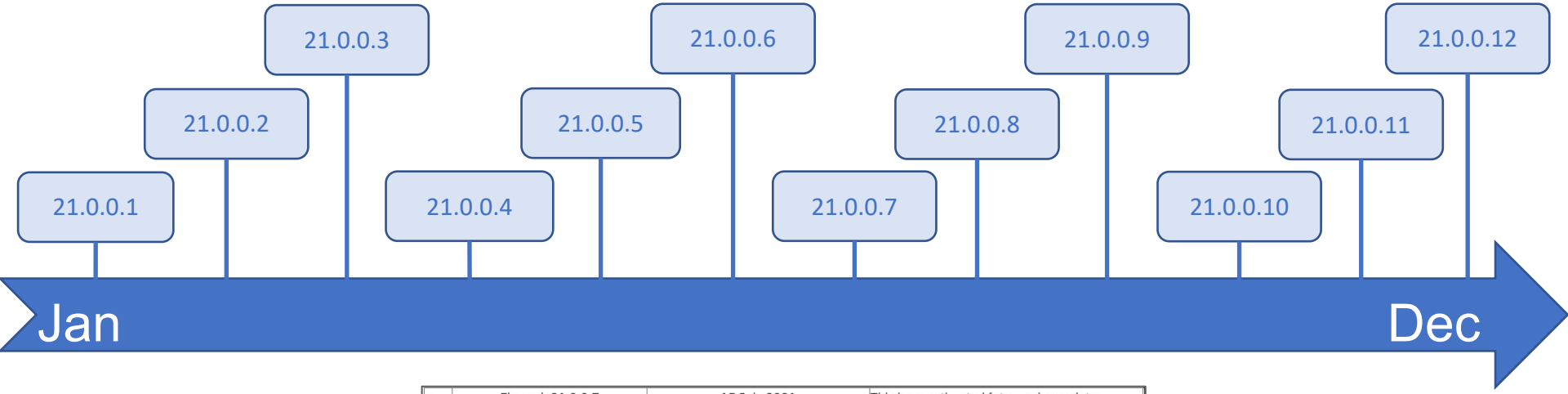
Apps

- No behavior changes in existing features
- New behaviors in new features

Choose your Java

- Java 11, 8
- AdoptOpenJDK
- IBM
- OpenJDK
- Oracle

4-Week Releases since 2019



▫	Fix pack 21.0.0.7	15 July 2021	This is an estimated future release date.
▫	Fix pack 21.0.0.6	18 June 2021	
▫	Fix pack 21.0.0.5	21 May 2021	
▫	Fix pack 21.0.0.4	23 April 2021	
▫	Fix pack 21.0.0.3	26 March 2021	
▫	Fix pack 21.0.0.2	26 February 2021	
▫	Fix pack 21.0.0.1	29 January 2021	
▫	Fix pack 20.0.0.12	27 November 2020	
▫	Fix pack 20.0.0.11	30 October 2020	
▫	Fix pack 20.0.0.10	2 October 2020	
▫	Fix pack 20.0.0.9	4 September 2020	
▫	Fix pack 20.0.0.8	7 August 2020	
▫	Fix pack 20.0.0.7	9 July 2020	
▫	Fix pack 20.0.0.6	12 June 2020	
▫	Fix pack 20.0.0.5	15 May 2020	
▫	Fix pack 20.0.0.4	17 April 2020	
▫	Fix pack 20.0.0.3	20 March 2020	
▫	Fix pack 20.0.0.2	21 February 2020	
▫	Fix pack 20.0.0.1	24 January 2020	

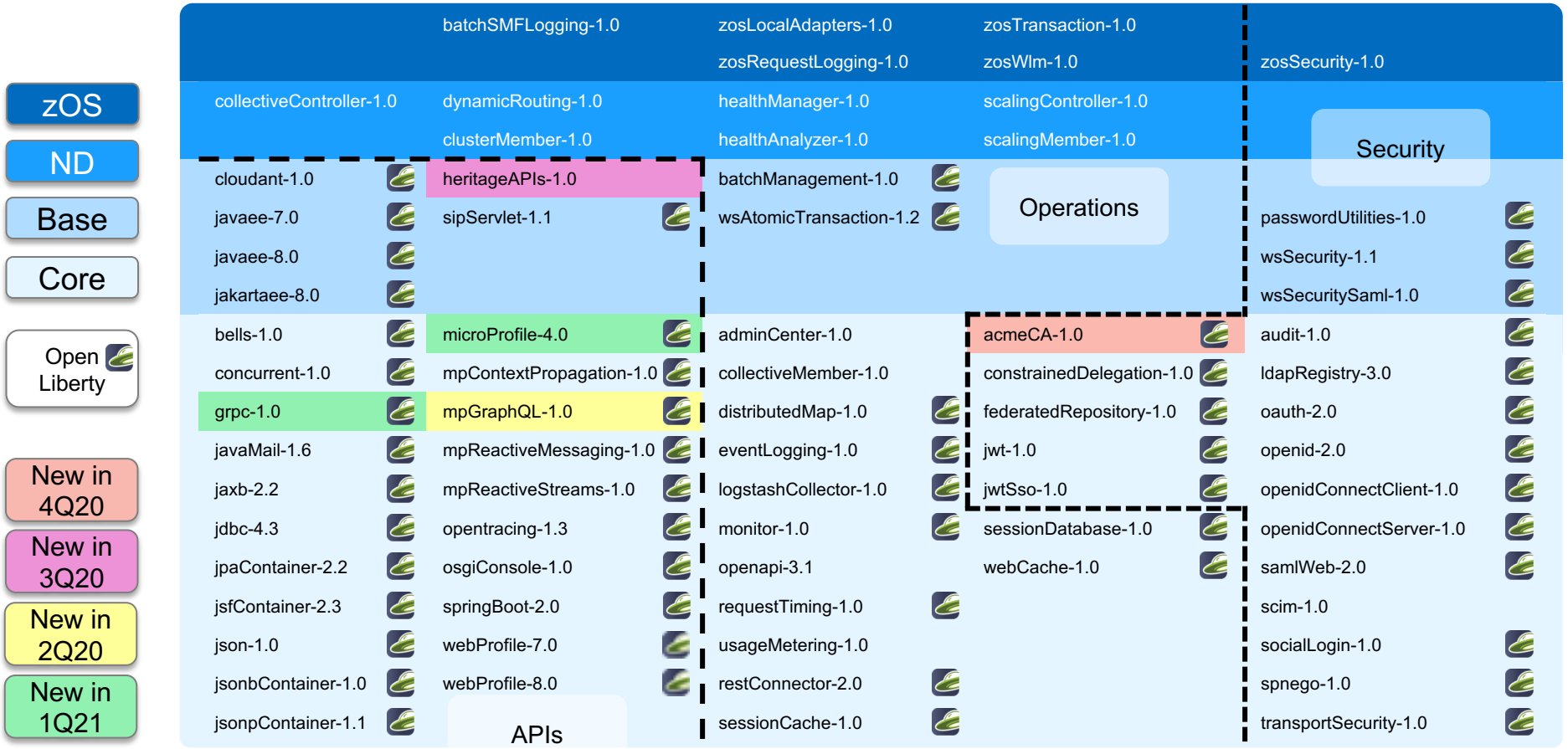
https://www.ibm.com/support/pages/latest-fix-packs-websphere-application-server

Java EE Compliance

Liberty had Java EE and Jakarta EE compliance with

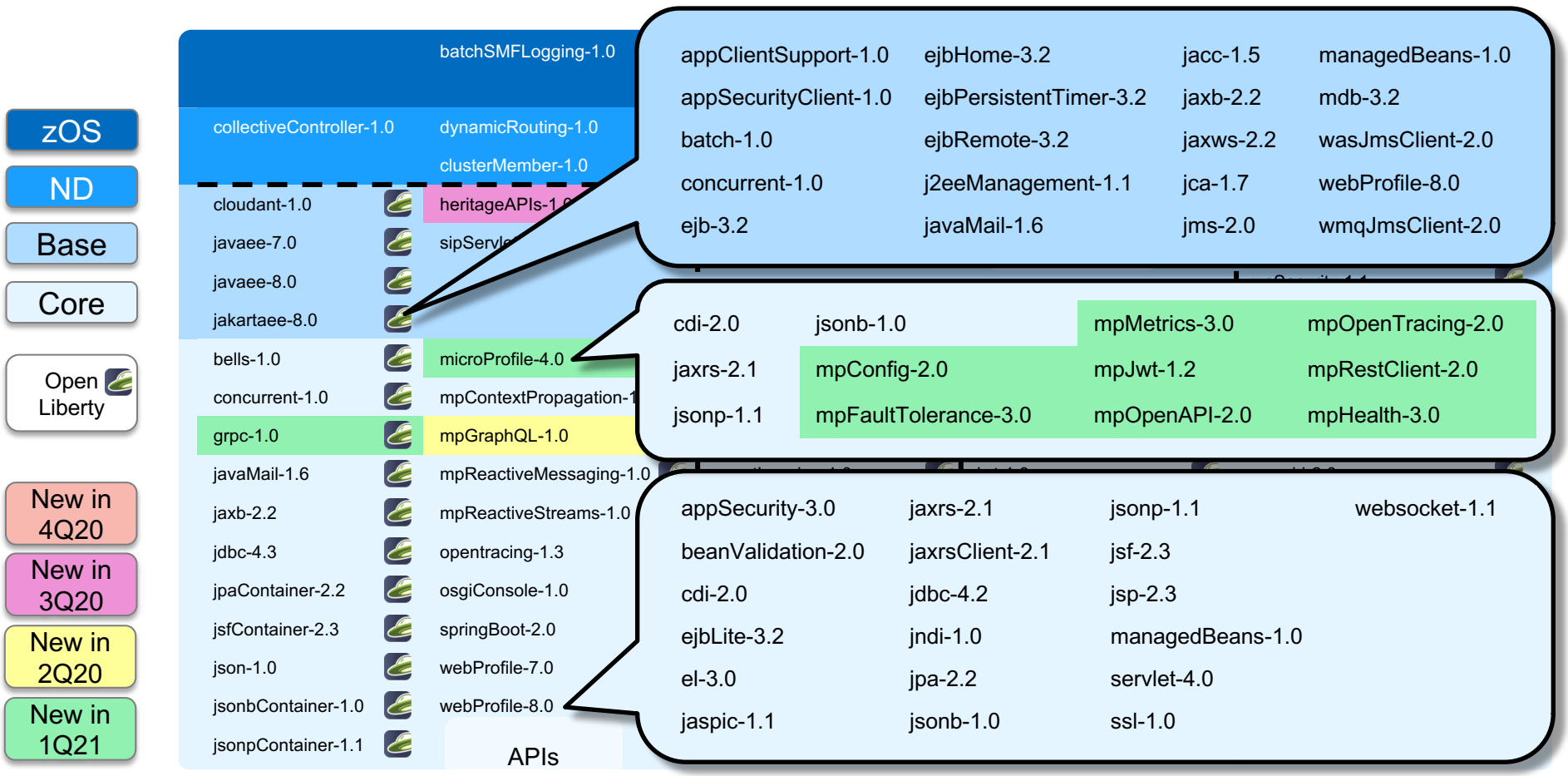
- Java EE 6 web profile
- Java EE 7 full platform
- Java EE 8 full platform
- Jakarta EE 8 full platform

Periodic Table of Liberty (21.0.0.3)



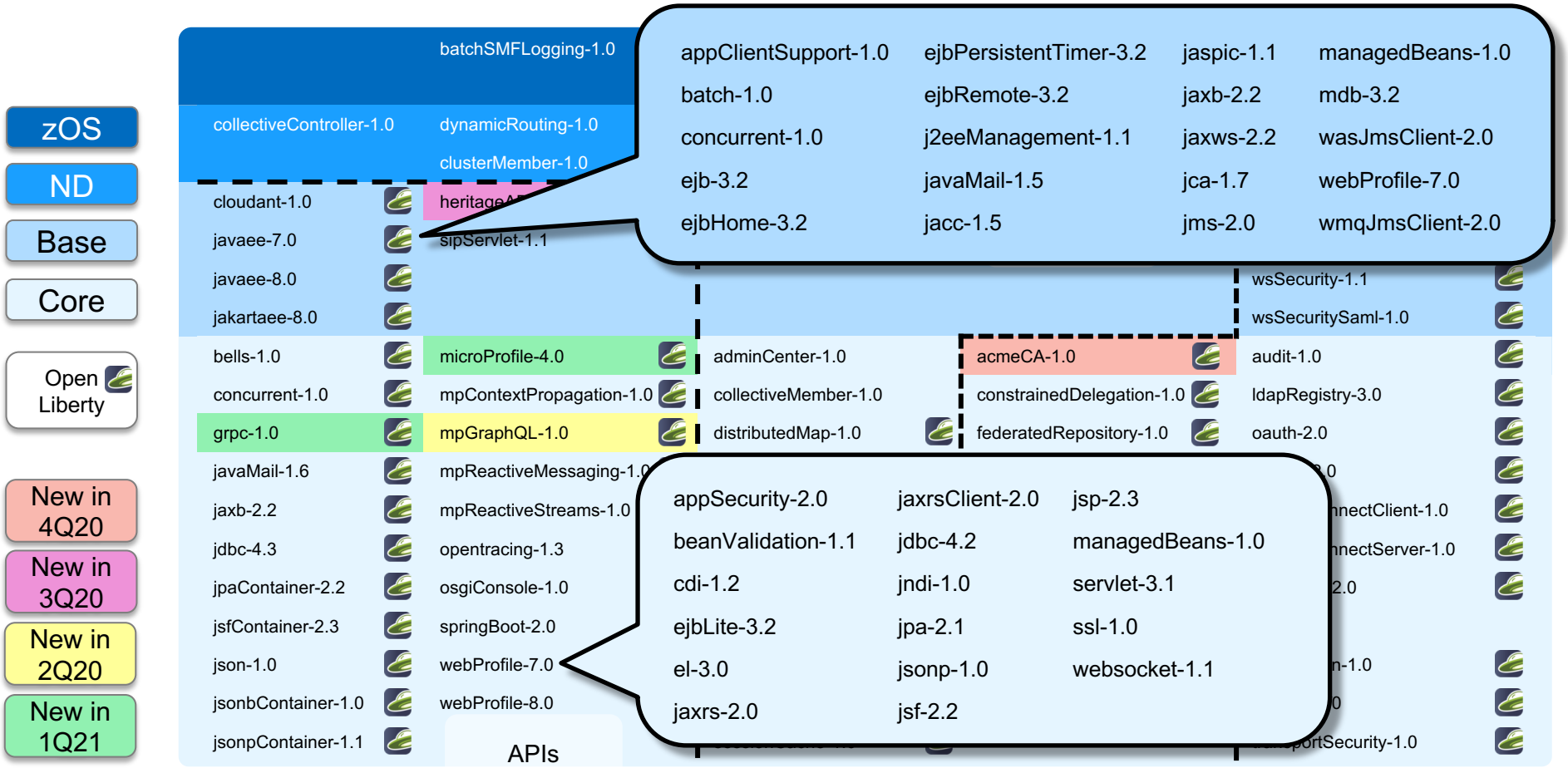
Revision Date 06/2016

Periodic Table of Liberty (21.0.0.3)



Revision Date 06/2016

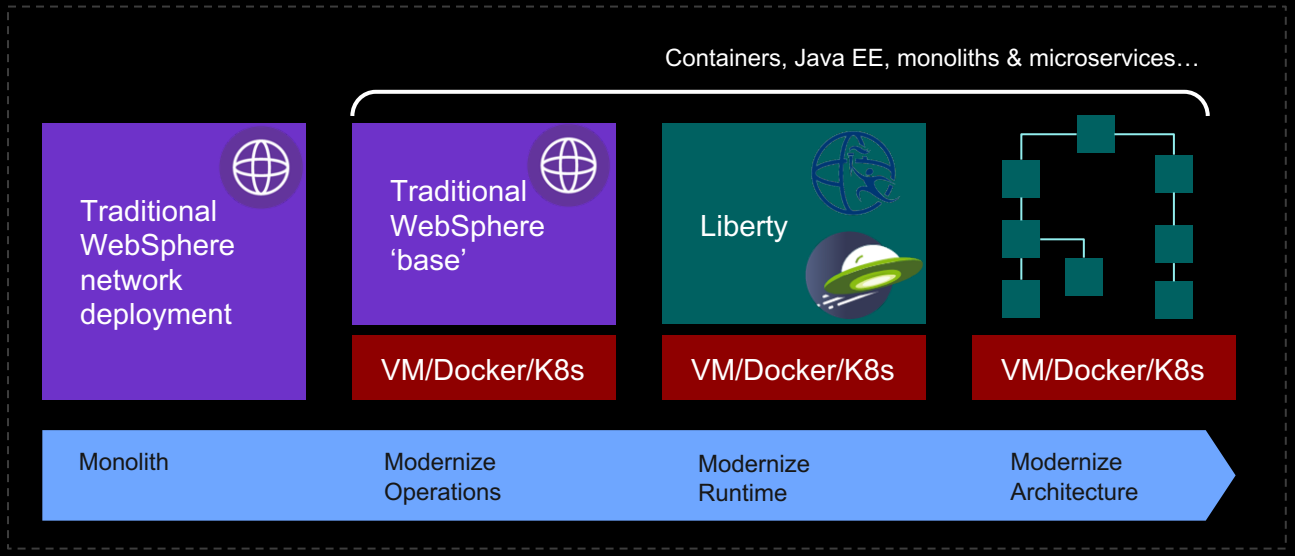
Periodic Table of Liberty (21.0.0.3)



Revision Date 06/2016

IBM Cloud Transformation Advisor

Accelerates the modernization journey by quickly discovering and analyzing on-premise Java EE and/or messaging workloads in the enterprise to help in determining and executing the optimum modernization steps for each. Leverages practitioner expertise to provide recommendations, detailed reports, and artifacts.



