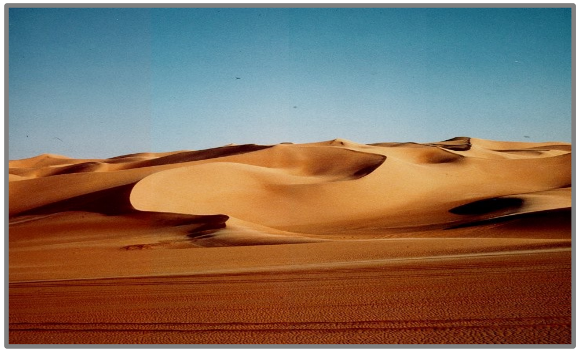
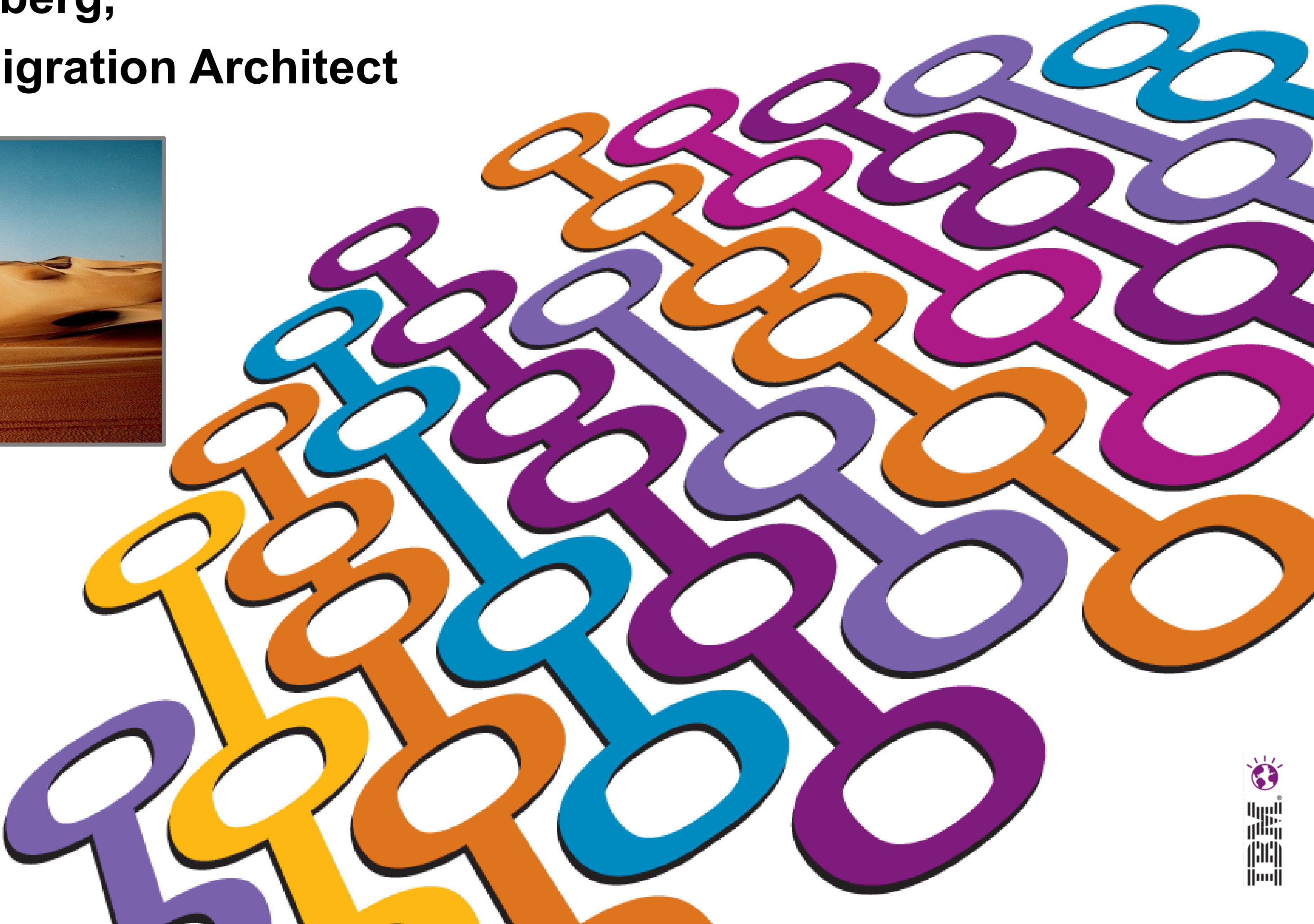
Shifting Sands?
Delivering WebSphere over multiple Java releases
- the Java perspective