Optimum modernization depends on workload needs! http://ibm.biz/cloudta

Workloads:

Java EE

- WebSphere Application Server
- Oracle WebLogic
- Red Hat JBoss
- Apache Tomcat

Integration

- IBM MQ
- App Connect Enterprise (IIB)

Open Source SDK for other extenders:

https://github.com/IBM/transformation-advisor-sdk

Modernizing Applications for Liberty

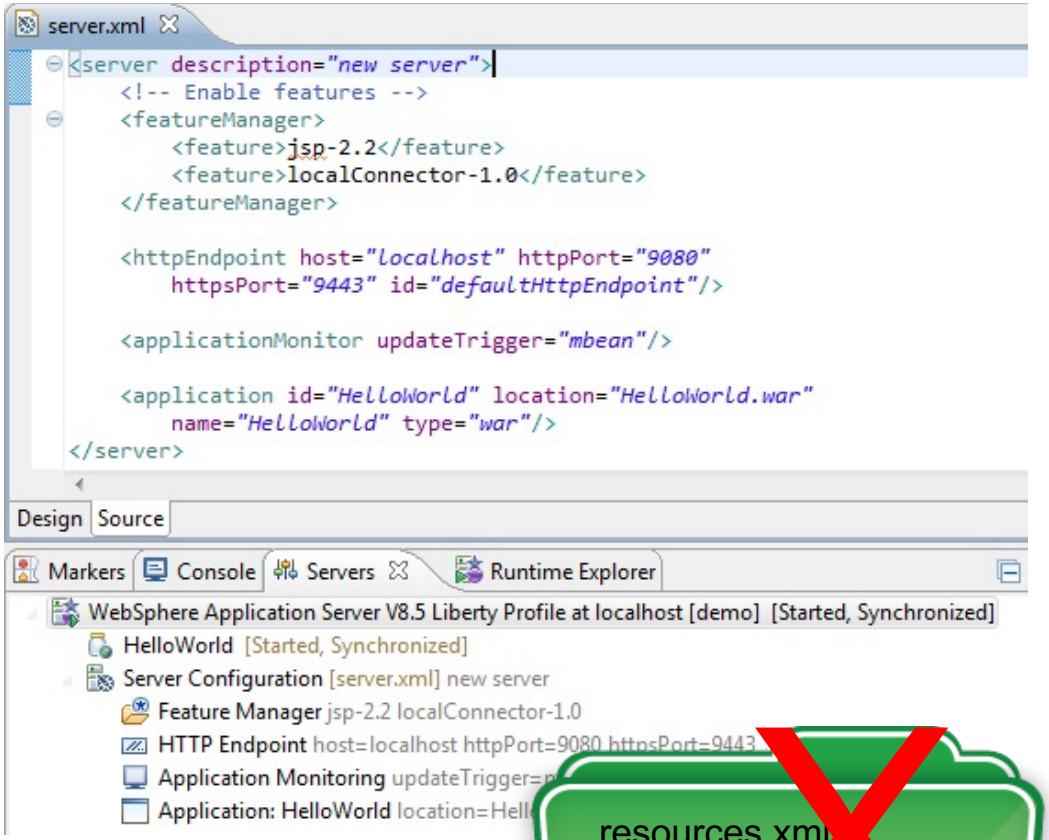
Transformation Advisor collects data using the Migration Toolkit for Application Binaries (binary scanner)

With a simple scan you can get the information you need to move applications from traditional WebSphere to Liberty

- Detailed analysis reports – shows all programming model differences
- Inventory reports – shows the contents of your application
- Technology evaluation reports – shows what Java EE technologies your application uses and what application server supports it

Simplified Liberty Server Configuration

- Simplest case: One XML file for all server config
- Editable within the workspace
- Exportable, shareable, versionable



No need for Admin Console, wsadmin, or extended EARs

Moving traditional WebSphere Configuration to Liberty

New
Dec 2019

Migration Toolkit for Application Binaries (binary scanner)

- Scans traditional WebSphere configuration in the context of an application deployment and produces Liberty server.xml for
  - JDBC
  - JMS
  - SIB
  - JCA Auth Alias
  - JVM Options
  - JNDI strings
  - Session Management
  - Application security
  - LDAP – standalone user registry
  - LDAP – federated user registry
  - User and group role mappings
  - SSL
- Creates a Liberty feature list for single EAR and WAR files.

Agenda

- v9, v8.5, v8.0 and v7.0 Feature Highlights
- High level view of Migration changes
- Modernizing to Liberty
- **Changes per version**
  - Changes in v9.0
  - Changes in v8.5
  - Changes in v8.0
  - Changes in v7.0
  - Changes in v6.1
  - Changes in v6.0

Changes in v9.0

New June 2016

- Administration changes
  - Default Jython version
  - New default coregroup wiring protocol
  - New java extensions directory - **$WAS_HOME/javaext**
  - Other miscellaneous changes

- Development changes
  - Development tool changes
  - Java SE 8 upgrade
  - Java EE 7 upgrade
    - CDI, JAX-RS, JPA implementation change
  - API removals
  - API deprecations

Changes in v8.5

- Administration changes
  - Some new required ports
  - A number of minor default setting changes
    - Information provided in the v8.5 IBM Documentation
    - https://www.ibm.com/docs/en/was-nd/8.5.5
- Development changes
  - Development tool changes
  - Java7 upgrade – Java6 is the default
    - Breaking changes: (AWT, Internationalization, IO, JAXP, Language, Networking, Text and Utilities)
  - JPA (2)
    - Custom settings are provided to provide compatibility
- Conversion of existing applications to Liberty

Changes in v8.0

- Administration changes
  - Installation changes
  - Centralized Install Manager
  - Install Factory alternative
  - WebServer Plug-in installation and configuration
  - Java Garbage collection and dump format changes
  - Security default changes
  - Other miscellaneous changes

- Development changes
  - Development tool changes
  - JEE 1.6
  - WebSphere API changes

Changes in v7.0

- Administration changes
  - SessionInitiationProtocol(SIP) Migration Considerations
  - zOS Migration tool
  - Administration script required changes
  - Port usage
  - Security Migration considerations
  - Mixed version considerations

- Development changes
  - Development tool change
  - JRE 6 impacts
  - JEE 5 impacts
  - WebSphere removed features
  - Increased usage of Open Source implementations included in WAS

Changes in v6.1

- Administration changes
  - Administration script required changes
  - zOS Migration tool
  - Install response file format changes
  - Port usage
  - Profile directory structure
  - New administrative tool IDE
  - Migration and Feature Packs

- Development changes
  - Development tool change
  - JRE 5 impacts
  - WebSphere changes and removed features

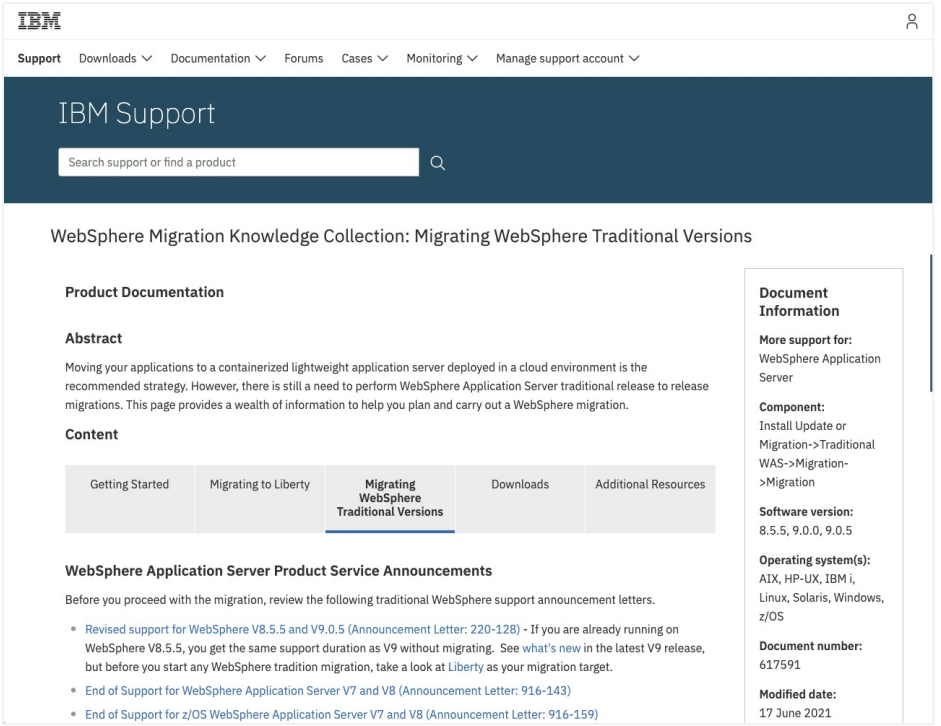
Changes in v6.0

- Administration changes
  - Administration script required changes
  - Port usage
  - Profiles
  - JMS engine redesign
  - CoreGroup considerations

- Development changes
  - Development tool change
  - J2EE 1.4 impacts
  - WebSphere API migration details

Enablement: IBM Migration Knowledge collection

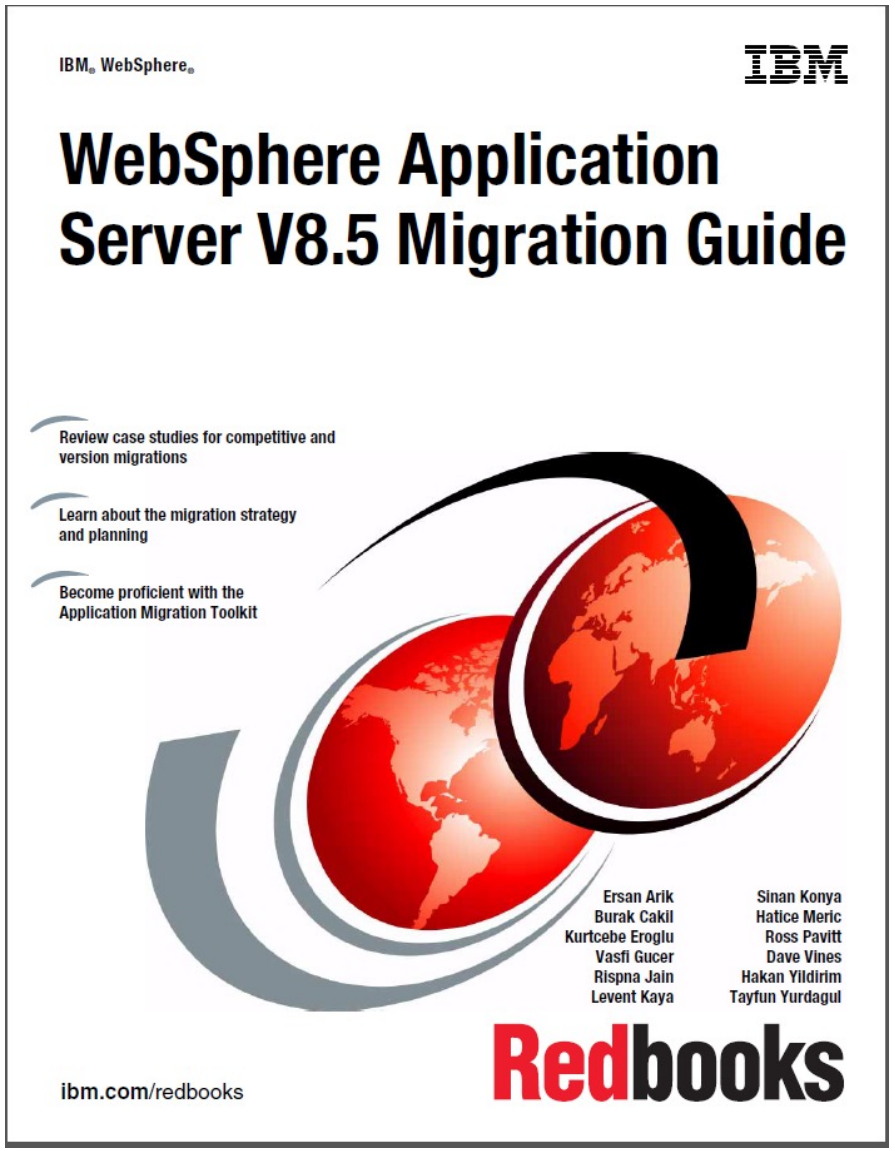
- This information and more is available online!
- General planning with detailed notes and WebSphere AppServer version specific information
- Updated with timely information
- Google:
  “websphere application server migration”



https://www.ibm.com/support/pages/websphere-migration-knowledge-collection-migrating-websphere-traditional-versions

Latest Redbook!

- Covers WAS version to version
- Covers migration from other Application Servers
  - Apache Tomcat
  - JBoss
  - Oracle AS
  - WebLogic
- SG24-8048
  - www.redbooks.ibm.com/abstracts/sg248048.html



References

Migration information

- WAS Migration Toolkit download page:
  - https://www.ibm.com/support/pages/websphere-migration-knowledge-collection-downloads
- IBM Migration Assist from WebSphere Level 2 Support Team
  - https://www.ibm.com/support/pages/node/462393 - 5
- IBM Software Accelerated Value program
  - http://www-01.ibm.com/software/support/acceleratedvalue/index.html
- WAS 8.5 WebSphere Migration Guide
  - http://www.redbooks.ibm.com/redpieces/abstracts/sg248048.html

- Talk with your IBM representative !

References - Planning

- Supported hardware and software information
  - https://www.ibm.com/cloud/websphere-application-server
  - http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html
- IBM Support Policies
  - Revised support for WebSphere V8.5.5 and V9.0.5 (Announcement Letter: 220-128)
  - http://www-1.ibm.com/support/docview.wss?uid=swg21256700
  - http://www-01.ibm.com/software/support/lifecycle/lc-policy.html
  - http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&appname=iSource&supplier=877&letternum=ENUSZP13-0568
- Installation Manager and Managing Repositories
  - https://www.ibm.com/support/pages/installation-manager-and-packaging-utility-download-documents
  - http://www.ibm.com/support/docview.wss?uid=swg27023967&aid=1
- Web Server plug-in technotes and Merge tool
  - http://www-1.ibm.com/support/docview.wss?uid=swg21160581
  - http://www-01.ibm.com/support/docview.wss?uid=swg21139573
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=iwspi-configuring-simple-load-balancing-across-multiple-application-server-profiles

References - Planning

- WebSphere supported Specification levels and pointers to JEE specifications
  - https://www.ibm.com/docs/en/was-nd/8.5.5?topic=overview-specifications-api-documentation
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=waso-java-ee-7-in-websphere-application-server-traditional
- WebSphere AppServer API Deprecations, removals and stabilizations
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=traditional-deprecated-stabilized-removed-features
- Changes in Default behavior
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=traditional-default-value-behavior-changes-from-previous-releases
- WebSphere Application Server V8.5 Concepts, Planning, and Design Guide
  - http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248022.html?Open
- Migrating WebSphere Compute Grid or Feature Pack for Modern Batch
  - https://www.ibm.com/docs/en/was-zos/8.5.5?topic=mpz-migrating-compute-grid-feature-pack-modern-batch-zos-systems
- IBM Media Center – WAS and Liberty Channel
  - https://mediacenter.ibm.com/channel/t/33964822

References – Planning for Liberty

- Open Liberty
  - https://openliberty.io/
- Open Liberty Docs
  - https://openliberty.io/docs
- Open Liberty Guides
  - https://openliberty.io/guides
- WebSphere Liberty Documentation
  - https://www.ibm.com/docs/en/was-liberty/base?topic=SSEQTP_liberty/as_ditamaps/welcome_liberty.html
- Why choose Liberty
  - https://ibm.biz/6ReasonsWhyLiberty
- Choose the right Java runtime
  - https://ibm.biz/ChooseJavaRuntime
- WebSphere and Liberty Community Spotlight – IBM Expert TV
  - https://ibm.biz/LibertyTV
- App Transformers – IBM Expert TV
  - http://ibm.biz/IBMExpertTV-AppTransformers

References - Training

- IBM Education Assistant
  - https://www.ibm.com/docs/en/rtw/8.7.0?topic=troubleshooting-education-assistant
  - https://mediacenter.ibm.com/channel/t/33964822
- WebSphere Application Server V9 Update
  - https://mediacenter.ibm.com/media/WebSphere+Application+Server+V9+technical+update/0_ttxciunh/33964822
- WebSphere Application Server V8.5.5 Technical Overview
  - http://www.redbooks.ibm.com/redpapers/pdfs/redp4855.pdf
- WebSphere Application Server: New Features in V8.5.5
  - http://www.redbooks.ibm.com/redpapers/abstracts/redp4870.html?Open
- Properties based configuration
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=wsadmin-using-properties-files-manage-system-configuration
  - http://www.ibm.com/support/docview.wss?uid=swg27039420

References - Configuration Migration

- What about my modernized application's configuration and tuning?
  - https://techtv.bemyapp.com/#/conference/60106cba71c1f8001b6e264b
- Tuning the application serving environment
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=performance-tuning-application-serving-environment
- WebSphere Application Server V8.5 Migration Guide
  - http://www.redbooks.ibm.com/abstracts/sg248048.html
- Changing host names and migrating profiles
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=servers-changing-node-host-names
- Migrating cells to new host machines using the command-line tool
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=mpc-migrating-cells-new-host-machines-using-command-line-tool
- Migration – Application Installation problems
  - http://www-01.ibm.com/support/docview.wss?uid=swg27008724&aid=13
- WAS z/OS Migration Performance Study
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101589

References - Development

- WebSphere Application Server Migration Toolkit
  - https://www.ibm.com/support/pages/websphere-migration-knowledge-collection-downloads
  - https://www.ibm.com/docs/wamt
- WebSphere Application Server Developer Tools for Eclipse
  - https://www.ibm.com/docs/wasdtfe?topic=installing-websphere-application-server-developer-tools-eclipse
- Java Compatibility
  - Java 11 - https://docs.oracle.com/en/java/javase/11/migrate/index.html (Liberty only)
  - Java 8 - http://www.oracle.com/technetwork/java/javase/8-compatibility-guide-2156366.html
  - Java 7 - http://www.oracle.com/technetwork/java/javase/compatibility-417013.html
  - Java 6 - http://www.oracle.com/technetwork/java/javase/compatibility-137541.html

References - Development

- Web services migration best practices
  - https://www.ibm.com/docs/en/was/9.0.5?topic=services-web-migration-best-practices
- Migration from Apache SOAP to web services
  - https://www.ibm.com/docs/en/was/9.0.5?topic=mws-migrating-apache-soap-web-services-jax-rpc-web-services-based-java-ee-standards
- JavaServer Pages specific Web container custom properties
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=configuration-javaserver-pages-custom-properties
- JMS Listener to Message Driven Bean migration
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=cjrmp-migrating-listener-port-activation-specification-use-mq-messaging-provider
- Using other web service engines in WAS
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=dws-using-third-party-jax-ws-web-services-engine
- JSF Migration
  - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=components-javaserver-faces-migration
- Best Practices for Integrating Open Source Software
  - http://www-01.ibm.com/support/docview.wss?uid=swg21639407

References - Operations

- UrbanCode Deploy
  - https://www.urbancode.com/products/urbancode-deploy/
  - https://www.urbancode.com/plugin/websphere-application-server-configure/
- Techniques for Managing Large WebSphere Installations
  - http://www.redbooks.ibm.com/redbooks/pdfs/sg247536.pdf
- IBM Java Runtimes
  - https://www.ibm.com/support/pages/semeru-runtimes-getting-started/
  - https://www.ibm.com/docs/en/sdk-java-technology/8
- wsadmin Primer
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101014
- JACL to Jython conversion assistant
  - http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012144

v9.0 Changes

Default/Behavior changes

New June 2016

Default value and behavior changes from previous releases of WebSphere Application Server traditional

- High availability and workload management
  - IBM_CS_OOM_ACTION
  - IBM_CLUSTER_REUSE_ORIGINAL_IOR
  - Proxy server setting
  - V9.0 coregroup member default protocol has changed
    - coregroup property: IBM_CS_WIRE_FORMAT_VERSION = 6.1.0
    - coregroup property: IBM_CS_HAM_PROTOCOL_VERSION = 6.0.2.31
    - Older releases also support this newer protocol
    - All servers in cell must be running the same protocol
- Resources
  - JPA data source error handling

Default/Behavior changes

New June 2016

Default value and behavior changes from previous releases of WebSphere Application Server traditional

- Web server plugin
  - esiEnable
  - KillWebServerUponParseErr
  - StrictSecurity
- Security
  - sslProtocol
  - com.ibm.websp
- Servlet custom properties
  - com.ibm.ws.webcontainer.RedirectWithPathInfo is ignored by the Servlet 3.1. To use this property and to revert to some changes to the Servlet 3.0 behavior, use the com.ibm.ws.webcontainer.servlet30compatibility property.

Java changes

New June 2016

- New java extension directory - **$WAS_HOME/javaext**
  - Independent of java directory
  - Contains iwsorbutil.jar

Jython

Updated
July 2016

- wsadmin tool default changes:
  - Default language is now Jython
  - Jacl is deprecated in v9.0 and is no longer the default language
    - Use “-lang jacl” to use Jacl.
  - Default Jython version is 2.7
  - Enable old default Jython version behavior with
    - command line ➔ -usejython21 true
    - wsadmin.properties ➔ com.ibm.ws.scripting.usejython21=true
  - For details on syntax and behavioral changes see this link: https://www.ibm.com/docs/en/was/9.0.5?topic=jython-v27-behavior-changes

Port usage

New June 2016

- v9 uses the same ports as V8.5.5
  - For complete port information, see
    - Network deployment: https://www.ibm.com/docs/en/was-nd/9.0.5?topic=cps-port-number-settings-1
    - Base: https://www.ibm.com/docs/en/was/9.0.5?topic=settings-port-number

Server Type	v5.x	v6.0	v6.1	v7.0	v8.0	v8.5	V9.0
Application (base)				21	21	21	21
Application (ND standalone)	16	17	18	21	21	23	23
Application (ND federated)				16	16	18	18
DMgr	11	17	17	11	11	16	16
Node Agent	11	17	17	11	11	14	14

Port usage summary

New June 2016

- Endpoint changes since V7.0

  (including default port assignments by server type):

  - V7.0 to V8.0 ➔ No changes.

  - V8 to V8.5.5 ➔ See table:

Server Type ➔ / ↓ EndPoint Name	Deployment Manager	Node Agent	Application Server	Job Manager
XDAGENT_PORT	7060	7061	--	--
STATUS_LISTENER _ADDRESS	9420	--	--	9425
OVERLAY_UDP _LISTENER_ADDRESS	11005	11001	11003	--
OVERLAY_TCP _LISTENER_ADDRESS	11006	11002	11004	--

  - V8.5.5 to V9.0 ➔ No changes.

Development considerations

WAS Developer Tools for Eclipse (WDT) & RAD

New
June
2016

The latest WDT available on Eclipse Marketplace supports traditional WebSphere V9.

The following enhancements are in WDT:

- New support in WDT for Docker, API discovery, Java EE 7
- The new Batch programming model is supported in WDT. The WebSphere Batch programming model has RAD-only support.
- SIP 1.1 is supported in WDT & Liberty. SIP 1.0 has RAD-only support.
- RAD added support for Mac.
- RAD no longer supports WebSphere V6.1

Java SE 8

Moving to Java SE 8

New June 2016

- The Compatibility Guide for JDK 8 provides details of the changes in Java
  - http://www.oracle.com/technetwork/java/javase/8-compatibility-guide-2156366.html
- Changes are organized in terms of
  - Binary Compatibility
  - Source Compatibility
  - Behavioral Compatibility
  - Java Class Files
  - Incompatibilities between Java SE 8 and Java SE 7
  - Incompatibilities between JDK 8 and JDK 7
  - Features Removed from Java SE 8
  - Features Removed from JDK 8
  - Deprecated APIs
- Possible application impacts are listed on the following pages with  meaning that a corresponding rule has been added to the migration toolkit.

Moving to Java SE 8

New June 2016

- AWT
  - Behavior change in exceptions when setting AWT focus traversal keys
    - In Java 8, the java.awt.Component.setFocusTraversalKeys() and the java.awt.KeyboardFocusManager.setDefaultFocusTraversalKeys() methods throw ClassCastException instead of IllegalArgumentException if any passed keystroke object is not an AWTKeyStroke object.
- Internationalization
  - The mechanism to select a locale service provider changed
    - Java 8, the mechanism to select a locale service provider changed. A new method in the LocaleServiceProvider class allows implementations to determine whether the given locale is supported.
  - Behavior change in month name formatting for some languages
    - In Java 8, when formatting date-time values using the DateFormat or SimpleDateFormat classes, context-sensitive month names are supported for languages that have different date formatting and standalone forms of month names. You might see a difference in the month name returned in strings formatted by the DateFormat or SimpleDateFormat classes or by methods on the DateFormatSymbols class.
- JAXP
  - Differences in class loading for JAXP service providers
    - Java 8 includes Java API for XML Processing (JAXP) 1.6, which handles class loading for service providers differently than previous versions.

Moving to Java SE 8

New
June
2016

- Language

java.lang.Thread.stop(java.lang.Throwable) is disabled

- In Java 8, the java.lang.Thread.stop(java.lang.Throwable) method is disabled. It was previously deprecated.

The TypeVisitor interface has been updated

- Java 8 added a new method to the javax.lang.model.type.TypeVisitor interface which will affect classes that implement TypeVisitor. A new IntersectionType is being introduced. A new TypeKind.Intersection enum constant was also added.

Behavior change in the construction of dynamic proxy classes

- In Java 8, calling java.lang.reflect.Proxy(InvocationHandler) with a null parameter throws a NullPointerException. Prior to Java 8, the constructor returns a proxy, but then any method call to that proxy would throw a NullPointerException.

Behavior change in new instance creation for non-public interfaces

- In Java 8, a code change is required to create a proxy instance for non-public interfaces located in a different package using the Proxy.getProxyClass and Constructor.newInstance methods.

Moving to Java SE 8

New
June
2016

- Management

  MBean and MXBean interfaces must be public

  - Java 8 enforces the requirement that MBean and MXBean management interfaces be public. The specification states that management functionality cannot be exposed with non-public interfaces, but this requirement was not enforced in Java 7 and prior versions.

- Math

  Behavior change in the BigDecimal stripTrailingZeros method for a zero value

  - Java 8 introduces a behavior change on the java.math.BigDecimal stripTrailingZeros method when it operates on a zero value with a nonzero scale. In prior versions, no zeros were stripped in this case.

Moving to Java SE 8

New June 2016

- Network

DatagramPacket constructor with SocketAddress no longer throws SocketException

- In Java 8, java.net.DatagramPacket constructors that accept a java.net.SocketAddress argument were changed to remove the SocketException declaration. This can cause a compilation error if the constructors are within a try block that catches either a java.net.SocketException or its superclass java.io.IOException.

– Changes in WWW-Authenticate Response Header

- In previous Java releases, the HttpURLConnection Digest Authentication implementation incorrectly quoted some values in the WWW-Authenticate Response Header. Those values are no longer quoted in Java 8.

– Default socket permissions have changed

- In previous Java releases, all code was able to bind any socket type to any port number greater than or equal to 1024. It is still possible to bind sockets to the ephemeral port range which varies on each system. The new behavior change is that binding sockets outside of the ephemeral range requires an explicit security permission.

– Removal of ftp from the list of required protocol handlers

- The ftp protocol was deleted from the list of protocol handlers that are guaranteed to be present in Java SE. The protocol handler was not actually removed, but it is not required to be provided.

Moving to Java SE 8

New
June
2016

- Security
  - Certificates are blocked if they contain RSA keys of less than 1024 bits in length.
    - In Java 8 with this new key size restriction, programs using X.509 certificates based on RSA keys less than 1024 bits will encounter compatibility issues with certification path building and validation. This key size restriction also impacts JDK components that validate X.509 certificates such as signed JAR verification, SSL/TLS transportation, and HTTPS connections.
- Text

Behavior change in rounding in the NumberFormat and DecimalFormat format methods

- In Java 8, the rounding behavior of the NumberFormat and DecimalFormat format methods changed to match the rounding of the binary representation of the number.

- Utilities

New methods in java.util.concurrent.ConcurrentHashMap

- In Java 8, the ConcurrentHashMap class introduced over 30 new methods. If you extend the java.util.concurrent.ConcurrentHashMap class, your class might need changes.

Behavior change in most Collection.removeAll and Collection.retainAll implementations

- Prior to Java 8, most implementations of Collection.removeAll(Collection) and retainAll(Collection) returned false and ignored a null parameter if the collection itself was empty. In Java 8, collections throw a NullPointerException when a null parameter is provided.

Moving to Java SE 8

New
June
2016

- com.sun
  - A number of com.sun packages were added to the list of restricted packages in JDK 8. com.sun packages are not intended for application use.
    - com.sun.media.sound
    - com.sun.corba.se
    - com.sun.mirror
    - com.sun.security.auth.callback.DialogCallbackHandler

Java EE 7 behavior changes

CDI 1.2 behavior differences

New
June
2016

- Java EE 6 CDI 1.0 was based on Web Beans 1.0 spec (JSR 299)
- Java EE 7 CDI 1.2 is based on the Weld implementation
- See **Contexts and Dependency Injection 1.2 behavior changes**
- Most of the differences are captured by the Migration Toolkit and the Binary scanner.
- Using Liberty, you can continue to use CDI 1.0 with the Java EE 6 platform.
- Using traditional WebSphere V9, you must upgrade.

CDI 1.2 behavior changes

New June 2016

- An interceptor for lifecycle callbacks may only declare interceptor binding types that are defined as @Target(TYPE)
- CDI recognizes implicit bean archives
- CDI scans for implicit beans when there is no beans.xml file
- Check for a behavior change in the InjectionPoint getAnnotated method
- Check for a valid schema in beans.xml
- Check for the enablement of interceptors, decorators and alternatives in other JAR files
- Classes that use both the Specializes and Alternative annotations are not injected into other modules
- Do not use the OpenWebBeans schema for beans.xml
- Producer fields on session beans must be static
- The openwebbeans.properties file is not used
- Transient fields in session-scoped beans cannot fail over successfully

EL 3.0 behavior differences

New
June
2016

Expression Language 3.0 has one change that might cause applications to break

- Behavior change in coerceToType method with null parameter

JAX-RS 2.0 behavior changes

New
June
2016

- Java EE 6 JAX-RS 1.1 was based on Apache Wink (JSR 311)
- Java EE 7 JAX-RS 2.0 is based on the CXF implementation
- See JAX-RS 2.0 behavior changes
- Most of the differences are captured by the Migration Toolkit and the Binary scanner.
- Using Liberty and traditional WebSphere V9, you can continue to use JAX-RS 1.1 with the rest of the Java EE 7 platform.

JAX-RS 2.0 behavior changes

New June 2016

- @Local JAX-RS interfaces must be implemented
- Configuration is required to use SSL in JAX-RS 2.0
- org.codehaus.jackson packages are not available
- Packaging Apache Wink APIs with your application might require application changes
- The Apache Wink APIs are not available
- The Apache Wink Client APIs are not available
- The com.ibm.websphere.jaxrs.server.IBMRestFilter class is no longer supported
- The org.apache.wink.client.handlers.LtpaAuthSecurityHandler class is no longer supported
- The org.apache.wink.common.model.atom package is not available
- The org.apache.wink.common.model.multipart package is not available
- Use the isReadable and isWriteable methods to check the media type

JMS Client 2.0 behavior differences

New
June
2016

- Check for a behavior change on message priority and the NoLocal attribute
- Check for a behavior change on setClientID and createDurableSubscriber methods

Servlet 3.1 behavior changes

New June 2016

- Check for a behavior change in the processing of the absolute-ordering element
- Check for a behavior change on asynchronous servlets
- Check for a behavior change on the getServerInfo method
- Check for a behavior change on the sendRedirect method
- Check for a behavior change on the ServletContextListener interface
- Check for a behavior change on the setComment method
- Check for a behavior change regarding duplicate elements in web descriptors
- Check for a behavior change with resource reference injection target merging
- Check for a behavior change with URL pattern mapping

JPA 2.1 behavior changes

New June 2016

- Java EE 6 JPA 2.0 is based on OpenJPA
- Java EE 7 JPA 2.1 is based on EclipseLink
- See Java Persistence API 2.1 behavior changes
- Many differences are captured by the Eclipse-based Migration Toolkit which also provides some quick fixes. The focus of the migration toolkit is on JPA annotation code style rather than the ORM files.
- The Binary scanner detects enhanced JPA classes and some of the persistence.xml issues. Use the source scanner if you plan on moving to JPA 2.1.
- For Liberty and traditional WebSphere V9, you can continue to use JPA 2.0 with the rest of the Java EE 7 platform. This is recommended if you use OpenJPA functionality.

JPA 2.1 behavior differences

New June 2016

- All entities must have a primary key
- Annotated getter methods must have a setter method
- Attributes with automatically generated values require configuration
- Disable the persistence unit second-level cache
- Do not use OpenJPA providers in the persistence.xml file
- ElementCollection annotations must be accompanied by a defined Column annotation
- Embeddable classes cannot have an Id annotation when referenced by an EmbeddedId annotation
- Embedded classes must be annotated as embeddable
- Entity objects with constructors must also have a default constructor
- java.util.Locale attributes must be converted

JPA 2.1 behavior differences

New June 2016

- JoinColumn annotations must be used with relationship mappings
- Mapping files are not processed during OpenJPA to EclipseLink migration
- OpenJPA and WebSphere JPA configuration properties must be migrated
- OrderColumn annotations are not supported on Set attributes
- org.apache.openjpa packages are not available
- Private accessor methods must have a Transient annotation
- Remove the Temporal annotation for some java.sql attributes
- Replace OpenJPA @PersistentCollection annotation with @ElementCollection and @Column
- Replace the Temporal annotation with a Converter annotation for some java.sql attributes
- The openjpa.jdbc.Schema configuration property must be migrated to the mapping file

JPA 2.1 behavior differences

New
June
2016

- The openjpa.LockManager configuration property must be migrated
- Unannotated collection attributes require a Transient annotation
- Unannotated entity attributes require a Transient annotation
- Validate IN expression syntax with a collection-valued input parameter

V9 API Removals

New June 2016

- Apache HTTP client API was removed
  - Can no longer be access by application code
  - Package it with your application if needed
- Service Component Architecture (SCA)
- Communications Enabled Applications (CEA) REST interface provided by system application commsvc.ear was removed
- Common Event Infrastructure (CEI) API was removed
- The JSF SunRI engine was removed
  - Package it with your application if needed

V9 API Deprecations

New June 2016

- CommonJ Timer and Work Manager APIs
  - Use Java EE 7 Concurrency Utilities instead
  - See Examples to migrate to EE Concurrency from Asynchronous beans and CommonJ
    - https://www.ibm.com/docs/en/was-nd/9.0.5?topic=mc-examples-migrate-ee-concurrency-from-asynchronous-beans-commonj
- Optional Java EE 7 technologies:
  - Asynchronous Beans
  - Enterprise JavaBeans (EJB) entity beans
  - Java API for XML-based RPC (JAX-RPC)
  - Java API for XML Registries (JAXR)
  - Java EE Application Deployment

v8.5 Changes

Default/Behavior changes

- Java virtual machine custom property: com.ibm.websphere.logging.useJULThreadID
  - Now is “false”, prior it was “true”
- HTTP transport channel settings "Discrimination failed" exception in the HTTP Channel
  - Response code is “500”, prior it was “403”
- WebServer plugin-cfg.xml file
  - IgnoreAffinityRequests parameter
    - Now is “false”, prior it was “true”
  - Maximum buffer size used when reading HTTP request content PostBufferSize element
    - Is now “0”, prior it was “64”
  - In v8.5.5.x ONLY – Change when failure to create an HTTPS connection
    - If the web server plug-in attempts create an HTTPS connection but fails, it does **NOT** create a connection. Previous versions rolled over to HTTP connection silently.
    - https://www.ibm.com/support/pages/apar/PM96173

Port usage

Updated June 2016

- v8.5 uses more ports than some previous versions
  - Can be an impact to those that tightly control port access
  - Can also cause more port conflicts
  - See
    - https://www.ibm.com/docs/en/was-nd/8.5.5?topic=cps-port-number-settings-1

Server Type	v5.x	v6.0	v6.1	v7.0	v8.0	v8.5	v9.0
Application	16	17	18	18	18	20	20
DMgr	11	17	17	12	12	16	16
Node Agent	11	17	17	11	11	13	13

Converting to Liberty

Converting to Liberty

- Application conversion
  - Liberty is assumed to be used for developing new applications
  - Applications that run on Liberty will run on Full Profile unchanged
  - Some changes may need to be made if porting existing full profile applications to Liberty. E.g.:
    - Location of persistence.xml files can be anywhere in the war in Full profile, for Liberty must be one of the following
      - <war>/WEB-INF/classes/META-INF/
      - <war>/lib/<jar>/META-INF/
      - <ear>/<library directory/<jar>/META-INF/
    - 3rd party APIs in Liberty require additional configuration (e.g.)
      - <application id="ERWW_Lite" location="ERWW_Lite_EBA_PT.eba"
        name="ERWW_Lite" type="ear">
        <classloader allowedApiTypes="spec,ibm-api,third-party"/>
        </application>

Converting to Liberty…

- Application conversion…
  - Some JPA query validation may differ (e.g.)
    - @NamedQuery(name="commentCount", query="SELECT **count(*)** FROM CommentTab comment WHERE comment.user.userid =:userid") should be:
    - @NamedQuery(name="commentCount", query="SELECT **count(comment)** FROM CommentTab comment WHERE comment.user.userid =:userid")
  - Incorrect Servlet 3.0 namespace value is allowed on Full Profile, not Liberty
    - "http://java.sun.com/xml/ns/**j2ee**" should be:
    - "http://java.sun.com/xml/ns/**javaee**"
  - Some open source frameworks, such as the Apache Object Relational Bridge, make use of the internal undocumented and unsupported API. This is not supported on Liberty
  - OSGI application's WAB manifest file may require update(s)
    - OSGI applications running on Liberty may require additional package imports to be specified in the WAB MANIFEST.MF file which are not required for the same application running on tWAS
    - E.g. javax.naming, javax.sql and javax.xml.bind

Development considerations

WAS Developer Tools for Eclipse (WDT) V8.5 & RAD V8.5

RAD

Install: IM

WebSphere Integration
- Support for WAS v6.1
- Test Environments for WAS v7.0, v8.0, v8.5
- Portal Tools / Portal Svr supt
- Profile applications on WAS
- Deploy to WebSphere or Portal instances in IBM SBDT cloud
- IWD 3.1, 3.0

Problem Determination
- Code visualization - class, sequence and topic diagrams
- Static analysis (code review)
- Code coverage: optimize unit testing
- Profiling