Stephen A Hellberg,

IBM Java SE Migration Architect





Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

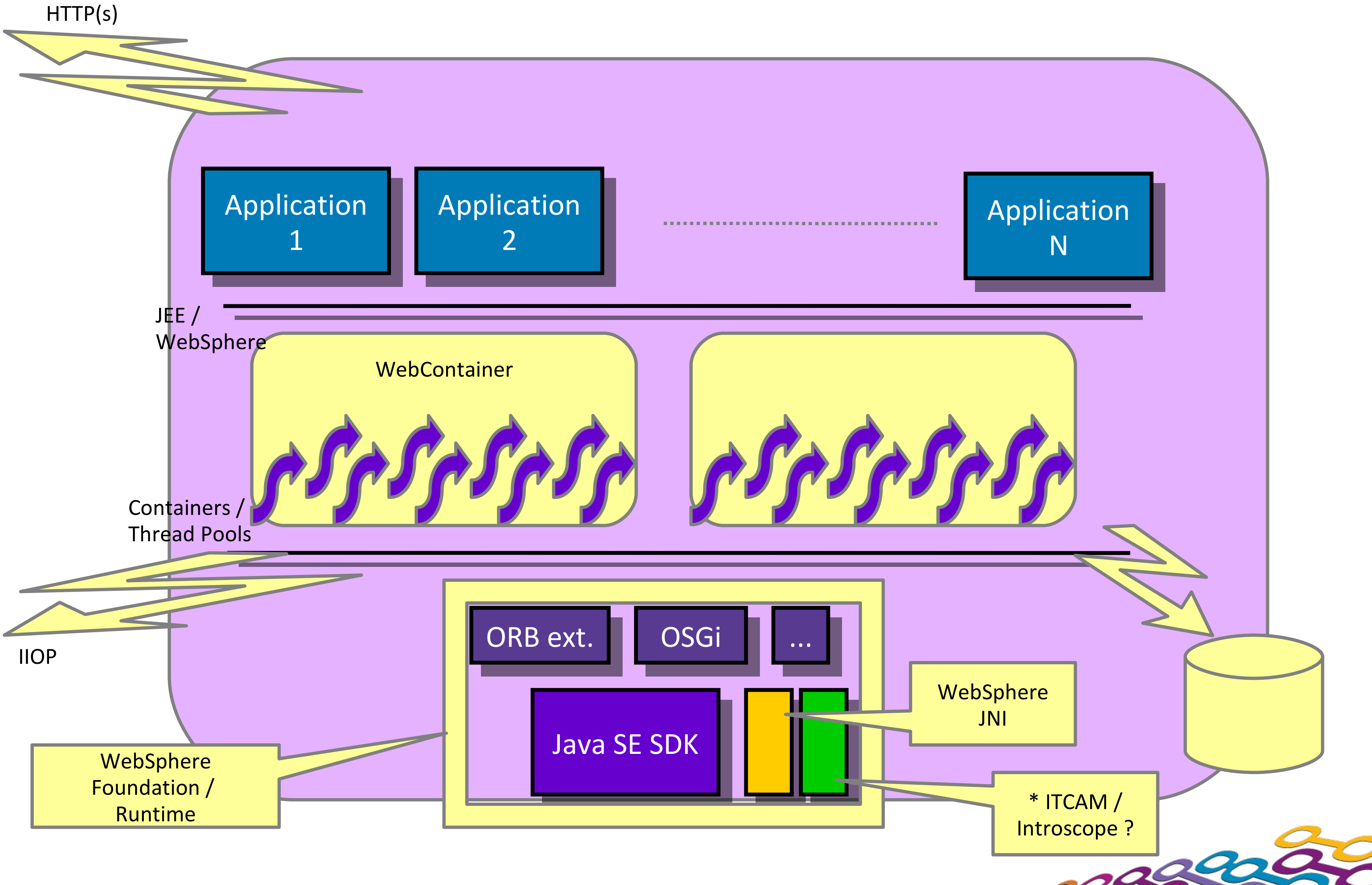
Agenda

- **WebSphere Runtime overview**
- **Migration overview**
    - **Timing**
    - **Motivation**
- Migration planning roadmap
    - Runtime contribution
- Overview of WebSphere changes by version
- Application (Code) Migration
    - Thematic Overview of Java SE changes by version
        - Java 6.0, Java 7.0
- Tuning / Performance
- Summary
- References

Introduction

- This presentation intends to educate and assist in performing Java-based WebSphere product Version-to-Version Migrations
  - WebSphere Application Servers (WAS), mostly
  - Other WebSphere-stack products
- Intends to alleviate concerns for your migration planning, and discusses some details for your awareness
- It does not prescribe one Migration path
  - Varies with customer policies
  - Varies with versions involved
  - Varies with customer procedures
- Use this information when building your own plan
- Get assistance if needed