Enterprise Connectivity
- J2C (EIS) tools
- CICS, and IMS Adapters
- Adapters for SAP, Siebel, JDE, Oracle, PeopleSoft

Team Productivity
- RTC integration
- Collaborative debug
- Collaborative code analysis

Programming Model Support
- SCA
- Java (WAS) Batch
- SIP/CEA
- XML (feature pack)

Extended Programming Model Support
- Advanced support for J2EE 1.4 and earlier:
  - EJB & Web Services deploy
  - DD editors
  - JAX-RPC
- Web:
  - Page designer
  - Struts, JSF support
  - iWidget support

WDT

Install: Eclipse update site

JEE Tools
- Advanced support for JEE 5+
- DD editors, enhanced project explorer, additional validation

Web Tools
- Advanced web development tools
- Rich page (WYSIWYG) editor for HTML, JSP
- Web 2.0 and Mobile support

WebSphere Integration
- Support for WAS v7.0, v8.0, v8.5
- Publish, start/stop the server
- Debug Jython/wsadmin scripts

WAS Extensions Support
- Binding and extension editors
- Support for non-spec extensions

OSGi Tools
- Full creation and editing support
- Blueprint editor and validation
- Visual Bundle Explorer

Liberty Integration
- Publish, start/stop the server
- Edit & manage server configuration

Eclipse (WTP, DTP)

Programming Model Support
- Basic creation, editing, and validation support for JEE applications:
  - Web, XML, JPA, EJB, EAR
- Database tools

WDT adds support for Mac, possibly with exception of WebSphere Integration

Revision Date 06/2016

Moving to Java SE 7 runtime

- AWT
  - The MouseEvent.getButton() method may return values outside of the [0-3] range in the plan
    - Previously, the MouseEvent.getButton method returned a value between 0 and 3 when the user clicked a button or used the scroll wheel. To accommodate newer models of mice with two scroll wheels, or four and five buttons, the method now returns a value from 0 to the number of buttons
  - Invoking Windows.setBackground may result in an UnsupportedOperationException exception
    - Legacy applications that apply a non-opaque background color to their frames may fail when the application is run on a system that doesn't support translucency effects
  - Toolkit.getPrintJob(Frame, String, Properties) now throws NullPointerException
    - Prior to this release, when invoking Toolkit.getPrintJob(Frame, String, Properties) in a headless environment, a HeadlessException is thrown instead of the specified NullPointerException
  - Various Toolkit methods now throw HeadlessException
    - Toolkit.isFrameStateSupported(int), and Toolkit.loadSystemColors(int[]), now throw a HeadlessException when used in a headless environment
  - The sun.awt.exception.handler System Property has Been Replaced with Official API
    - The sun.awt.exception.handler System Property is replaced with Thread.UncaughtExceptionHandler class

http://www.oracle.com/technetwork/java/javase/compatibility-417013.html#behavioral

Moving to Java SE 7 runtime

- Internationalization
  - Separation of User Locale and User Interface Locale
    - The default locale can be independently set for two types of uses: the format setting is used for formatting resources, and the display setting is used in menus and dialogs. The new Locale.getDefault(Locale.Category) method takes a Locale.Category parameter. Previous behavior can be restored
  - UTF-8 implementation is updated to conform to Corrigendum to Unicode 3.0.1
    - Previously, there were 5- and 6-byte forms of utf-8 sequences that were allowed. These are now rejected
- IO
  - java.io.File.setReadOnly and setWriteable Methods Have New Behavior
    - No longer set the DOS readonly attribute on directories. This means that these methods will fail, by returning false, if the file is a directory. To preserve the relationship with canWrite, the canWrite method returns true if the file is a directory.
- JDBC
  - New JDBC Methods, Including new Methods in Interfaces
    - There are new methods to support JDBC 4.1. This includes methods added to the java.sql.Connection, java.sql.Driver, javax.sql.CommonDatasource, and java.sql.Statement interfaces

http://www.oracle.com/technetwork/java/javase/compatibility-417013.html#behavioral

Moving to Java SE 7 runtime

- JAXP
  - The XSLTProcessorApplet Class is Removed
    - The XSLTProcessorApplet class is an application-level convenience class that had various problems. It has been removed.
  - JAX-WS Server Throws a SOAP Fault when it Encounters a DTD
    - SOAP Message Construct, the XML infoset of a SOAP message MUST NOT contain a document type declaration (DTD) information item.
- Language
  - The ThreadGroup.setMaxPriority Method Now Behaves as Specified
    - Previousy, the ThreadGroup.setMaxPriority did not behave as specified if the passed-in value was less than Thread.MIN_PRIORITY: it reset the input value to Thread.MIN_PRIORITY. The specification states that a value less than Thread.MIN_PRIORITY will be ignored.
  - java.lang.Character.isLowerCase/isUpperCase Methods Are Updated to Comply with the Specified Unicode Definition
  - The TypeVisitor interface has been updated
    - To model the language changes in this release, several updates were made to javax.lang.model.* including adding a method to the javax.lang.model.type.TypeVisitor interface
  - Do not define methods as final on java.lang.Throwable
    - Affects classes that extend Throwable. Methods addSuppressed and getSuppressed have been added.

http://www.oracle.com/technetwork/java/javase/compatibility-417013.html#behavioral

Moving to Java SE 7 runtime

- Networking
  -  Server Connection Shuts Down when Attempting to Read Data When http Response Code is -1
    - HTTP protocol handler will close the connection to a server that sends a response without a valid HTTP status line. When this occurs, any attempt to read data on that connection results in an IOException
- Text
  - The java.text.BreakIterator.isBoundary(int) Method Now Behaves as Specified
    - The java.text.BreakIterator.isBoundary(int) method now returns false, as specified, when the given offset is out of bounds, rather than throwing an IllegalArgumentException.

http://www.oracle.com/technetwork/java/javase/compatibility-417013.html#behavioral

Moving to Java SE 7 runtime

- Utilities
  - Updated sort behavior for Arrays and Collections may throw an IllegalArgumentException
    - The new sort implementation may throw an IllegalArgumentException if it detects a Comparable that violates the Comparable contract
  - Inserting an Invalid Element via constructor or puts methods into a TreeMap or TreeSet Throws an NPE
    - Previously it was possible to insert invalid null elements and elements not implementing the Comparable interface into an empty TreeMap or TreeSet. Additional elements would cause the expected NullPointerException or ClassCastException. Most other operations upon the collection would also fail.
  - Formatter.format() Now Throws FormatFlagsConversionMismatchException
    - The Formatter.format(String,Object...) method now throws a FormatFlagsConversionMismatchException exception when the "#" flag is specified for conversion "s" and the following argument is not a Formattable instance (including the special case "null").

http://www.oracle.com/technetwork/java/javase/compatibility-417013.html#behavioral

JPA differences

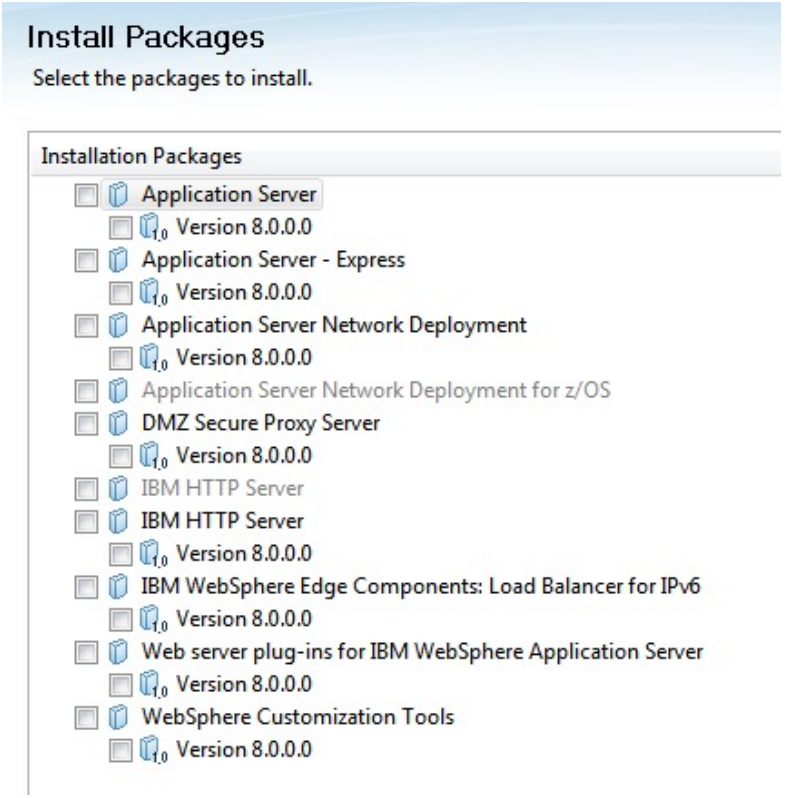
- Change in JPA cascade strategy
  - Behavior change for entity relationships that use cascade types PERSIST, MERGE and ALL.
    - The previous release would check the database for the existence of the related Entity before persisting the relationship to that Entity. This resulted in an extra Select being sent to the database. Now code was added so that when cascading a persist to a related Entity without persistence state, the persist (insert) will happen without first checking the database. This may result in an EntityExistsException if the related Entity already exists in the database.
    - To revert this behavior to the previous release, set the value of the openjpa.Compatibility property CheckDatabaseForCascadePersistToDetachedEntity to true
- Change in JPA MetaModel code generation concerning ListAttribute
  - In previous releases the MetaModel implementation generated a ListAttribute for every array. This behavior is correct if the array is annotated as a PersistentCollection, but not correct for un-annotated arrays (e.g. byte[], char[]). Now this behavior was corrected so that arrays which are not stored as PersistentCollections will use a SingularAttribute instead of a ListAttribute.
    - Behavior can be reverted by setting the Compatibility property <UseListAttributeForArrays to true in persistence.xml property name="openjpa.Compatibility" value="UseListAttributeForArrays=true"

v8.0 Changes

Administrative considerations

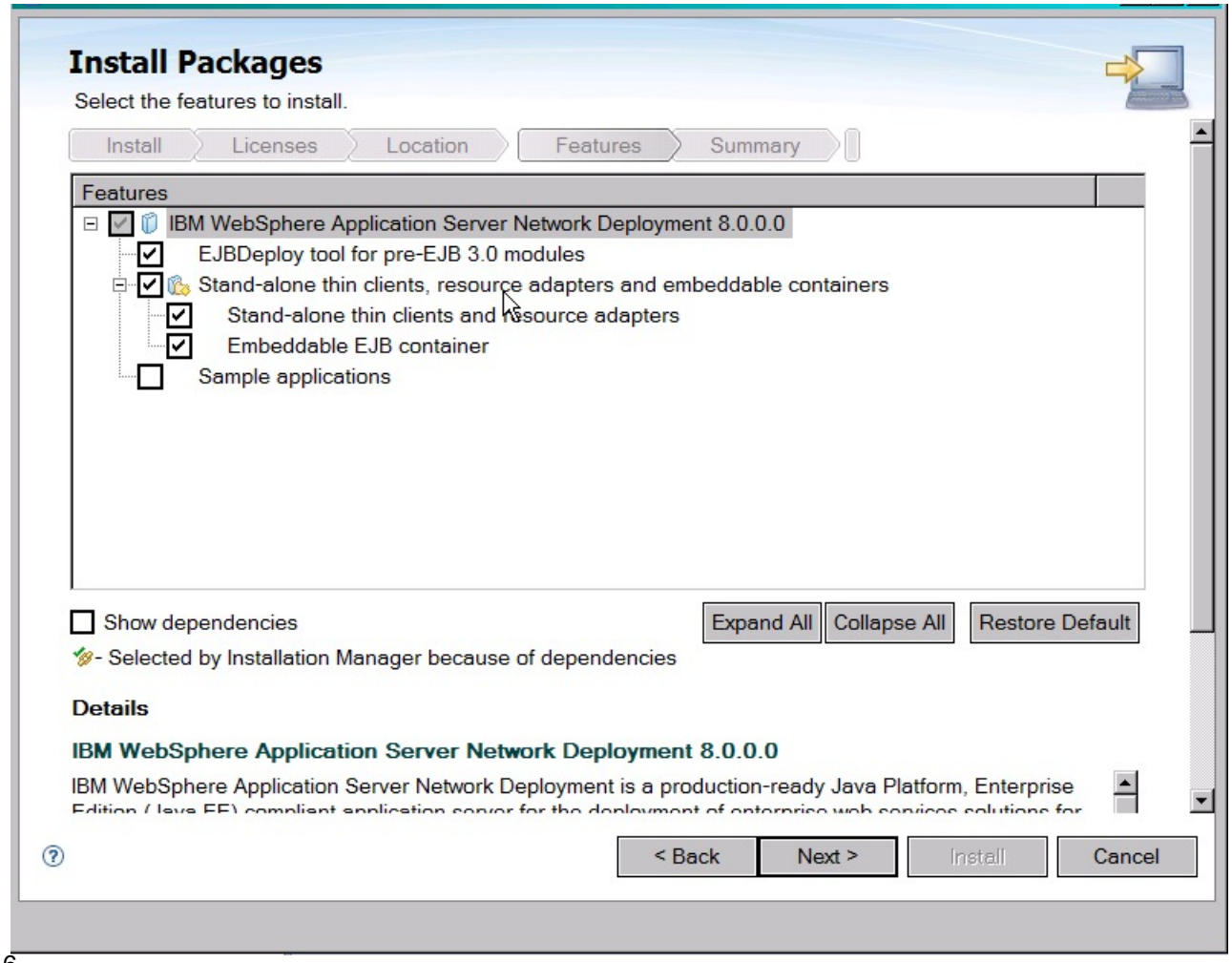
Install, Update, Uninstall via IBM Installation Manager

- Using IBM Install Manager as the install technology for the WebSphere Application Server and associated products
  - Faster installation
  - More customer control
  - Including z/OS
  - Use remote or local repositories
- A side effect is that Install response files from previous versions will need to be redone
- Another is uninstall cannot be done via script



More optionally installable features

- More optionally installable features
  - Allows to drive down the footprint of the installed product



Centralized Install Manager

- CIM is available from the Job Manager and DManager
  - Job Manager based solution spans the boundaries of the cell
  - Install targets are specified in agentless fashion
  - Install and config job scheduling is supported
- CIM is able to remotely install WebSphere Application Server, IBM HTTP Server, Application Clients, DMZ Security Proxy Server, and Web Server Plug-ins
- Better scalability due to more distributed architecture
- “CIM v7” function is still available in Deployment Manager along side with new function
- z/OS scenarios are supported

Install Factory replacement

- Install Factory is based on ISMP install technology and is not present in WAS v8
- Equivalent function is provided by combination of IBM Installation Manager and Centralized Install manager (depending on customer scenario)
  - IBM Installation Manager
    - Ability to install to specific level of service (GA + fixpacks + iFixes) in one step for multiple products
  - Centralized Install Manager
    - Ability to centrally manage enterprise wide installations, creation of profiles and execution of wsadmin scripts and other commands
- IBM provides a Packaging Utility tool to simplify the management of the content for IBM Installation Manager repositories

WebServer Plug-in installation and configuration

- Prior to v8.0, the Plug-ins installation wizard runs as part of product Install
  - Installs the plug-in module,
  - Configures the Web server and a Web server configuration definition in the application server, if possible.
- Change in v8.0
  - User must install both the plugin and the WebSphere Customization Toolbox
  - Then run Web Server Plug-ins Configuration Tool to configure the Web server and a Web server configuration definition in the application server, if possible

Java Garbage collection and dump format

- Java default GC policy changed from “optthroughput” to “generational”
  - May require higher memory allocation
  - Recommended starting point:
    - Set the tenured heap to the previous maximum heap size
      - ie. -Xmos = -Xms and -Xmox = -Xmx
    - Allocate the nursery with additional heap space
- Java verboseGC and Heapdump output formats have changed
  - Will require upgrades to associated ISA based tools

Security default changes

	WAS8	Prior release
VMM Active Directory for User look up for performance	(ObjectCategory=User)	(ObjectClass=Person)
VMM Active Directory for Group membership lookup for performance	"memberOf" attribute.	"member" attribute
VMM LDAP attribute search cache dist policy	None	Push
EJB/CSIv2 transport	SSL-required	Supported
Session security integrity	Enabled	Disabled
"Use available authentication data when an unprotected URI"	Checked	Uncheck
HttpOnly	Enabled	Disabled
Generated Certificate Key Length	2048	1024
DataPower Certificated	Now need to import	

Other default setting changes

- Oracle10gDataStoreHelper not supported on JRE 1.6, which is what is used by v8.0
  - Use Oracle11gDataStoreHelper instead
- ORB's socket connect time-out changed from zero to 10 seconds.
  - A value of zero means use the time-out set by the native operating system TCP/IP layer, usually set to 75 seconds in most operating systems, which has caused problems
- Asynchronous beans work manager “work request queue size" default calculation change
  - For v8.0 is the larger of maxThreads or 20, previously average of minThreads and maxThreads.
- Support for IBM Java Developer Kit for IBM i, which is also referred to as Classic JVM has been removed
  - Use IBM Technology for Java on IBM i, which includes IBM Java Standard Edition (SE) 32-bit and IBM Java SE 64-bit.