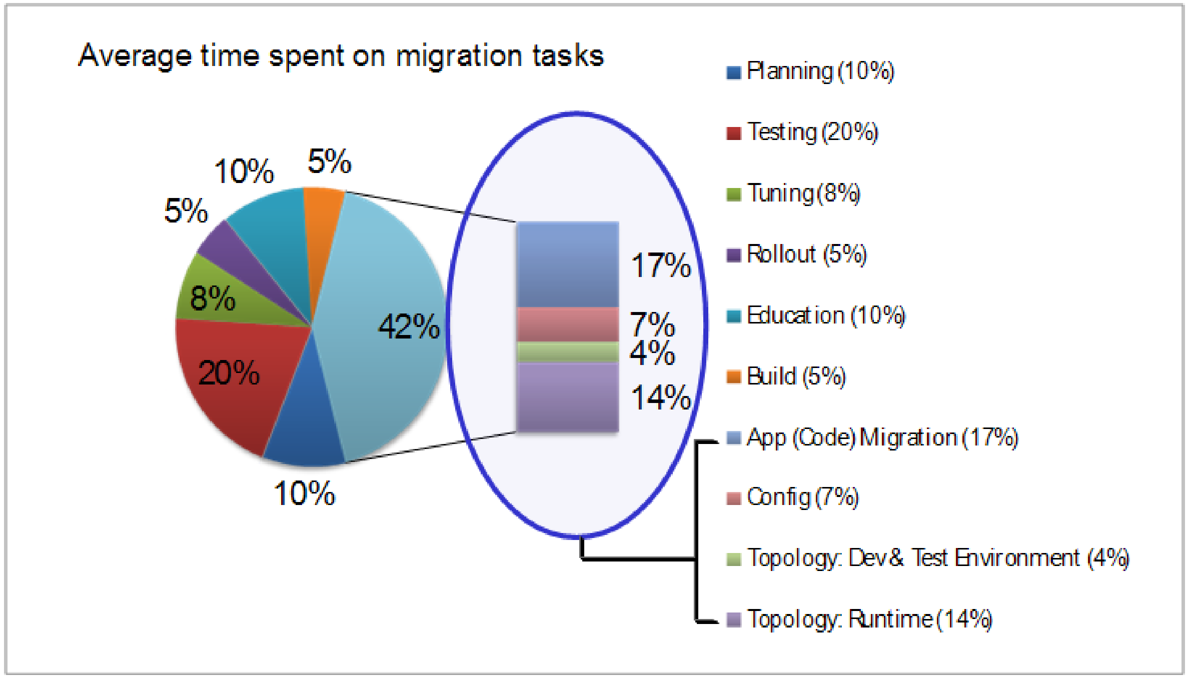
WebSphere Runtime Overview

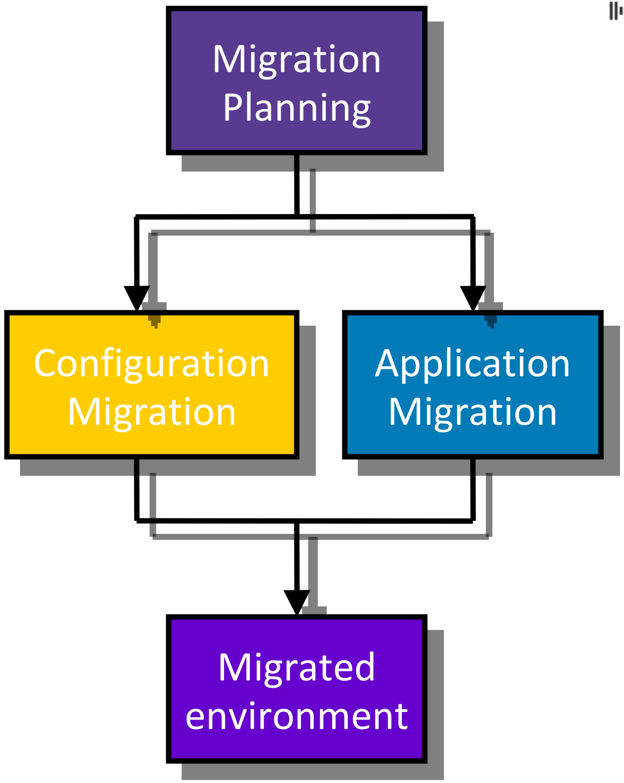


Agenda

- WebSphere Runtime overview
- Migration overview
  - Timing
  - Motivation
- **Migration planning roadmap**
  - **Runtime contribution**
- Overview of WebSphere changes by version
- Application (Code) Migration
  - Thematic Overview of Java SE changes by version
    - Java 6.0, Java 7.0
- Tuning / Performance
- Summary
- References

Migration Aspects





- The whole migration process involves a variety of planned steps
  - The scope of changes percolating up from the Runtime is not huge
- Tasks impacted: (by Java SE changes) ~25%
  - 'Topology:Runtime', **Application (Code) Migration**, Testing & Tuning

Agenda

- WebSphere Runtime overview
- Migration overview
    - Timing
    - Motivation
- Migration planning roadmap
    - Runtime contribution
- **Overview of WebSphere changes by version**
- Application (Code) Migration
    - Thematic Overview of Java SE changes by version
        - Java 6.0, Java 7.0
- Tuning / Performance
- Summary
- References

Choosing the “right” version

- Question: “Which WebSphere Application Server version should I migrate to?”

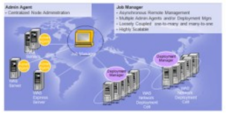
- Answer, as always - “It depends”
  - End of Service date for your current version
    - WebSphere Application Server v5.1 was September 2008
    - WebSphere Application Server v6.0.2 was September 2010
    - WebSphere Application Server v6.1 is September 2013
  - Characteristics of your targeted version
    - End of Service date for the version – most likely an estimate
    - Stability in lifecycle
    - JEE/JDK levels relative to targeted version
    - New WebSphere Application Server features you want
    - Version requirements of vendor or IBM products that you have or want
    - Version your Enterprise has committed towards
    - What about fixpack level? – is there an implicit company policy?

Comparing Versions 7.0, v8.0 and v8.5 – the factors

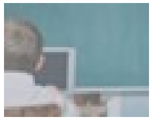
- End-Of-Life
  - Later versions will have a longer end-of-life. There is no specific way to estimate end-of-life, but clearly newer versions will be supported longer.
- Funtionality and Performance
  - Both v7.0 and v8.0 have the same JRE API level – JRE 1.6.
    However, v8.0 has shown some better performance characteristics, has JEE6 and various functional improvements over v7.0.
    Both JRE 1.6 and 1.7 are provided by v8.5 and also shows improved performance compared to v8.0 as well as JEE6.
- Stability
  - Many customers will wait for at least one fixpack before going live.
- Usage of Products with Specific WAS pre-req levels
  - THIS IS THE WILDCARD.
  - Many customers use other products/3rd party packages in their environments besides WAS. Those that are certified on particular WAS versions can cause planning challenges unless they support v8.0 or v8.5.
  - Some customers will be forced to go to an earlier version for this reason.
- Timing
  - Given the previous factors how would these factor into a project plan.

Why Migrate off WebSphere Application Server v6.x?

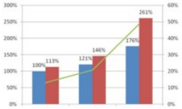
- Flexible Management: offloading administrative processes, centralized asynchronous administration in stand-alone and cell topologies
- Centralized Installation Manager (CIM): push installation of fix packs, interim fixes to nodes in the cell.
- v8.0: Expanded CIM capabilities, including installation across multiple cells and stand-alone servers without agents on remote nodes
- v8.0: IBM Installation Manager for all offerings on all platforms, including remote/local/custom repository installation, single-pass installation of product with iFixes and/or fix packs
- v8.0: Dynamic bundle updates and application extensions of running OSGi applications
- Properties file-based configuration utility
- v8.0: Monitored directory for install, update and uninstall of J2EE applications and modules with no tool or IDE required
- v8.0: Commands for template-based cell creation, transitioning nodes from one host and/or platform to another
- v8.0: High-performing binary logging and tracing with post-collection filtering, merging and formatting capabilities


Admin Agent
- Centralized Node Administration
Job Manager
- Asynchronous Remote Management
- Multiple Admin Agents and/or Deployment Mgrs
- Loosely Coupled: one-to-many and many-to-one
- Highly Scalable
Job Manager
WAS Server
WAS Express Server
Deployment Manager
WAS Network Deployment Cell
Deployment Manager
WAS Network Deployment Cell

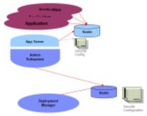
- Core programming model support: Java EE5, SIP, Portlet, v8.0: Java EE6, OSGi, SCA, XML, CEA and Java Batch.
- Feature packs: Web 2.0 and Mobile . v7.0 only: EJB3, Web Services, XML, CEA, SCA, Java Batch, OSGi
- WebSphere Business Level Applications (BLAs): lifecycle management of multi-component applications
- **Java 6 SE. Enhanced JVM with runtime enhancements and JIT optimizations.**



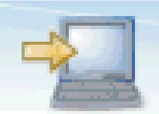
- **Performance gains in installation, startup, application deployment, runtime, cluster and server creation in large topologies**
- **Runtime provisioning: start only functions required by applications for memory and space efficiencies**



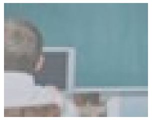
- New security auditing capabilities
- Kerberos authentication and Single Sign On
- WebSphere Secure Proxy
- Fine-grained administrative security: roles scoped to cells, node groups, nodes, clusters, servers, and applications. Filtered console.
- Multiple security domains: separate user/administrative domains at cell, server, cluster, or SIBus scopes.
  v8.0: Multiple security domain support for federated repositories (VMM).
- v8.0: Stronger default security settings
- v8.0: Distributed identity propagation for z/OS



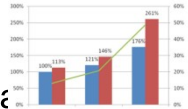
WebSphere Application Server Version v8.0 vs. v7.0



- IBM Installation Manager for all offerings on all platforms, including remote/local/custom repository installation, single-pass installation of product with iFixes and/or fix packs
- Expanded Job Manager and Centralized Installation Manager capabilities, including scheduled asynchronous installation across multiple cells and stand-alone servers without requiring agents on remote nodes



- Additional core programming models: Java EE6, OSGi, SCA, XML, CEA and Java Batch
- **Enhanced JVM with runtime enhancements and JIT optimizations**



- Performance gains in installation, startup, application deployment, runtime, cluster and server creation in large topologies
- z/OS application performance gains when combined with z196 hardwa tions
- z/OS RAS attribute enhancements

- Stronger default security settings
- Multiple security domain support for federated repositories (VMM)
- Distributed identity propagation for z/OS

- Monitored directory for install, update and uninstall of J2EE applications and modules with no tool or IDE required
- Dynamic bundle updates and application extensions of running OSGi applications
- Command for template-based cell creation
- Commands to transition nodes from one host and/or platform to another, and damaged node recovery
- High-performing binary logging and tracing with post-collection filtering, merging and formatting capabilities

```
addNode dmgr_host [dmgr_port]
[-asExistingNode]
…
```

- Liberty profile, WebSphere Virtual Enterprise and WebSphere Compute Grid capabilities in v8.5

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

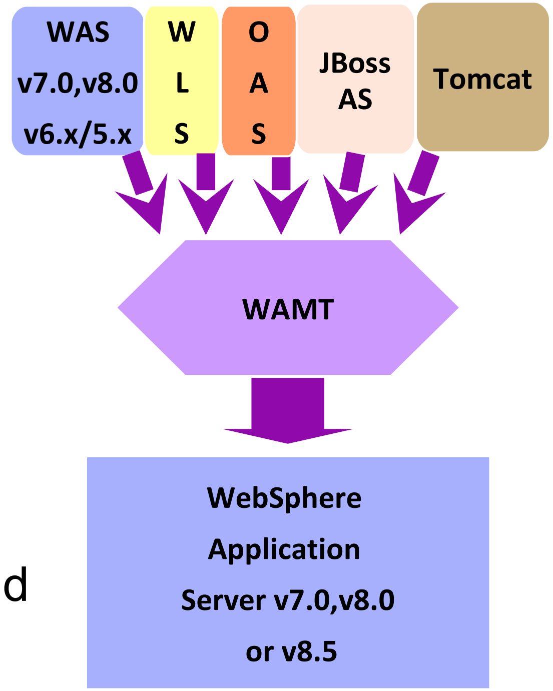
Operations and Control

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

WebSphere Application Migration Tool

- WAS Application Migration tool
  - Analyzes customer application source code to find migration problems:
    - Removed features
    - Deprecated features
    - Behavior changes
    - **JRE 5, 6 and 7 differences**
    - Java EE specification changes or enforcements
  - **Free** download and supported -> use in Rational and Eclipse development environments
    - v7.0: Support for v5.1 through v6.1
    - v8.0: Support for v5.1 through v7.0
    - v8.5: Support for v5.1 through v8.0
- Application Migration
  - <u>Few significant impacts in v8.0, **None** in v8.5</u>
    - No required change in Java Runtime specification level
    - Java EE6 upgrade in v8.0 but 84% of the sub-specification is unchanged



No more wading through documentation and source code or performing trial-and-error to determine what will break in your application!

Overview

- Delivered as an Eclipse feature
  - ✓ Installs on eclipse (Versions 3.4.2 –> 3.7 and 4.2)
  - ✓ Rational Application Developer Version 7.5+, 8.0.x+ and 8.5.x
  - ✓ With WebSphere Application Server Developer Tools for Eclipse

- Based on Rational Software Analyzer technology

- Current release is Version v3.5
  - ✓ Free Downloads
    - http://www.ibm.com/developerworks/websphere/downloads/migration_toolkit.html
  - – The WebSphere Version to Version tool is now available for install using Eclipse Marketplace.