Other default setting changes…

New August 2013

- ClusterConfigCommands command group for the AdminTask object Parameter: -resourcesScope
  - Now is “cluster”, prior it was “both”
- Web container custom properties com.ibm.ws.webcontainer.throwpostconstructexception
  - Now is “true”, prior it was “false”
- Thread pool, Thread inactivity timeout
  - “60000” milliseconds, prior it was “35000”
- Webserver plugin changes
  - ServerIOTimeout setting changed to 900s
    - Was “60” in v7.0 and “0” in previous versions
  - Accept content for all requests setting
    - Now is “true”, prior it was “false”

Port usage

- v8.0 uses more ports than some previous versions
  - Can be an impact to those that tightly control port access
  - Can also cause more port conflicts

Server Type	v4.0	v5.x	v6.0	v6.1	v7.0	v8.0
Application		16	17	18	18	18
DMgr	n/a	11	8	17	12	12
Node Agent	n/a	9	11	10	11	11

Development considerations

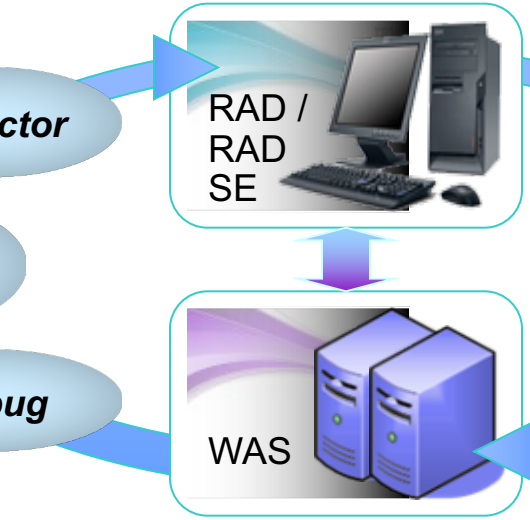
Integrated Tooling Support Through Rational Application Developer (RAD) & Rational Application Developer Standard Edition (RAD SE)

Web 2.0 & Mobile

Extend SOA and Java EE assets to the glass & mobile devices via dynamic, rich JSF, DOJO & mobile web applications

OSGi

Build dynamic, modular, and easily manageable applications



SOA

Assemble Web services and SCA components into heterogeneous business applications

Java EE 6

Develop and test Java EE 6 applications with annotation based programming

WAS Integration

Hot deploy incremental changes to WAS

Modern Batch

Integrated programming model support for batch applications

RAD & RAD Standard Edition

RAD

Team Productivity
- Integration with RTC
- Collaborative debug
- Collaborative code analysis

Problem Determination
- Code visualization - class, sequence and topic diagrams
- Static analysis (code review)
- Code coverage: optimize unit testing

Enterprise Connectivity
- J2C (EIS) tools
- CICS, and IMS Adapters
- WebSphere Adapters for SAP, Siebel, JDE, Oracle, PeopleSoft

RAD SE

Programming Model Support
- Create, edit, validate applications:
  - Specs / Standards:
    - Java EE (Web, EJB, Web Services, JAX-RS…)
    - SCA, OSGi, SIP, XML
  - Web (JSF, Dojo, JavaScript, Web 2.0)
- Debug applications on WAS
- Database tools
- Integration with ClearCase SCM Adapter, ReqPro, RUP

WebSphere Integration
- WAS test servers: v6.1, v7.0, v8.0, remote support for WAS 6.0
  - Publish, start/stop the server
- WAS Feature pack support
- Create and debug Jython and wsadmin scripts
- Portal Tools & Portal Server support
- Profile applications on WAS
- Find and deploy to WebSphere or Portal instances in the IBM SBDT cloud

IBM Assembly and Deploy Tools for WebSphere Administration (IADT)

Rapidly assemble & deploy applications to WebSphere Application Server environments

Key Capabilities:

- Import and validate applications
- Edit deployment descriptors and binding files
- Edit EAR-level configuration (Enhanced EAR)
- Create and debug Jython and wsadmin scripts
- Deploy EJB and web services
- Deploy applications to local or remote WAS v8 servers
- Debug applications on WAS v8

- IADT tools replace the previously available IBM Rational Application Developer Assembly and Deploy function
- Restricted to assembly and deployment usage only

Revision Date 06/2016

Moving to JEE 6

Bean Validation impacts versus Struts

New July 2013

- The Bean Validation component assumes file web-inf\validation.xml is it's control file
  - If this file was previously used in the application a delay will now occur in Bean Validation initialization code
    - Used by validator plug-ins of Struts
  - A warning is placed in the application server log
    - CWNBV0005W
  - The application will delay start between a number of millisecond to a number of SECONDS

EJB Changes

Interface	Detail	Old Behavior	New Behavior
EJB	EJBs in web modules	Ignored	Now processed, this may result in latent errors reported
@ApplicationException	in EJB 3.1 inherit is a new keyword and defaults to true	inherit=false	inherit=true, can impact subclassing exceptions behavior

JAX-WS Changes

Interface	Detail	Old Behavior	New Behavior
JAX-WS client	Some calls which result in local exceptions caused by an something invalid host or port	WebServiceException is thrown on the invoked method with an empty message	The Handlers handleFault message will be called with a SOAPFaultException
JAX-WS client	Handling of Policy:Addressing in WSDL	ignored	processed
SOAPMessage	getSOAPHeader and getSOAPBody behavior if there are no headers present	No error	Will now throw a SOAPException

Note: JAX_RPC has been deprecated in JEE6, stabilized by WAS

JCA Enhancement

Interface	Detail	Old Behavior	New Behavior
SQL exception error code	Paused datasources had no unique return code	Always returned 0	2147117569

JPA Changes

Interface	Detail	Old Behavior	New Behavior
EntityManager	refresh(…) method is passed a null	No Error	IllegalArgument exception is thrown
detach(Object entity)	New method on numerous APIs, JPA 1.x already has similar method, <T> T detach(<T> pc)	<T> T detach(<T> pc)	change from <T> T detach(<T> pc) to the new <T> T detachCopy(<T> pc)

- zOS DB2 different in JPA when using time related query
  - Your database is running DB2 for z/OS.
  - You are using a named query and access the database with native SQL.
  - The native query uses the time related field as an SQL parameter, but the query is not compatible with the column definition for the database table.
  - Exception: org.apache.openjpa.lib.jdbc.ReportingSQLException: THE DATE, TIME, OR TIMESTAMP VALUE 1 IS INVALID. SQLCODE=-18x, SQLSTATE=22007
  - See the Supported data conversions [link] topic in the DB2 9.7 Information Center for more information on compatibility.
- In some other cases v8.0 behavior may differ from earlier versions
  - Can revert to older (e.g. WAS 6.0) JPA provider as an alternate JPA provider in WAS 8.0
  - https://www.ibm.com/docs/en/was-nd/8.5.5?topic=provider-using-third-party-persistence-providers

JSF Changes

- The default JSF implementation was changed from SUN RI to MyFaces.
  - MyFaces will be at JSF version 2.0. SUN RI will be left at JSF version 1.2 and will be used only for backward compatibility.
  - Customers who want to take advantage of JSF 2.0 features will already have to rewrite their JSPs to use Facelets.
- More exceptions are passed on in JSF 2.0 that would not have surfaced in JSF 1.2
- JSF library override support:
  - https://www.ibm.com/docs/en/was-nd/8.5.5?topic=files-configuring-javaserver-faces-implementation

JSP Changes

Interface	Detail	Old Behavior	New Behavior
JSP-Property-Groups	is-xml and page-encoding configuration options should only apply to those JSPs which match the url pattern	Matching JSPs AND their included JSPs	Matching JSPs only

JSTL Changes

Added Dec 2013

Interface	Detail	Old Behavior	New Behavior
Java and EL reserved keywords	Variable names using these keywords are no longer allowed (See Note below)	Supported	Parse exceptions

- The code responsible for checking for the use of reserved keywords as EL variable identifiers was enhanced, making the checking more strict. The variable checking code not only checks for reserved EL keywords, but also Java reserved keywords
- The preferred method of resolution is to modify the JSTL/EL code to eliminate the usage of the reserved keywords as variable names. However, if this is not possible or practical, a custom JVM property *org.apache.el.parser.SKIP_IDENTIFIER_CHECK* can be set to *true* on the WebSphere Application Server JVM that will relax the variable identifier checks. The relaxed checking behavior is equivalent to the behavior observed in WebSphere Application Server versions prior to version 8.0.

OSGi Changes

Interface	Detail	Old Behavior	New Behavior
Composite bundles (.cba)	Scope of bundles packaged within composites	Public visibility outside of the Composite	Private visibility within the Composite

SIP Changes

Interface	Detail	Old Behavior	New Behavior
SIPFactory	createRequest and createAddress(String sipAddress) with URI as input or return parameters should be enclosed within brackets (“<>”)	Brackets ignored	Brackets handled correctly

Servlet Changes

Interface	Detail	Old Behavior	New Behavior
StoredResponse	return type of getHeaderNames **(Note, not a required change)**	Enumeration	Collection<String>
HttpSession invalidate	Use the HttpServletRequest logout method which was introduced in Java EE 6 as part of the Servlet 3.0 specification rather than the HttpSession invalidate method.	Programmatic logout not part of specification.	Programmatic logout is provided as part of specification.

WebSphere removals

- Apache SOAP implementation
  - Use standard WebServices support instead
  - See references for pointer to migration information
- com.ibm.websphere.ant.tasks.StopServer. setHost (java.lang.String host)
- com.ibm.websphere.rsadapter.WSConnectJDBCDataStoreHelper
  - See OracleDataStoreHelper rule. It combines the helper static and the helper class in one rule.
- com.ibm.websphere.servlet.error.ServletErrorReport.getCause ( )

Using Spring with WAS

Added Dec 2013

- In general the Spring Framework can be made to work with WAS fairly easily, see below reference for more details
- There is a known conflict starting in WAS v8.0 when using Spring due to a conflict with the JEE6 @Asynchronous annotation
  - Customer code using the annotation-driven search method in the Spring configuration for @Async annotation. @Async is the like functionality to @Asynchronous and the Spring code searches for both annotations against the classpath. The solution is to remove annotation-driven @Async searches by Spring.
  - Customer code using Spring prototype beans. A prototype bean is like a template and every time you want to use the template you have to create a hard copy that you actually use. When you create that hard copy, Spring pours over the class to resolve all of the container dependencies and resolve any annotations it's responsible for. Spring uses a bean post-processor for @Async and for every prototype that was hard-copied, Spring would run the post-processor even if the class did not have the @Async (or @Asynchronous) annotation in it. The @Async bean post-processor will increase method calls length. The solution is to stop creating Spring prototype beans.

v7.0 Changes

Administrative considerations

WSAdmin jython support

New June 2009

- Change to the string returned for “__name__” system variable
  - This system variable is set based on whether a file has been passed directly on the jython command line
    - Allows you to create a “main” program
  - Change was made to be compatible with standard Jython behavior
    - Prior to v7.0 the returned string was “main”
    - In v7.0 it is now “__main__”
  - Can check both for mitigation:
    - if __name__ == ‘__main__’ or ‘main’:

WSAdmin differences

New April 2010

- Checking for existence of resources
  - In some cases when checking for existence of a resource an exception would be thrown if the object did not exist
  - Now a return code will be returned and not an exception
  - E.g. if {[catch {set jp_name [$AdminConfig showAttribute $jp_tpl_id name]}]}
- Some Datasource definitions now require relationalResourceAdapter field
  - Either correct the connection factory defintion, or delete the data source with a wsadmin command and recreate it with the relationalResourceAdapter properly set
  - See https://www.ibm.com/support/pages/node/358243

launchClient differences

New August 2013

- Running multiple instances of launchClient in the same address space is not supported
  - Lock required for OSGi implementation
  - Resolve by adding the following to the invocation of launchClient
    - -Dosgi.configuration.area.readOnly

JAX-WS difference

New August 2013

- Using JAX-WS dynamic ports via the service method Service.addPort method
  - Default changed to visible to the instance of the service that did the addPort
  - Can cause OutOfMemory when migrating from older versions
  - Can revert behavior
    - jaxws.share.dynamic.ports.enable=true
  - Further information
    - https://www.ibm.com/support/pages/apar/PM63537
    - https://www.ibm.com/docs/en/was-nd/8.5.5?topic=clients-developing-dynamic-client-using-jax-ws-apis

Security default changes

Updated January 2009

- Old style LTPA version1 is disabled by default
  - In Prior versions both LTPA version 1 and 2 were sent for full interoperability
  - Can turn back on for interoperability needs is required
    - Set “com.ibm.ws.security.ssoInteropModeEnabled” = true instead of false
- WebSeal TAI interceptor is deprecated and discouraged
  - Available if the Migration tools are used to build the new cell
    - Migration ensures the old environment is carried forward
  - Not available by default as an option in the console otherwise
    - Can be added through the console only if necessary
      - Global security > Trust association > Interceptors
      - Add the interceptor using the “Add” capability and “com.ibm.ws.security.web.WebSealTrustAssociationInterceptor”
  - You should move to using com.ibm.ws.security.web.TAMTrustAssociationInterceptorPlus

Security logout servlet

New February 2009

- The ibm_security_logout servlet allows the specification of a web page to be displayed after the logout processing completes.
  - This web page was unrestricted so the user can be redirected anywhere. This was a security exposure and the behavior has changed
- The behavior now defaults to within the same host that the request is currently executing on
  - Properties are provided to revert behavior
- See APAR 71126
  - Also included in fixpacks 6.0.2.33 and 6.1.0.23.

Security Cipher changes

New June 2009

- Cipher strength changes in v7.0
  - Could affect interoperability between v6.1 and v7.0 systems
    - Appears as a security handshake error
  - Some have been removed and some have been changed
    - See the Speaker notes for detailed changes
  - Can mitigate by using one of the supported ciphers
    - Configured via the ssl.client.props file
  - See for more information:
    - https://www.ibm.com/docs/en/was/8.5.5?topic=configuration-sslclientprops-client-file

WebSphere MQ maintenance concern

Updated Nov 2009

- For zOS make sure the key APARs are installed
  - Information APAR II14484
    - https://www.ibm.com/support/pages/apar/II14484
- Be careful to use the correct MQ queue manager level in the correct mode (bindings or client)
  - See for more information:
    - https://www.ibm.com/support/pages/node/358077

WMQ Version	BINDINGS mode	CLIENT Mode
IBM WebSphere MQ v6.0.0.0 – v6.0.2.4	Not supported	Supported
IBM WebSphere MQ V6.0.2.5 or later	Supported WebSphere MQ v6.0.2.5 or later must be installed on the same machine as the application server	Supported
IBM WebSphere MQ V7.0.0.1 or later	Supported WebSphere MQ v7.0.0.1 or later must be installed on the same machine as the application server	Supported

WebSphere MQ usage and v7.0.0.0

New June 2009

- Adjusting the WebSphere MQ resource adapter configuration for profiles between maintenance level 7.0.0.0 and later levels
  - Profiles created at WebSphere Application Server maintenance level 7.0.0.0, the WebSphere MQ resource adapter binary files are located in each profile.
  - In profiles created at WebSphere Application Server maintenance level 7.0.0.0 Fix Pack 1 or later, these binary files are located in the *app_server_root* directory

- For profiles created at v7.0.0.0 level you must manually adjust the resource adapter configuration so that the profiles use the set of WebSphere MQ resource adapter binary files from the *app_server_root* directory

  - See for more information
    - https://www.ibm.com/support/pages/node/727251

MQ Topics and Queues default to read ahead

New August 2012

- The MQ topics and queues now default to read ahead under the following conditions:
  - When using the configuration migration tools to migrate from older WAS versions
  - When creating via wsadmin AdminConfig object
- The expected default of "As for queue definition" or "As for topic definition" is applied when:
  - The administrative console is used to create a WebSphere MQ JMS Queue or Topic object.
  - When using the createWMQQueue and createWMQTopic commands, with AdminTask

JSP class reloading

New February 2009

- "JSP enable class reloading" was added in v6.1 and was defaulted to “on”
  - This was different than v6.0 and earlier and sometimes caused performance problems due to the JSP reload rate
  - In v7.0 this value is reverted to “false”
  - Can be controlled via a custom property
  - See APAR PK71698

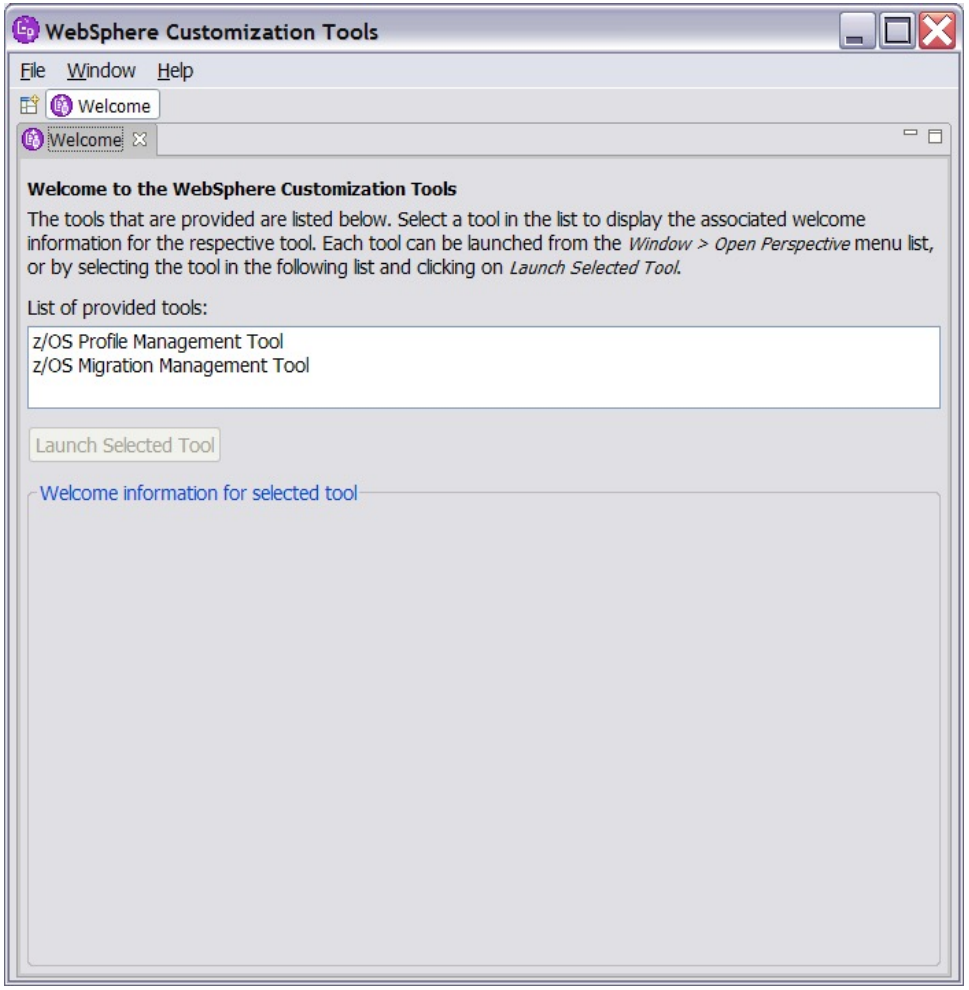
Plugin Server Timeout Default Value

New November 2009

- The HTTP Server default value of *serverIOTimeout* changed from “0” to “60” in v7.0
  - Affects behavior of GET requests
  - Can cause unexpected behavior because the old default reverted to the setting dictated by the OS
    - Varies between 5 and 30 minutes
  - Can simply change setting back to “0” if required

z/OS Migration Management Tool

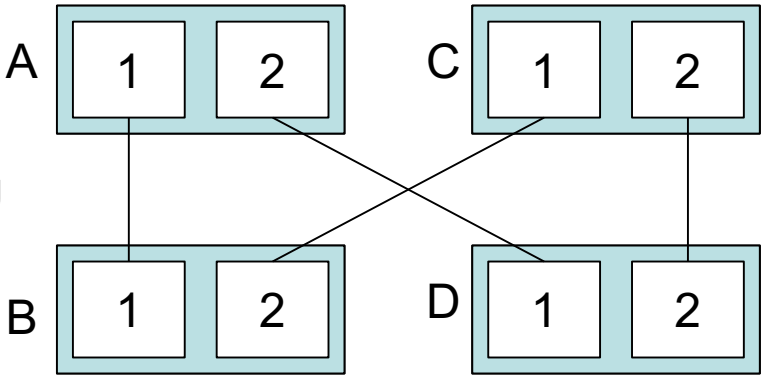
- Only alternative for generating zOS Migration JCL jobs
- Available via WebSphere Configuration Tool (WCT)
- Launch from Main menu



SIP Migration steps

1. Target node A for migration. Quiesce A1, A2 and their replication partners B1 and D1.
   - This leaves node A ready for migration because it has no replication partners active (so we won't hit our replication bytes versioning issue)
   - Stop those 4 servers once quiesce is complete
   - Migrate node A to 7.0 and get it up and running.
2. Target node B for migration. Quiesce B2 and its replication partner C1. Note B1 was stopped already.
   - Stop B2 and C1 once quiesce is complete
   - Migrate node B to 7.0 and get it up and running
3. Target node C for migration. Quiesce C2 and its replication partner D2. Note C1 was stopped already
   - Stop C2 and D2 once quiesce is complete
   - Migrate node C to 7.0 and get it up and running
4. Target node D for migration
   - Note that D1 and D2 were already stopped
   - Migrate node D to 7.0 and get it up and running

SIP will require additional steps for live migration



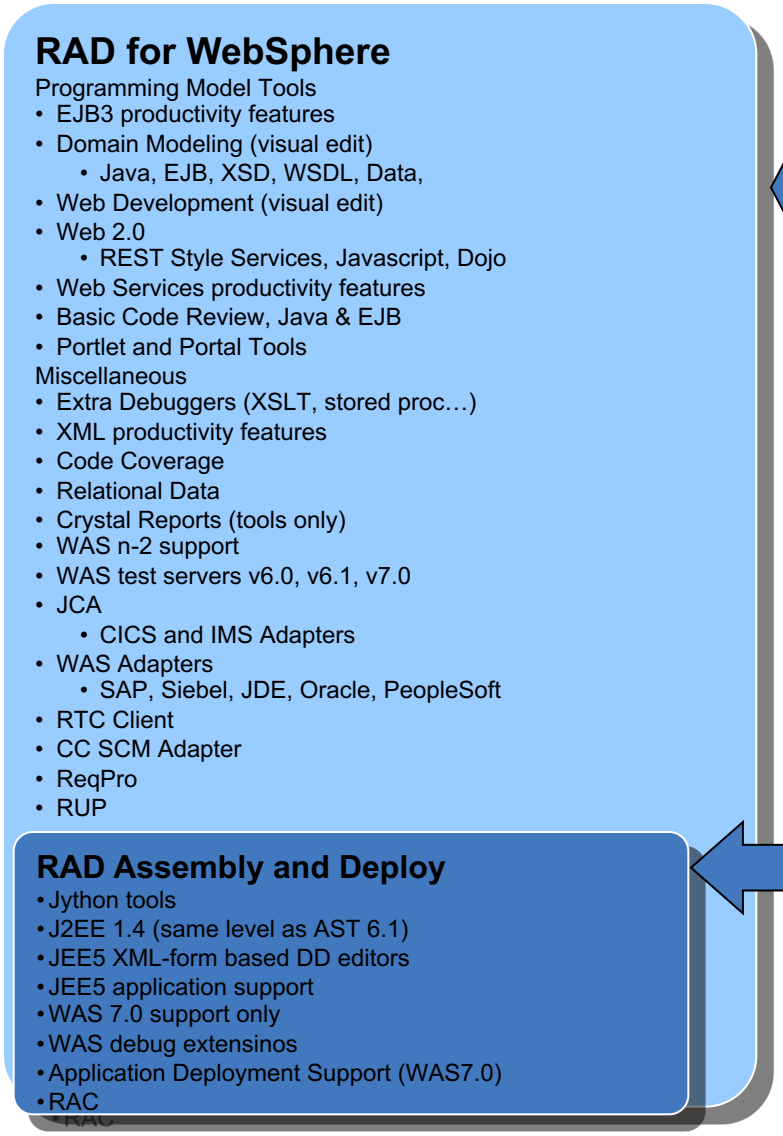
Port usage

- v7.0 uses more ports than some previous versions
  - Can be an impact to those that tightly control port access
  - Can also cause more port conflicts

Server Type	v4.0	v5.x	v6.0	v6.1	v7.0
Application	8	16	17	18	18
DMgr	n/a	11	17	17	12
Node Agent	n/a	9	11	10	11

Development considerations

WAS v7 Tools Packaging



RAD for WebSphere

- Comprehensive tools to improve developer productivity…targeting IBM Middleware
- Full-featured Integrated Dev Environment (IDE)
- Tutorials, demos, wizards and sample code

RAD Assembly and Deploy

- Perpetual license as part of WAS
- Support for creating, building, testing and deploying J2EE 1.4 applications
- Basic support for Java EE 5 applications
- Support for building, testing and deploying Java EE 5 applications

Development tool overview

- IBM Rational development tools
  - RAD v7.5 is the new development toolset
  - RAD v7.5 has similar requirements as RAD v7.0
    - Footprint and performance improvements in key scenarios
  - Based on Eclipse v3.4
  - Support for SOA, Web Services and Portal development
  - Support for Java6
  - Previous WebSphere Application Server supported runtimes
    - WebSphere Application Server v6.0
      - Includes support for Web 2.0 Feature Pack
    - WebSphere Application Server v6.1
      - Includes support for EJB 3.0, Web Services and Web 2.0 Feature Packs
  - Other supported runtimes
    - WebSphere Application Server v7.0
    - DB2 Universal Database™ (UDB) V9
    - Portal v6.0 and v6.1

Moving to JRE 6

JRE 6 impacts

- For an introduction, see the “Java SE 6“
  - http://java.sun.com/javase/6/
- Applications using the new language features and JRE 6 can be deployed only to v7.0 nodes.
  - When compiling applications can specify '-source' and '-target' modes for earlier JRE targets
  - E.g. '-source 1.4', and 'target 1.4'

Java Serialization

- Serialization is not compatible across JRE 1.4 and earlier releases – force UUIDs as a general practice

- Any new features in Java that result in new classes can cause ambiguous references
  - If these new classes match ones already defined in your program

JRE 6 source compatibility

Updated August 2012

JRE 6 is generally upwards source-compatible with JRE 5 except for some minimal factors:

- Some APIs in the sun.* packages have changed. These APIs are not intended for use by developers. Developers importing from sun.* packages do so at their own risk
- Java SE 6 In some cases, javac can now reject previously accepted, yet incorrect programs.
  - Properly Rejects Illegal Casts
  - EJB business methods which were not declared as public
  - FilterMapping that was mapped to a non-existing servlet
- Debug and Profiler interfaces have changed
  - Java Virtual Machine Debug Interface (JVMDI) have been removed, Java Virtual Machine Profiler Interface (JVMPI) has been disabled
- Non class files have been moved from rt.jar in Java SE 6
  - Java applications that specify -Xbootclasspath:<path to rt.jar> and request any resource files will fail since these resources now reside in a different jar file called resources.jar.
- Miscellaneous API changes
  - The Duration and XMLGregorianCalendar equals() methods now return false for null parameter
  - java.beans.EventHandler Enforces Valid Arguments
  - java.util.List - Correct behavior when access a larger index than possible. Now throws an IndexOutOfBoundsException instead of ArrayIndexOutOfBounds.
- Relatively minor impacts – see the sun site for a complete list. 145

Change in IBM Java 6 XSLT implementation

Added August 2012

- **Can result in high CPU utilization on existing apps that used XSLT**
- The default XSLT processor is changed to the XL TXE-J compiler. The compiling XSLT processor has different performance characteristics.
- It is best suited to situations where the same stylesheets are used many times and cached
- If the application was not designed with these differences in mind, it is best to continue using the old interpretive processor, which is still included in IBM Java 6
  - Revert to old XSLT compiler by setting javax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl system property

Moving to JEE 5

Some JEE minor specification changes

- This page focuses on those interfaces in v7.0 that have been upgraded to JEE5 with no alternative to using the older support in J2E 1.4. For example:
  - JSP has been upgraded to 2.1 from 2.0 – there is no way to run at JSP 2.0 level
  - EJB has been upgraded to 3.0, however EJB 2.1 applications are still run at 2.1
- Portlet 2.0
  - getProtocol for included servlets / JSPs no longer returns null, but ‘HTTP/1.1’

Use of .xmi versus .xml files

New August 2012

- For pre-JEE5 support you must continue to use .xmi bindings and extensions files
- For JEE 5 and later you must use .xml bindings and extensions files
- Approaches for migrating from .xmi to .xml files
  - Done automatically by RAD/RSA when you migrate an app from pre-JEE5 to or later.

Some JSP 2.1 changes

New March 2010

- New reserved character
  - The character sequence '#{' is now reserved by JSP. So If you are using '#{' in template text or as a literal in an attribute value for a 1.2-based taglib, the sequence will have to be escaped.
- Resource injection can cause some JSPs to no longer compile
  - Large JSPs may now be over the 64K limit due to new Resource injection support
  - Can turn this off via setting in ibm-ext-web.xmi in 7.0.0.11 and later
    - JSP attribute "disableResourceInjection" to true

Some JSP 2.1 changes…

New July 2010

- Redefining taglibs in a current scope no longer supported
  - Set the following JSP attribute: .
    - <jsp-attribute name="allowTaglibPrefixRedefinition" value="true"/> .
  - Or set a webcontainer custom property for the server (note this will affect all apps on the server):
    - com.ibm.wsspi.jsp.allowtaglibprefixredefinition=true
- A tag library directive that defines a prefix must occur before that prefix is used in a custom tag
  - Set a webcontainer custom property for the server (note this will affect all apps on the server):
    - com.ibm.wsspi.jsp.allowtaglibprefixusebeforedefinition=true
- Multiple occurrences of properties in the jsp:output element no longer supported
  - Set a webcontainer custom property for the server (note this will affect all apps on the server):
    - com.ibm.wsspi.jsp.allowjspoutputelementmismatch=true

Differences from v6.1 Feature Packs

Updated February 2009

- EJB 3.0 Feature Pack for v6.1
  - EJB 3.0 binding file errors - some applications may fail to start on WebSphere Application Server V7.0 because uniqueness checks are now performed on names used in the EJB 3.0 bindings file
  - Using runtime jars implicitly - When you deploy your Enterprise JavaBeans™ (EJB) applications, you might receive a runtime ClassNotFoundException exception because the class path entry Java™ archive (JAR) file is not exported or published and uses one or more runtime jars. This is a warning only.
  - V7.0 Does **not support** the use of bean managed persistence (BMP) and container managed persistence (CMP) entity beans in EJB 3.0-level modules. BMP entity beans **are supported** in the Feature Pack for EJB 3.0

Differences from v6.1 Feature Packs…

- WebServices Feature Pack for v6.1
  - In v7.0, JAX-WS annotations are supported only in modules whose version is Java™ EE 5 or later.
  - In v7.0 JAXB is provided as part of JRE6. The factory implementation is different in v7.0 than v6.1
- Both WebServices and EJB Feature packs
  - To preserve compatibility with v6.1 Feature Packs you must enable one of the following properties to request scanning during application installation and server startup:
    - UseWSFEP61ScanPolicy property for Feature Pack for Web Services
    - UseEJB61FEPScanPolicy property for Feature Pack for EJB 3.0

Differences from v6.1 Feature Packs…

Updated March 2012

- Both WebServices and EJB Feature packs
  - Servlet 2.5 modules in JEE 1.4 applications were allowed in v6.1and .xmi files were used to define bindings
  - Starting with WAS v7.0 Servlet 2.5 modules are not allowed to be deploy to WAS v6.1 in specification level JEE 1.4 or earlier applications and .xmi files cannot be used for Servlet 2.5 modules, must use .xml instead
  - Resolve be either:
    - Setting the Servlet level to 2.4 or earlier if still need to deploy to WAS v6.1
    - If Servlet 2.5 or later is required convert .xmi files to .xml. RAD has some migration tools to assist
      - See page on converting .xmi files

Using private/Open Source interfaces

Updated July 2013

- Problems can occur for applications older than v7.0 embed their own Open Source implementations
  - WebServices (AXIS based) and JAX-B (in Java6) is provided in WAS v7.0
  - Other Open Source implementations are used with WAS as well
  - Class conflicts can exist if those implementations are provided within the applications
- There are three practical solutions
  1. Use WAS Isolated Shared Library support to continue to use private version
     - Remove the jars from the application, create and deploy a shared library with the isolated class loader option
     - Use Shared Library class loader support to reference the shared library
  2. Turn off WAS WebServices or other support (not as fool proof)
     - May still have problems because of other support like JAX-B or others
     - See references section for information on how to turn off WAS WebServices
  3. Traditional Class loader options (not as fool proof)
     - Use PARENT_LAST
     - May still have a conflict between the two implementations

See https://www.ibm.com/support/pages/node/494997

JSF 1.2 impacts

Updated January 2014

- Rendering differences
  - Some fixes to resolve some previous problems with content-interweaving between JSF and non-JSF tags
  - Affects ViewHandler extenders, some custom scripting and some 3rd party packages that now need to support JSF 1.2
    - E.g. Tiles
  - And maybe others like
    - Tomahawk, Trinidad, IceFaces, Facelets, etc
- JSTL Tag Evaluation
  - In WAS6 and earlier, JSF evaluates JSTL tags after ScriptCollector preRender method was executed. Whereas in WAS7, JSF evaluates JSTL tags before ScriptCollector preRender method is invoked and fails as backing bean property used in the JSTL tag is not populated by the ScriptCollector preRender method.
    - This change will require a redesign of your JSP pages to avoid the dependency on execution order.

JSF 1.2 impacts…

New March 2010

- JSF Portlets
  - Higher memory consumption is reported, may result in larger sessions
- JSF 1.0 application impacts
  - Will either need to change to JSF 1.2 or restructure the application
  - If restructuring the application will need to pull the JSF 1.0 modules out of the application and into a shared library
  - See Speaker notes for more details

Scanning for annotations…

New June 2009

- JEE5 introduces support for annotations
  - Requires a new step to scan Java annotations during application installation which can take significant amount of time
  - For a Web module, that includes both the classes packaged directly within the WAR file (under the WEB-INF/classes directory), and classes that are packaged in JAR files within the WAR file
- Ways to optimize applications directly
  1. Only mark modules as JEE5 level when they contain JEE 5 content
  2. If the module is known to have no annotations, use the "metadata-complete" flag
     - See speaker notes for more details
  3. Restructure the application to place utility JAR files, which are known to contain no annotations information, into shared libraries and/or root of the EAR
     - These are not scanned for annotations
  - See for more information:
    - https://www.ibm.com/docs/en/was/8.5.5?topic=installations-metadata-module-settings
- Ways to use configurable filtering (7.0.0.5 and later)
  - Can identify which modules and/or Java packages to ignore for annotations processing
  - Ignore-Scanning-Archives and Ignore-Scanning-Packages. A default set of values are provided in the amm.filter.properties file that resides in <WAS_HOME>/properties.
  - See for more information:
    - https://www.ibm.com/support/pages/apar/PK87053

Scanning for annotations…

New November 2009

- More files are now processed (e.g. .zip files)
  - There have been some problems now that additional file types are processed as part of the scanning
  - One example is encrypted zip files that used to be ignored prior to v7.0
  - Some problems can be avoided starting in v7.0.0.7 by the following:
    1. Go to the AdminConsole of the Deployment Server, navigate to:
       - Application servers --> <deployment server name> --> Java and Process Management --> process Definition --> control --> Java Virtual Machine --> Custom Properties
       - Add com.ibm.websphere.application.migration.disabled, to list with its value set to true and save the configuration
    2. Go to <profile_home>/properties/wsadmin.properties
       - Edit and add com.ibm.websphere.application.migration.disabled=true
- See for more information:
  - https://www.ibm.com/support/pages/apar/PK92880

WebSphere API migration details

WebSphere removed support

Updated September 2009

- Derby Network Server Provider using the Universal JDBC driver
  - Use the Derby Network Server using Derby Client instead.
- Support for the DB2 legacy CLI-based Type 2 JDBC Driver and the DB2 legacy CLI-based Type 2 JDBC Driver (XA) has been removed
  - Instead, use the DB2 Universal JDBC Driver
- The following Java Database Connectivity (JDBC) drivers
  - WebSphere Connect JDBC driver, Microsoft® SQL Server 2000 Driver for JDBC , WebSphere SequeLink JDBC driver for Microsoft SQL Server
  - Alternatively use the DataDirect Connect JDBC driver or Microsoft SQL Server 2005 JDBC driver
  - Also see the WebSphereConnectJDBCDriverConversion command to convert data sources from the WebSphere Connect JDBC driver
- Integrated Cryptographic Services Facility (ICSF) authentication mechanism
  - Alternatively use the Lightweight Third-Party Authentication (LTPA) mechanism
- Support for the following Security custom properties

Old Property	New Property
com.ibm.security.SAF.unauthenticatedId	com.ibm.security.SAF.unauthenticated
com.ibm.security.SAF.useEJBROLEAuthz	com.ibm.security.SAF.authorization
com.ibm.security.SAF.useEJBROLEDelegation	com.ibm.security.SAF.delegation

WebSphere removed interfaces

- All classes in the com.ibm.websphere.servlet.filter package
  - ChainedRequest , ChainedResponse , ChainerServlet and ServletChain
  - Alternatively use javax.servlet.filter classes
- Web services gateway customization API
  - Alternatively replace your existing filters with a combination of JAX-RPC handlers and service integration bus mediations
- The following miscellaneous classes

Old Class	New class
com.ibm.websphere.servlet.session.UserTransactionWrapper	Store a UserTransaction directly into the HTTP session
com.ibm.websphere.rsadapter.DataDirectDataStoreHelper	com.ibm.websphere.rsadapter.ConnectJDBCDataStoreHelper
com.ibm.websphere.rsadapter.MSSQLDataStoreHelper	com.ibm.websphere.rsadapter.MicrosoftSQLDataStoreHelper

DistributedMap updated support

Updated February 2013

- DistributedMap entrySet() method was not available on V7.0 or higher
  - Caused ClassCast exception
  - Returned a set of keys instead of a set of key/value pairs
  - Reverted to old behavior key/value pairs in fixstream
  - Fixed in 7.0.0.27, 8.0.0.6 and 8.5.0.2
  - www-01.ibm.com/support/docview.wss?uid=swg1PM71965

v6.1 Changes

Administration Changes

Administration script required changes

- Administration scripting changes SIB Bus creation
  - Securing requires a new parameter (busSecurity) instead of the previous parameter (secure)
- Changes for removed features
  - Support for the Secure Authentication Service (SAS) IIOP security protocol.
  - Support for the Common Connector Framework (CCF).
  - Support for the IBM Cloudscape Version 5.1.x database.
- Simplified Certificate/Key Management
  - Movement away from dummy key files to new certificate model
  - Property file changes
    - New file: ssl.client.props
    - Reorganized files: soap.client.props, soap.server.props
  - SSL configuration changes
    - SSL repertoire changed, but old model still supported as well

Administration script required changes

- Foreign cell names bootstrapAddress change
  - Foreign cell name bindings enables your applications to access other applications in other cells
  - “bootstrapAddress” has changed to “bootstrapAddresses”
  - Existing bootstrapAddress attribute is still supported and is deprecated