WebSphere. software

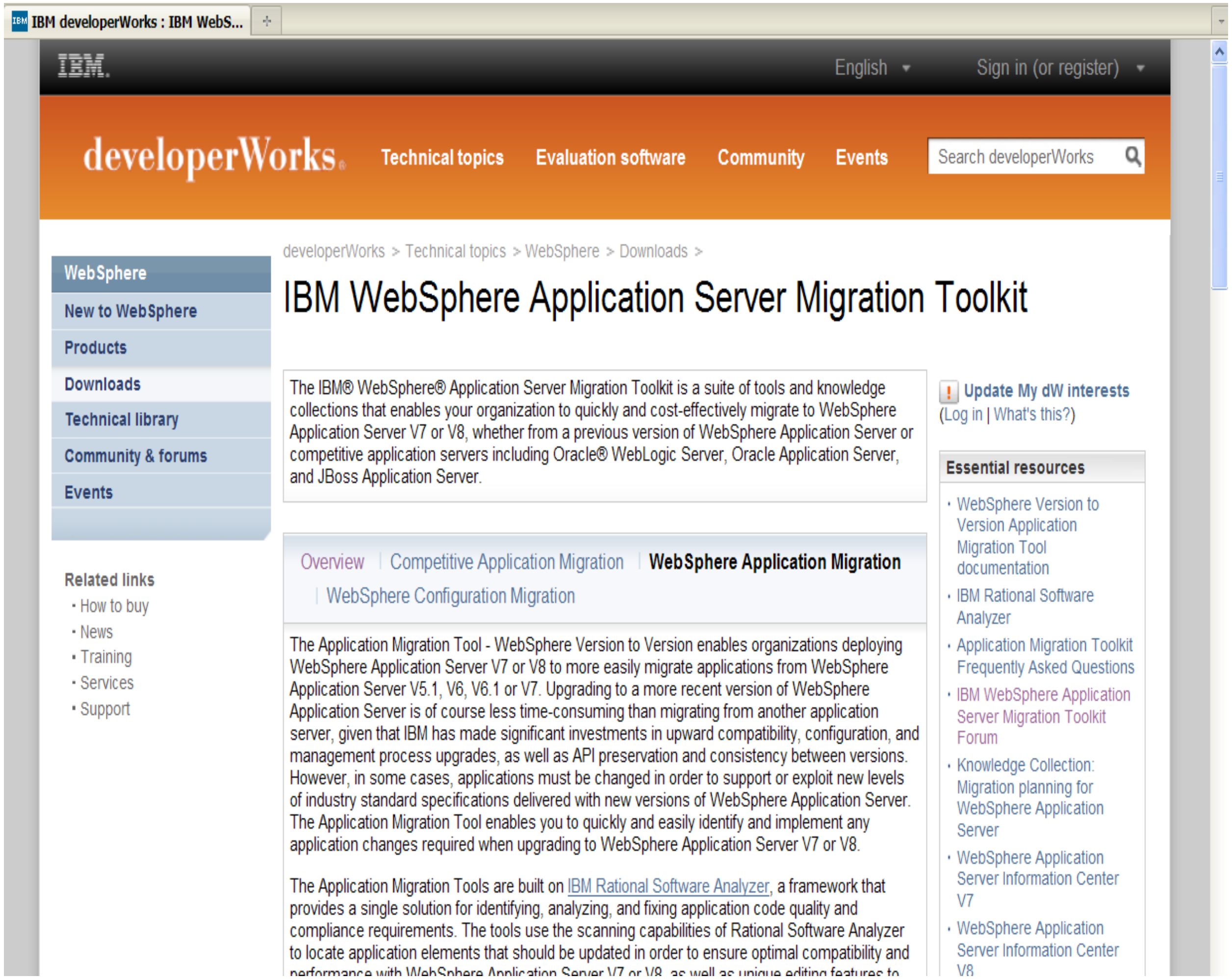
Rational Software Analyzer
Improve application health with automated code quality

Toolkit Installation

- **Download The Toolkit(s)**
  - ✓ Free Download
  - ✓ http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/index.html
  - ✓ Or Eclipse Marketplace for Version to Version

- **Select the support you want**

- **Install/Use an IDE**
  - ✓ Eclipse v3.4.2 - v3.7 and 4.2
  - ✓ Rational Application Developer (RAD) v7.5, v8.0, v8.5


IBM developerWorks : IBM WebS...
IBM.
English
Sign in (or register)
developerWorks
Technical topics
Evaluation software
Community
Events
Search developerWorks
WebSphere
New to WebSphere
Products
Downloads
Technical library
Community & forums
Events
Related links
How to buy
News
Training
Services
Support
developerWorks > Technical topics > WebSphere > Downloads >
Update My dW interests
(Log in | What's this?)
Essential resources
WebSphere Version to Version Application Migration Tool documentation
IBM Rational Software Analyzer
Application Migration Toolkit Frequently Asked Questions
IBM WebSphere Application Server Migration Toolkit Forum
Knowledge Collection: Migration planning for WebSphere Application Server
WebSphere Application Server Information Center V7
WebSphere Application Server Information Center V8
Overview
Competitive Application Migration
WebSphere Application Migration
WebSphere Configuration Migration

IBM WebSphere Application Server Migration Toolkit

The IBM® WebSphere® Application Server Migration Toolkit is a suite of tools and knowledge collections that enables your organization to quickly and cost-effectively migrate to WebSphere Application Server V7 or V8, whether from a previous version of WebSphere Application Server or competitive application servers including Oracle® WebLogic Server, Oracle Application Server, and JBoss Application Server.

The Application Migration Tool - WebSphere Version to Version enables organizations deploying WebSphere Application Server V7 or V8 to more easily migrate applications from WebSphere Application Server V5.1, V6, V6.1 or V7. Upgrading to a more recent version of WebSphere Application Server is of course less time-consuming than migrating from another application server, given that IBM has made significant investments in upward compatibility, configuration, and management process upgrades, as well as API preservation and consistency between versions. However, in some cases, applications must be changed in order to support or exploit new levels of industry standard specifications delivered with new versions of WebSphere Application Server. The Application Migration Tool enables you to quickly and easily identify and implement any application changes required when upgrading to WebSphere Application Server V7 or V8.

The Application Migration Tools are built on IBM Rational Software Analyzer, a framework that provides a single solution for identifying, analyzing, and fixing application code quality and compliance requirements. The tools use the scanning capabilities of Rational Software Analyzer to locate application elements that should be updated in order to ensure optimal compatibility and performance with WebSphere Application Server V7 or V8, as well as unique editing features to

Previous Releases

Release dates

- Version 1.0 released December 2009
  - ✓ Oracle WebLogic J2EE application migration
- Version 1.1 released April 2010
  - ✓ JBoss and JEE 5 support application migration
- Version 1.2 released July 2010
  - ✓ Additional rules for JBoss and WebLogic application migration
- Version 2.0 released September 2010
  - ✓ WebSphere Version to Version application migration (to v7.0)
- Version 2.1 released December 2010
  - ✓ Oracle applications server application migration
- Version 3.0 released June 2011
  - ✓ WebSphere Version to Version application migration (to v8.0)
- Version 3.0.1 released November 2011
  - ✓ Eclipse 3.7 and RAD 8.0.4 support, enhancements to Version to Version application migration (to v8.0)
- ✓ Version 3.5 released June 2012
  - ✓ WebSphere application migration (to v8.5) including support for Java 7

Complete control over the migration process. Knowledge-based migration assistance improves the reliability of the migration

Current Release

v3.5

- Support for v8.5 for all tools, including traditional WAS and Liberty
- Apache Tomcat Application Migration
- Java 7 rules
- Sun/IBM Java Differences
  - ✓ Migrate from Sun/HP Hardware to to any IBM JDK platforms
- Java Rules now available in all competitive tools
  - ✓ Previously only in version to version tool
- Frameworks (Spring, Hibernate, SEAM) - phase 1
- Dynamic Ruleset Generation
  - Much improved ease of use in selecting migration rule sets
- RAD v8.5 Support