Security script changes

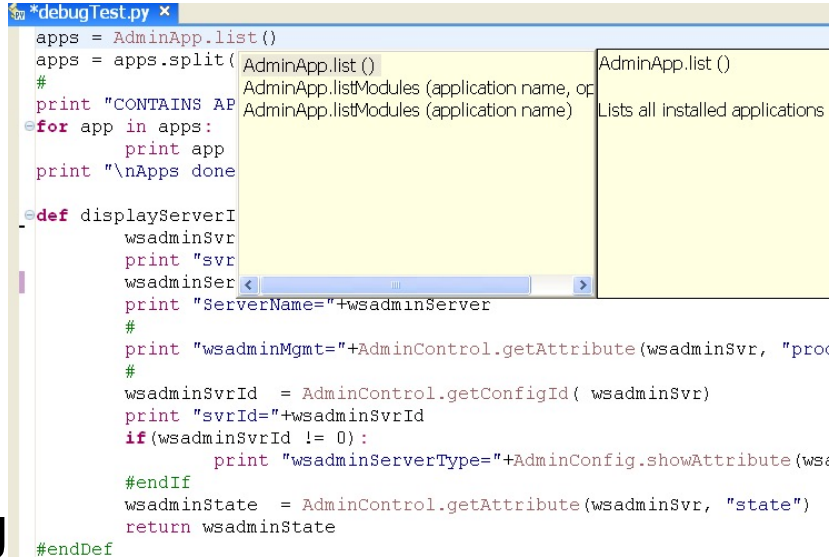
New March 2007

- Security model has been enhanced
  - Existing scripts will still work with existing model
    - But you are not able to take advantage of new model
  - At some point you will want to rework these scripts to use the newly provided AdminTasks
  - Additional documentation has been provided on the new model
    - Commands for the security enablement group of the AdminTask object
    - Automating SSL configurations using scripting
    - Creating self-signed certificates using scripting
    - See Others in Speaker notes

Administrative script tools

New December 2006

- WebSphere admin automation tools for the creation and maintenance of wsadmin Jython files
  - Includes an editor with an outline view, color syntax highlighting and context sensitive code completion
  - Integration with the new "command assist" function available in the admin console
    - Intercepts generated commands for easy insertion into Jython Script



- Jython debug support provides an integrated debugger for stepping through the execution of a wsadmin Jython script
- Jacl2Jython conversion assistant

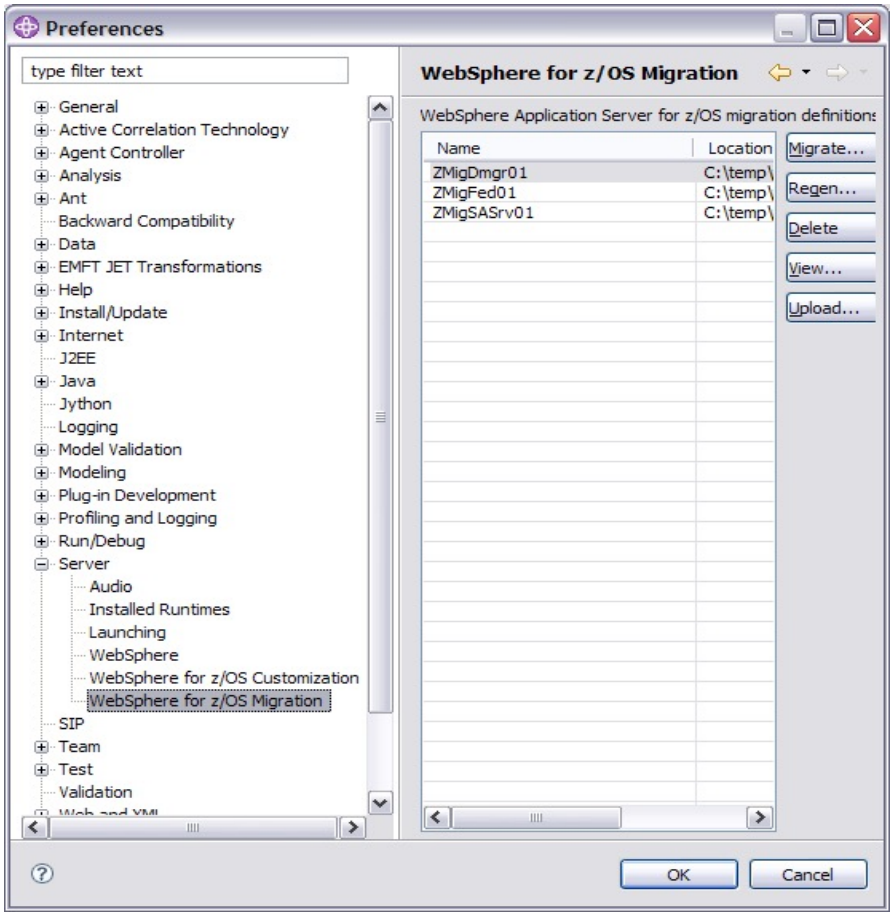
WebSphere configuration removals

- SAS and zSAS are no longer supported
  - Deprecated since v5.1
  - It was used for Interoperability with older versions
  - Use CSIv2 instead
- zOS specific configuration changes
  - DB2 for zOS Local JDBC Provider (RRS) is removed
    - Use DB2 Universal driver instead
    - Migration tooling for JDBC Provider conversion
      - See Resources under “zOS Specific”
  - System SSL supported for Daemon only
    - Use JSSE/JSSE2 support instead
- Log Analyzer is removed
  - Use the Log and Trace Analyzer tool for Eclipse in the Application Server Toolkit (AST)

z/OS Migration Management Tool

New February 2008

- New alternative for generating zOS Migration JCL jobs
- Available with Application Server Toolkit (AST) 6.1.1 or later
- Launch from Preferences window



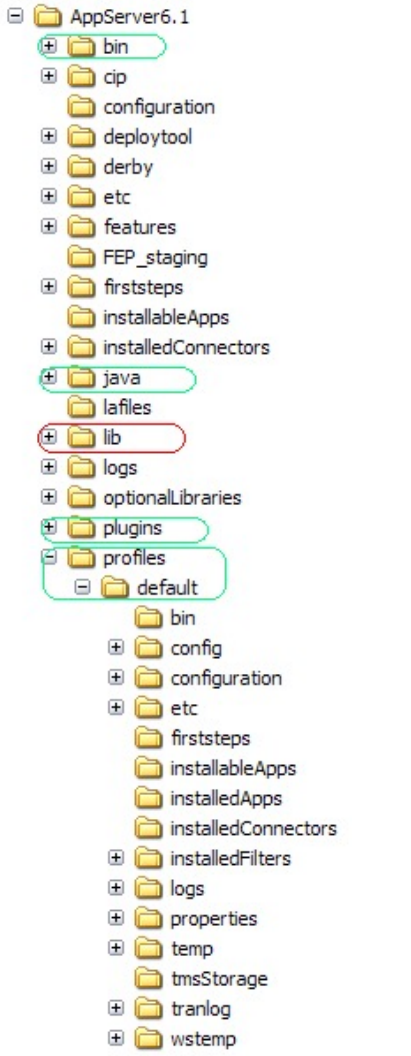
Profile management tools replaced

New March 2007

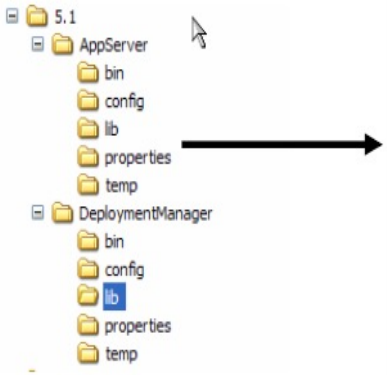
- zOS support is unchanged from v6.0 to v6.1
- Profile creation is supported by two types of commands
  - Scripts
    - v6.0 is wasprofile.sh(bat)
    - v6.1 is manageprofiles.sh(bat)
      - Note that wasprofile.sh(bat) is deprecated but still supported
    - Same command line parameters
  - Graphical User Interface (GUI)
    - v6.0 is bin\ProfileCreator\pctXXX.exe
      - “xxx” varies based on the Operating System
    - v6.1 is bin\ProfileManagement\pmt.sh(bat)

Profiles

- Increased flexibility and other benefits
- Beware that some scripts may break
  - Use USER_INSTALL_ROOT\logs
  - Not WAS_HOME\logs
- config, bin, log… directories now exist under each profile



- /lib and jar locations have changed
  - Avoid building your own install images – use the supported ones
- Classes directory is removed
  - Use Shared libraries



Directory structure changes

New March 2007

- Direct references to WebSphere jar files
  - The structure of WebSphere jar files changes in v6.1
  - Some of the jars have moved from
    - /lib to /plugins
  - If you referenced these jars directly in scripts you will have to make changes
  - However, it is not always as easy as finding the classes in the new location
    - OSGI needs to be initialized
  - The safest approach is to call the appserver/bin/setupCmdLine shell in your scripts
  - This will setup the correct classpaths for you

Install response file changes

New March 2007

- For all but zOS, the response files have changed
  - The ISMP format (-W option=value and –P option=value) replaced with an -OPT option, takes “sub-options”

V6.0 option	V6.1 option
-W silentInstallLicenseAcceptance.value=”true”	-OPT silentInstallLicenseAcceptance.value=”true”
-W detectedexistingcopypanelInstallWizardBean.choice=”installnew”	**-OPT** installType="installNew"
-P samplesProductFeatureBean.active=”true” (only used if detectedexistingcopypanelInstallWizardBean.choice="addFeaturesAndFixes") **(used for incremental installs)**	**-OPT feature="samplesSelected"** for samples when –OPT installType=”installNew” or –OPT installType=”addFeature” **-OPT feature="noFeature"**for no samples, when -OPT installType="installNew"(installType cannot equal addFeature if you don’t select to install any features)
-P javadocsProductFeatureBean.active=”true”	No equivalent
-P wasProductBean.installLocation=	**-OPT** installLocation=Note this format will vary based on OS
-W ndsummarypanelInstallWizardBean.launchPCT=”false”	-OPT **createProfile="false"**
-W ndsummarypanelInstallWizardBean.launchPCT=”true”	**-OPT** createProfile="true"**And see the next entry in this table**
-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation	**See Speaker notes for this entry**

Port usage

- v6.1 uses more ports than previous versions
  - Can be an impact to those that tightly control port access
  - Can also cause more port conflicts

Server Type	v4.0	v5.x	v6.0	v6.1
Application	8	16	17	18
DMgr	n/a	11	17	17
Node Agent	n/a	9	11	10

Migration and Feature Packs

New February 2008

- Some restrictions exist with Feature Pack usage
  - Cannot migrate to any v6.1.x profile that has been augmented for any v6.1 feature pack
  - Can only augment a new v6.1.x standalone server or custom profile
- Deployment Manager profiles
  1. Migrate a v5.x or v6.0.x deployment manager to a v6.1.x deployment manager profile
  2. Migrate all the federated nodes to v6.1
  3. Augment the v6.1.x deployment manager profile with the Feature Pack that you want to use
- Standalone server or custom profiles
  - Can either Migrate or augment for Feature Pack usage but cannot do both
  - Alternatives if you want both:
    - Migrate as you normally would
    - Add a new cell specifically for Feature Packs
    - Add a new node in an existing cell for Feature Packs

Development Changes

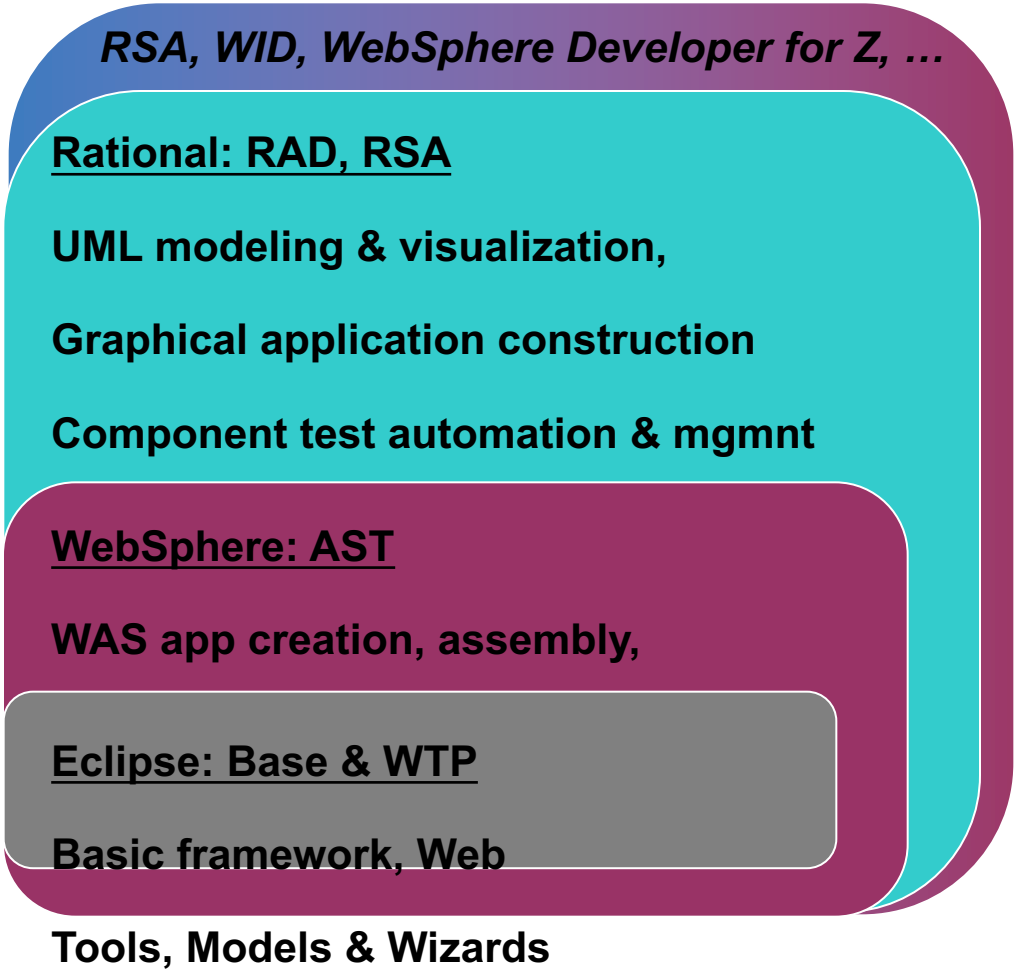
Development tool overview

New January 2007

- IBM Rational development tools
  - RAD v7.0 is the new development toolset
  - RAD v7.0 has similar requirements as RAD v6.0
    - Performance improvements in key scenarios
  - Based on Eclipse v3.2
  - Support for SOA, Web Services and Portal development
  - Support for Java5
  - Previous supported runtimes
    - WebSphere Application Server v5.1 and v6.0
  - New supported runtimes
    - WebSphere Application Server v6.1
    - DB2 Universal Database™ (UDB) V9
    - Portal V6

Positioning of the Tool Offerings

New December 2006



- RAD, RWD, RSA provide the first class IBM design/construction tools for the WebSphere Application Server
- Focus is on the graphical construction of applications and code generation for maximum developer productivity
- AST ships with WAS, and provides complete WebSphere support
- It is focused at a more basic (textual) level than RAD, but provides support to create & deploy WAS apps
- Eclipse WTP (Web Tools Platform) is a new open source project

Development tool overview

New October 2006

- AST is a proper subset of RAD
  - No Embedded WebSphere Test Environments
    - Must use an installed WebSphere Application Server v6.1 environment to test
  - No support for previous versions of WebSphere Application Server
  - No Page designer and site designer (web site tools)
  - No JSF tools
  - No Code review tools
  - No Component testing tools

Moving to JRE 5

JRE 5 impacts

- Applications using the new language features and JRE 5 can be deployed only to v6.1 nodes.
  - When compiling applications can specify '-source' and '-target' modes for earlier JDK targets
  - E.g. '-source 1.4', and 'target 1.4'
- Java Serialization
  - Serialization is not compatible across JRE 1.4 and earlier releases – force UUIDs as a general practice

- Any new features in Java that result in new classes can cause ambiguous references
  - If these new classes match ones already defined in your program

JRE 5 source compatibility

- JRE 5 is generally upwards source-compatible with JRE 1.4.2 except for:
  - Some APIs in the sun.* packages have changed. These APIs are not intended for use by developers. Developers importing from sun.* packages do so at their own risk
  - Variables named 'enum.‘ - The word 'enum' has become a language keyword
    - NOTE – Some IBM WebServices generated code may include “enum” as package names – should regen (change com.ibm.ws.webservices.engine.enum… to com.ibm.ws.webservices.engine.enumtype…)
  - Generification – Most source code that uses generified classes, constructors, methods, and fields will continue to compile in 5.0, though some will not.
  - Ambiguous references to classes with base names of 'Proxy,' 'Queue,' or 'Formatter.‘ – these are new classes in some JDK 5 packages
  - JAXP – a variety of changes, some Binary incompatibility
    - See http://java.sun.com/j2se/1.5.0/docs/guide/xml/jaxp/JAXP-Compatibility_150.html
  - JDBC 5.0 - comparing a java.sql.Timestamp to a java.util.Date by invoking compareTo on the Timestamp results in a ClassCastException. Binary incompatibility
  - BigDecimal's toString() method behaves differently than in earlier versions. J2SE 5.0 added toPlainString() to BigDecimal, which behaves exactly like the toString() method in earlier versions. Binary incompatibility
  - Direct use of private implementations of XML and XSL parsers is strongly discouraged
    - Can use existing classloader support to use an application class path, not the Java virtual machine bootstrap class path

JRE 5 verboseGC output

New July 2010

- The GC output format varies depending on the garbage collection policy that you use

Title	Description	URL
MustGather: Analysing Verbose GC Output for -Xgcpolicy:gencon	Format of verbose GC output from the gencon GC policy	https://www.ibm.com/support/pages/node/344441
MustGather: Analysing Verbose GC Output for -Xgcpolicy:optavgpause	Format of verbose GC output from optavgpause GC policy	https://www.ibm.com/support/pages/node/344407
MustGather: Analysing Verbose GC Output for -Xgcpolicy:optthruput	Format of verbose GC output from optthruput GC policy	https://www.ibm.com/support/pages/node/344385

JRE 5 and JSSE2

- Package **com.ibm.net.ssl (JSSE)**
  - Affects classes related to creating and configuring secure socket factories
    - E.g KeyManager, TrustManager, X509KeyManager, X509TrustManager
  - Deprecated since JRE 1.4 and replaced with javax.net.ssl package
  - Removed as from JRE 5

JDK 5 feature usage and JSPs

New November 2007

- If JSP's use any JDK 1.5 specific code, will get an error during compile
  - The default runtime compiler setting for JSPs is JDK 1.3
- There are two solutions
  1. (Ok) Application installation
     - ✓ The option to specify the JDK Source level when installing
  2. (Better) Inside RAD
     - ✓ The Web Extension tab
       - ✓ In JSP Attributes table add “jdkSourceLevel jdkSourceLevel” as name and 15 as value.
     - ✓ Adds an entry to Web Project/WEB-INF/ibm-web-ext.xmi
     - E.g. <jspAttributes xmi:id="JSPAttribute_1" name="jdkSourceLevel" value="15"/>

WebSphere API details

WebSphere removed APIs

- Common Connector Framework (CCF) is removed
  - Deprecated since v5.1
  - Use J2C instead
- Support for Cloudscape v10.1 instead of v5.1
  - Cloudscape is **not** supported for direct customer production use
  - Is Derby v10.1 plus NLS and QA
  - Affects JDBC driver configuration
  - Changes in some DB types; some conversions required
- Mozilla Rhino Javascript is removed
  - Required licensing agreement
  - Download from the Web instead
- Java Document Object Model (JDOM) is removed
  - Download from the Web instead

WebSphere removed APIs…

- Security
  - Removed *com.ibm.websphere.security.CustomRegistry*,
    - Use: *com.ibm.websphere.security.UserRegistry*

Change in EJB setRollbackOnly()

New February 2009

EJB setRollbackOnly() change description:

- EJB1:method1 and EJB2:method2 have Container Managed Transactions and are defined with a transaction attribute of 'TX_REQUIRED'.
- EJB1 and EJB2 are hosted in the same WebSphere Application Server
- HTTP request invokes EJB1:method1 outside of any transaction. (The Enterprise JavaBeans Container for EJB1 creates a new transaction) EJB1:method1 invokes EJB2:method2. (EJB2:method2 runs under the same transaction)
- EJB2:method2 invokes setRollbackOnly(). EJB2:method2 returns normally. The HTTP request receives a RemoteException org.omg.CORBA.TRANSACTION_ROLLEDBACK.
- In WebSphere Application Server V6.0 and earlier, the HTTP request receives the business result of EJB1:method1, not a RemoteException.
  - In each case the transaction is rolled back, but in V6.1 the HTTP request does not receive the business results returned by EJB1:method1.