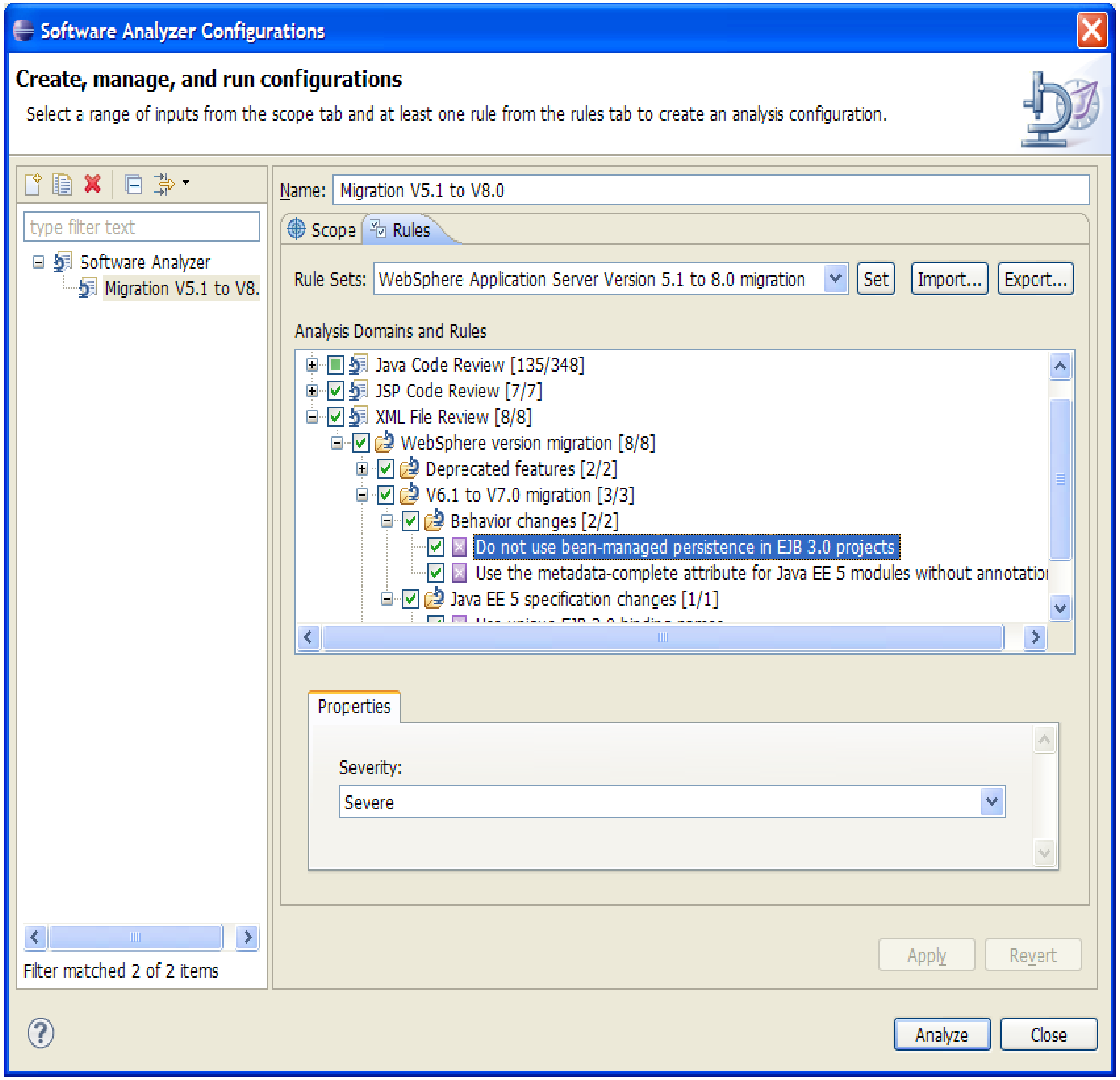
Application Migration

- How does it work?
  - ✓ Uses the concepts of rules, categories, code reviews (providers), and quick fixes to organize and manage the migration issues
    - Rules: Encapsulate the necessary logic to capture a migration issue (example: if the code contains a removed API com.ibm.* then…)
    - Categories: Organize the related rules.
    - Code Reviews: Manage artifacts that can be scanned (.java, .xml, .MF, .jsp files)
    - Quick Fix: Take action on a rule's result to fix the migration issue
  - ✓ Analyze (scan) the application source to find areas that require migration
  - ✓ Provide results with direct navigation to the source code or the deployment descriptor that needs to be updated
  - ✓ Provide multi-tier help on migration issues
  - ✓ Automatically change the source code and resolve the migration issue.
  - ✓ Generate reports to better estimate the scope of the required application changes

Application Migration

➡Flexible Migration Configuration

- ✓ Create multiple configurations
- ✓ Predefined Rule sets
- ✓ Import & Export Rule sets
- ✓ Adjustable Analysis scope
- ✓ Analysis scope can be selected from the Project Explorer as well.
- ✓ Integrated Help



Software Analyzer Configurations

Create, manage, and run configurations

Select a range of inputs from the scope tab and at least one rule from the rules tab to create an analysis configuration.

type filter text

- Software Analyzer
  - Migration V5.1 to V8.

Filter matched 2 of 2 items

Name: Migration V5.1 to V8.0

Scope | Rules

Rule Sets: WebSphere Application Server Version 5.1 to 8.0 migration | Set | Import... | Export...