JNI on Solaris 10 x86_64

New December 2006

- Changes between v6.0.x versus v6.1.x on 64 bit
  - In v6.0.x WebSphere Application Server is a 32bit application
    - The JVM is a 32 bit application
    - JNI calls performed by customer applications are also 32 bit
  - In v6.1.x WebSphere Application Server is a 64bit application
    - JNI calls performed by customer applications are also 64 bit

v6.0 Changes

Administration changes

Administration script required changes

- Parsing the string output of the ObjectName class:
  - configID used in v6 contains a vertical bar character ('|") instead of a colon character (':')
  - In general try and use a different technique such as queryNames
- regexp Jacl command
  - Version of Jacl is different in v6.
    - 1.3.1 versus 1.2.6
  - regexp command supports only **tcl 8.0** command syntax. Some of your existing scripts may fail, e.g.
    - “error while eval'ing Jacl expression: couldn't compile regular expression pattern: ?+* follows nothing”

Administration script evolutionary

- Transaction log directory change:
  - Old: ApplicationServer:TransactionService
  - New: ServerEntry:RecoveryLogs
  - Uses old location until new location is set
- HttpTransports (n/a to zOS v6.0)
  - HTTP Transports replaced by ChannelFramework
  - Affects all scripts accessing HTTP Transports
- ProcessDefinition (n/a to zOS)
  - “processDef” changed to “processDefs”
- Migration runtime tools can be used to maintain script compatibility with:
  - HttpTransports and processDef

Administration script zOS only changes

- Several bugs in zOS scripting were fixed when moving to v6.0 – causes some breakage
- New exceptions are thrown for some conditions
  - E.g. stopping server and starting applications when conditions are already met
  - Fix by catching the exceptions:

```
if { [catch {“yourCommandHere”}] }
  { puts "it failed" }
else
  { puts "it was ok" }
```

- taskInfo command keywords changed
  - Old: "module", "EJB", "uri", …
  - New: "Module", "EJB", "URI", …

Administration console port

- The administrative console port number changed from 9090 to 9060
  - Changed in order to minimize port conflicts on some operating systems
  - Runtime Migration tooling will add v5.x value to the configuration

Port usage

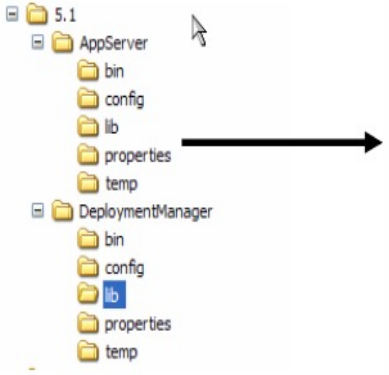
- V6.0 uses more ports than previous versions
  - Can be an impact to those that tightly control port access
  - Can also cause more port conflicts

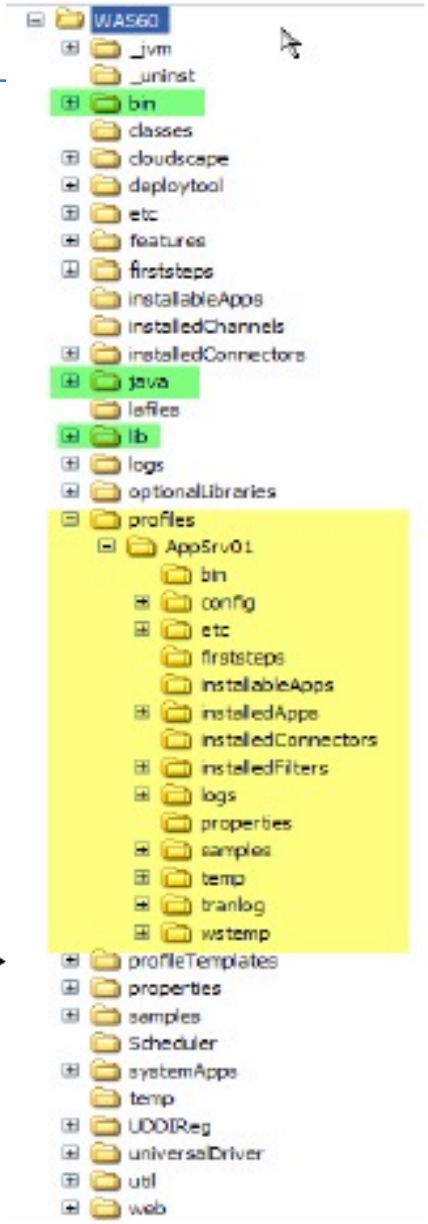
Server Type	v4.0	v5.x	v6.0
Application	8	16	19
DMgr	n/a	11	17
Node Agent	n/a	9	11

Profiles

- Increased flexibility and other benefits
- Beware that some scripts may break
  - Use USER_INSTALL_ROOT\logs
  - Not WAS_HOME\logs
- config, bin log… directories now exist under each profile

- /lib and jar locations may also change
  - Avoid building your own install images – use the supported ones



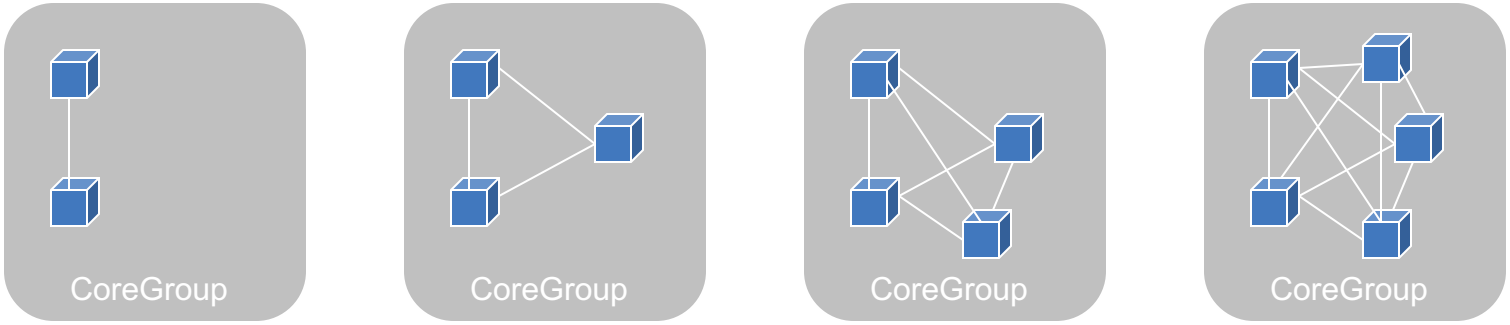


JMS Support

- Embedded messaging implementation changed
  - Existing JMS resources are still supported unchanged but not optimal
  - Should upgrade to v6 JMS Resources when feasible
  - JMS Server configuration changed
    - JMSServer object no longer exists
  - Data on v5.x queues are not migrated automatically
    - See SibMsgMigrationUtility.ear
  - DIRECT no longer supported on ConnectionFactories instead it should be changed to QUEUED

CoreGroup considerations

- Each Application Server, NodeAgent and DMgr are added to the default CoreGroup by Migration tools
- Performance concerns for larger cells
  - v6.0 - recommend maximum around 50 objects per CoreGroup
  - v6.1 and later - recommend maximum around 100 objects per CoreGroup
- Mitigate by managing CoreGroups manually
  - Each CoreGroup must contain at least one NodeAgent or DMgr process
    - Requirement removed in v7.0 and later
  - Put all members in the same cluster in the same CoreGroup



HA Manager disablement

Updated March 2006

- If necessary and meets your environment, HA Managers can be disabled by using a wsadmin script
- For v6.0 This also disables:
  - Memory-to-memory replication (any of the following)
    - Http Session memory-memory replication
    - Dynacache cache replication
    - EJB statefull session bean failover
  - Singleton failover when WebSphere JMS provider is configured in a cluster
  - Workload management routing for EJB IIOP and JMS components
  - On-demand configuration routing for Proxy server and Web services
- For v7.0 This also disables:
  - Workload management routing for Proxy Server (HTTP and WS-Addressing) and SIP requests
  - Resource adapters configured for high availability
- Be sure to disable all HA Managers within a CoreGroup
  - https://www.ibm.com/docs/en/was-zos/8.5.5?topic=applications-disabling-enabling-high-availability-manager

Development Changes

Development tool overview

- IBM Rational development tools
  - RAD v6.0 is the development tool for WAS v6.0
  - Very compatible with WSAD
  - Based on Eclipse 3.0
  - “Similar” hardware requirements to WSAD
    - Available RAM (768Min, 1G recommended)
  - Supports J2EE 1.2, 1.3 and 1.4 applications
  - Test support for WebSphere v5.0, v5.1 and v6.0
  - Migration of WSAD v5.1 workspace to RAD v6.0
  - Projects import directly from WSAD v5.1
    - Project import from WSAD v5.0 may work
      - Alternatively create a new project and import the source

J2EE 1.4 impacts

- Some changes may be required due to tightening of J2EE levels
  - More deployment descriptor errors are found during application installation
    - Typically the problems are hand-edited changes and used to work or were ignored in prior versions
  - Some JSP behavior and interface changes
  - Some Servlet behavior and interface changes
  - Some Web Services behavior changes

JSPs useBean tag

- JSP useBean tag
  - Tightened conformance to JSP 1.2
  - Behavior change starting in v5.1
  - Old: <jsp:useBean id="pg" **class=**"exp.ObjExp.exampleTbl" scope="request" />
  - New: <jsp:useBean id="pg" **type=**"exp.ObjExp.exampleTbl" scope="request" />

JSPs and unnamed packages

- In JDK 1.4, importing classes from the unnamed package is not valid
  - As of JSP 2.0, it is illegal to refer to any classes from the unnamed (a.k.a. default) package
    - Surfaces as a translation error
    - This also affects older applications that run on v6.0
  - For example, if myBeanClass is in the unnamed package, and you reference it in a jsp:useBean tag
    - **<%@page import="myBeanClass" %>** . . .
    - <jsp:useBean id="myBean" class="myBeanClass" scope="session"/>

JSP page encoding

Updated May 2008

- Prior to JSP 2.0, JSP pages in XML syntax determined their encoding by
  - Examining the pageEncoding or contentType attributes of their *page* directive,
  - Defaulting to ISO-8859-1
- JSP 2.0 encoding is controlled by the XML specification.
  - These JSP documents must be changed to include an appropriate XML encoding declaration prolog.
  - <?xml version="1.0" encoding="UTF-8"?>

JSP page encoding scope

- Page encodings are determined differently
  - In JSP 1.2, on a per translation unit basis
  - In JSP 2.0, on a per-file basis.

- Example: if a.jsp statically includes b.jsp, and a page encoding is specified in a.jsp but not in b.jsp,
  - In JSP 1.2 a.jsp's encoding is used for b.jsp,
  - In JSP 2.0, the default encoding is used for b.jsp.

JSP request.getAttribute() behavior

- JSP Engine in v5.1 and earlier
  - Casts the returned value to “String”
- JSP Engine in v6.0
  - No longer does the cast, returns “Object”
- Impacts those applications dependant on the old behavior
  - Fix available in 6.0.2.11 and later (PK20187)
  - Configurable settings
    - Webcontainer CustomProperty “com.ibm.wsspi.jsp.useStringCast ”
    - JSPAttribute in the extensions file (useStringCast)

JSP Tag Library change

- Change in how stringently tags are validated
- “teiclass” is now checked for valid class defintion

```
<tag> <name>StateFinder</name>
<tagclass>com.company.sample.tags.SomeFinder</tagclass>
<teiclass>empty</teiclass>
<bodycontent>Jsp</bodycontent>
<info>
...
```

- “empty” is now checked for existence, produces a warning message

JSP fragments and tags

Updated May 2008

- JSP fragments within an If-Else condition no longer compile
  - Used to assume that the same tag variables to be declared twice in an If-Else condition
  - Configurable setting
    - com.ibm.wsspi.jsp.usescriptvardupinit
    - PK29373 - v6.0.2.17, v6.1.0.5 and later
- Tag <tsx:repeat> using “index” attribute now produces “java.lang.Integer” instead of “int”
  - Configurable setting
    - com.ibm.wsspi.jsp.userepeatint
    - PK26741- v6.0.2.15, v6.1.0.9 and later

WebContainer changed settings

Updated May 2008

- The “+” sign is treated incorrectly in a URI
  - The plus sign character is incorrectly treated as a special character that needs to be decoded when it appears in the request URI.
  - Configurable setting
    - com.ibm.ws.webcontainer.decodeURLPlusSign
    - PK23481 – v5.1.1.11, v6.0.2.11 and later
- PathInfo omitted while redirecting by response.sendRedirect()
  - With this fix the extra path information until the last trailing slash will be appended to the uri and then redirected to the resource.
  - Configurable setting
    - com.ibm.ws.webcontainer.RedirectWithPathInfo
    - PK23779 – v6.0.2.13 and later

WebContainer configurable settings

Updated May 2008

- Different behavior for trailing "/" in uri.
  - In 5.x, if the default URI mapping for "/" is not overridden by an application, a request for a URI matching the context root of a web application that does NOT end with "/" will, as a convenience, redirect to "/"
  - Configurable setting
    - com.ibm.ws.webcontainer.redirectcontextroot
    - PK27974 – v6.0.2.15, v6.1.0.3 and later
- Default content-type setting on v6 is "text/plain“; in v5 is "text/html."
  - Configurable setting
    - com.ibm.ws.webcontainer.contenttypecompatibility
    - PK27527 – v6.0.2.13, v6.1.0.2 and later

Servlet URL requirement

- Slash prepend required for getResourcePath(…) or and **getResourceAsStream**(…) to avoid MalformedURLException
  - Support of Servlet 2.3 requirement
  - Even if your own in-house code makes correct usage of this method other 3rd party packages may not(ex: Apache Struts Framework)
  - Custom property in WebContainer avoids this
    - Property="prependSlashToResource"
    - Setting is global to all Applications running on a server

Java Server Faces (JSF)

New February 2008

- Java Server Faces is included in v6.0 and later
  - For v6.0 the level is JSF 1.0
  - For v6.1 the level is JSF 1.1
- This may conflict with other JSF enablers you have used in prior versions of WebSphere Application Server
  - One example is MyFaces
  - Either convert to use the shipped level of JSF
  - Or you can continue to use your JSF support in most cases by using classloader support
    - Set Classloader to PARENT_LAST

JSP reserved character sequence

- JSP reserved character sequence due to JSF inclusion
  - “The character sequence '#{' is now reserved by JSP. So If you are using '#{' in template text or as a literal in an attribute value for a taglib, the sequence will have to be escaped.”

Client datasource access

New
May
2008

- The client container support was added in v6.0
  - Prior to this you could look up a datasource from a non-server process
    - Required override a security setting in j2c.properties tag
  - In v6.0 and later, the expectation is to configure the datasource, and it's security data, in the client configuration tool.
    - The setting in the j2c.properties file is no longer used and there is no equivalent.

Restriction creating threads

New May 2008

- Applies to programs running in the Application server process
  - The Client container has no restrictions
- In v5.0 and v5.1 (J2EE 1.3) there was a restriction that you could not create threads in EJBs
  - Alternative is to use MessageDrivenBeans (MDBs)
- In v6.0 and later (J2EE 1.4) the restriction was clarified to include not being able to create threads in either the Web or EJB containers
  - Alternative is to use the WorkManager

Web Services SOAP change

New August 2008

- Additional SOAP message validation has been added in v6.0 and later
- May result in some messages that were originally invalid to no longer work
  - For example:
    - Bad XML: https://216.94.21.10/demo-api/services/TransactionInitiator
    - Should have been: <hdr1>https://216.94.21.10/demo-api/services/TransactionInitiator</hdr1>
  - Now results in: org.xml.sax.SAXException: WSWS3700E: Error: Non-whitespace character content
- The only alternative is to correct the invalid messages

JMS Support

- J2EE 1.3 applications run unchanged
  - Benefits can be found by upgrading
- J2EE 1.4 upgrade
  - Biggest change in in MDB descriptors
  - Activation Specifications instead of Listener Port
    - However some behavior may not be available via Activation Specifications. One popular one is the ability to start and stop Listener Ports
- Scripts that make use of MDB's *listener port retry count* must be changed to use *maximum failed deliveries*

WebSphere API migration changes

JNDI direct references

New November 2007

- JNDI direct references were deprecated in v6.0
- However, this has been reversed and the pattern is no longer deprecated
  - Starting with v6.0.2.19 and v6.1.0.5
- JNDI indirect references are the recommended pattern
  - Settings are created in application resource references
  - Another level of indirection that can be set instead of changing code
  - Some Security credential settings are defaulted using the JNDI direct pattern
- For more information see
  - https://www.ibm.com/support/pages/node/343893
  - https://www.ibm.com/docs/en/was/8.5.5?topic=tnd-troubleshooting-namespace-problems

WebSphere removed APIs

- Usually few or no impacts to applications
  - Admin (EARUtils only)
  - Als
  - Anttasks
  - Ras
  - Security
- PME components
  - Activity Session
  - AsyncBeans
  - Dynacache
  - ObjectPool
  - Scheduler
  - Userprofile

Copyright and Trademarks