Analysis Domains and Rules

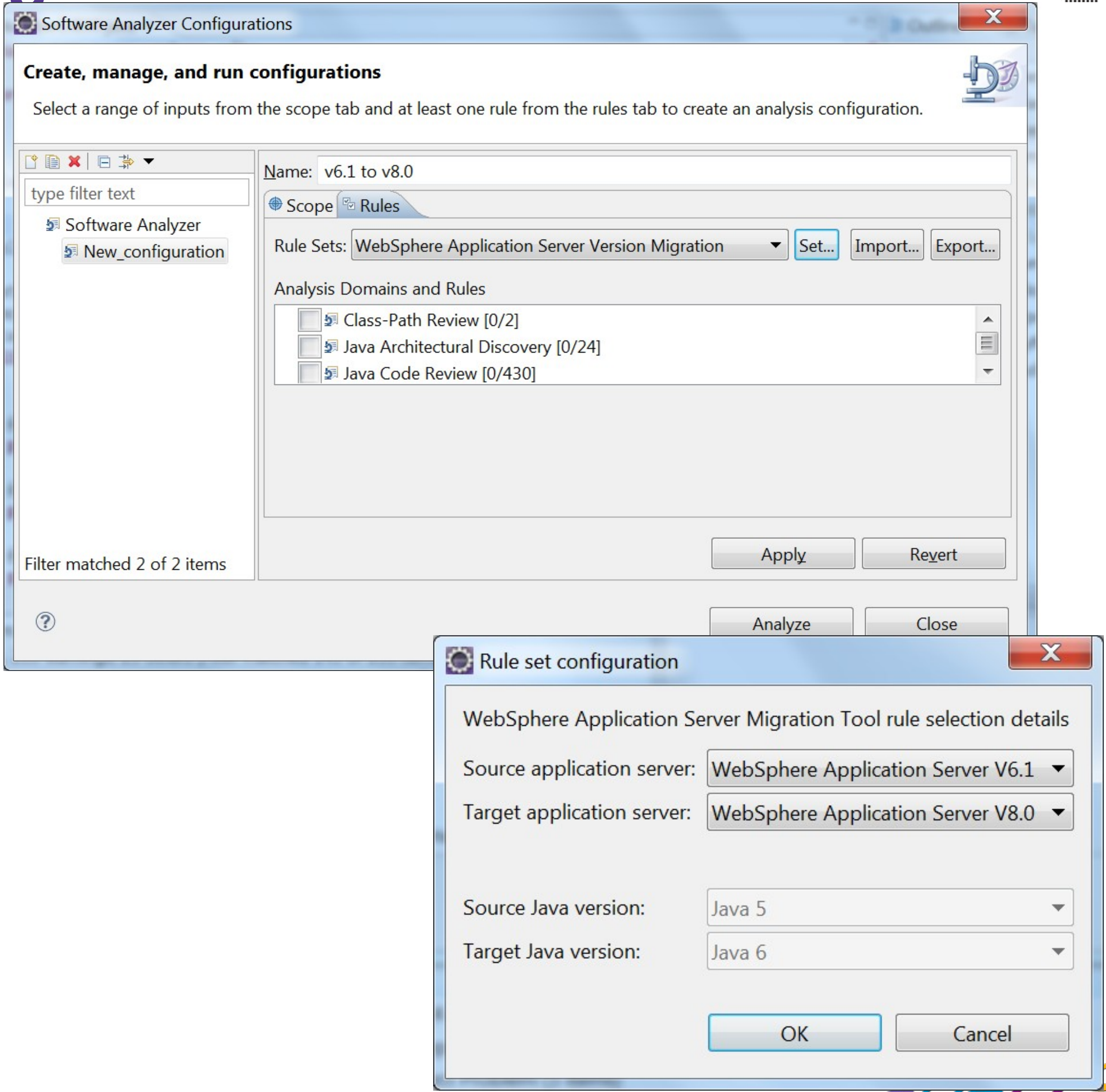
- ▣ Java Code Review [135/348]
- ☑ JSP Code Review [7/7]
- ☑ XML File Review [8/8]
  - ☑ WebSphere version migration [8/8]
    - ☑ Deprecated features [2/2]
    - ☑ V6.1 to V7.0 migration [3/3]
      - ☑ Behavior changes [2/2]
        - ☑ Do not use bean-managed persistence in EJB 3.0 projects
        - ☑ Use the metadata-complete attribute for Java EE 5 modules without annotatio
      - ☑ Java EE 5 specification changes [1/1]

Properties

Severity: Severe

Application Migration

- **WebSphere Rule set**
  - ✓ You can migrate
    - ✓ WAS v7.0
    - ✓ WAS v8.0
    - ✓ WAS v8.5
  - ✓ You can migrate (on WAS v8.5)
    - ✓ Java6
    - ✓ Java7



Software Analyzer Configurations

Create, manage, and run configurations

Select a range of inputs from the scope tab and at least one rule from the rules tab to create an analysis configuration.

type filter text

- Software Analyzer
  - New_configuration

Filter matched 2 of 2 items

Name: v6.1 to v8.0

Scope | Rules

Rule Sets: WebSphere Application Server Version Migration

Analysis Domains and Rules

- □ Class-Path Review [0/2]
- □ Java Architectural Discovery [0/24]
- □ Java Code Review [0/430]

Rule set configuration

WebSphere Application Server Migration Tool rule selection details

Source application server: WebSphere Application Server V6.1

Target application server: WebSphere Application Server V8.0

Source Java version: Java 5

Target Java version: Java 6

Java SE

Java SE support is included in all scenarios

Java SE Migration

- ✓ J2SE 1.5
- ✓ Java SE 6
- ✓ Java SE 7 – new in v8.5
- ✓ Some API changes
- ✓ Mainly behavior changes
- ✓ Specification clarifications
- ✓ New methods added to interfaces
- ✓ New runtime exceptions thrown

Scanned Files	Rules	Quick Fixes
Java SE 5	14	3
Java SE 6	4	0
Java SE 7	16	0
Sun to IBM Java compatibility	7	4

Application Migration

Install tool into Eclipse, RAD

- ✓ Can install multiple tools

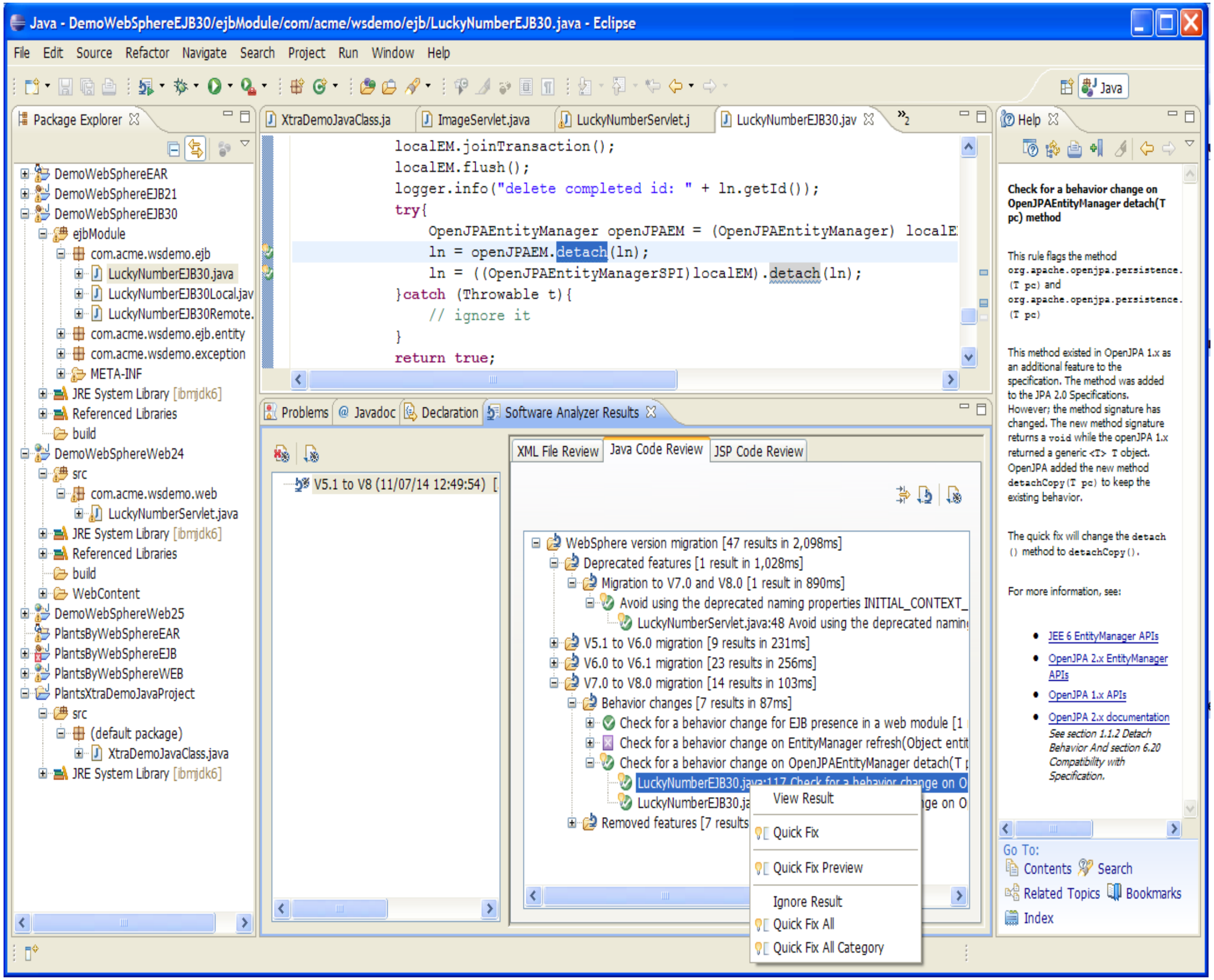
Access the tool once installed

Running the tool

- ✓ Analysis, getting help & Converting
  - Help
  - Quick Fix Review
  - Quick Fix
  - Quick Fix All

Direct navigation from the result to the source code

- ✓ Java
- ✓ XML
- ✓ JSP
- ✓ MF



```
DemoWebSphereEAR
DemoWebSphereEJB21
DemoWebSphereEJB30
  ejbModule
    com.acme.wsdemo.ejb
      LuckyNumberEJB30.java
      LuckyNumberEJB30Local.jav
      LuckyNumberEJB30Remote.
    com.acme.wsdemo.ejb.entity
    com.acme.wsdemo.exception
    META-INF
  JRE System Library [ibmjdk6]
  Referenced Libraries
  build
DemoWebSphereWeb24
  src
    com.acme.wsdemo.web
      LuckyNumberServlet.java
  JRE System Library [ibmjdk6]
  Referenced Libraries
  build
  WebContent
DemoWebSphereWeb25
PlantsByWebSphereEAR
PlantsByWebSphereEJB
PlantsByWebSphereWEB
PlantsXtraDemoJavaProject
  src
    (default package)
      XtraDemoJavaClass.java
  JRE System Library [ibmjdk6]
```

```
            localEM.joinTransaction();
            localEM.flush();
            logger.info("delete completed id: " + ln.getId());
            try{
                OpenJPAEntityManager openJPAEM = (OpenJPAEntityManager) localE
                ln = openJPAEM.detach(ln);
                ln = ((OpenJPAEntityManagerSPI)localEM).detach(ln);
            }catch (Throwable t){
                // ignore it
            }
            return true;
```

```
V5.1 to V8 (11/07/14 12:49:54) [
```

```
WebSphere version migration [47 results in 2,098ms]
  Deprecated features [1 result in 1,028ms]
    Migration to V7.0 and V8.0 [1 result in 890ms]
      Avoid using the deprecated naming properties INITIAL_CONTEXT_
        LuckyNumberServlet.java:48 Avoid using the deprecated namin
  V5.1 to V6.0 migration [9 results in 231ms]
  V6.0 to V6.1 migration [23 results in 256ms]
  V7.0 to V8.0 migration [14 results in 103ms]
    Behavior changes [7 results in 87ms]
      Check for a behavior change for EJB presence in a web module [1
      Check for a behavior change on EntityManager refresh(Object entit
      Check for a behavior change on OpenJPAEntityManager detach(T p
        LuckyNumberEJB30.java:117 Check for a behavior change on O
        LuckyNumberEJB30.ja                                    ge on O
    Removed features [7 results
```

Check for a behavior change on OpenJPAEntityManager detach(T pc) method

This rule flags the method `org.apache.openjpa.persistence.` `(T pc)` and `org.apache.openjpa.persistence.` `(T pc)`

This method existed in OpenJPA 1.x as an additional feature to the specification. The method was added to the JPA 2.0 Specifications. However; the method signature has changed. The new method signature returns a `void` while the openJPA 1.x returned a generic `<T> T` object. OpenJPA added the new method `detachCopy(T pc)` to keep the existing behavior.

The quick fix will change the `detach()` method to `detachCopy()`.

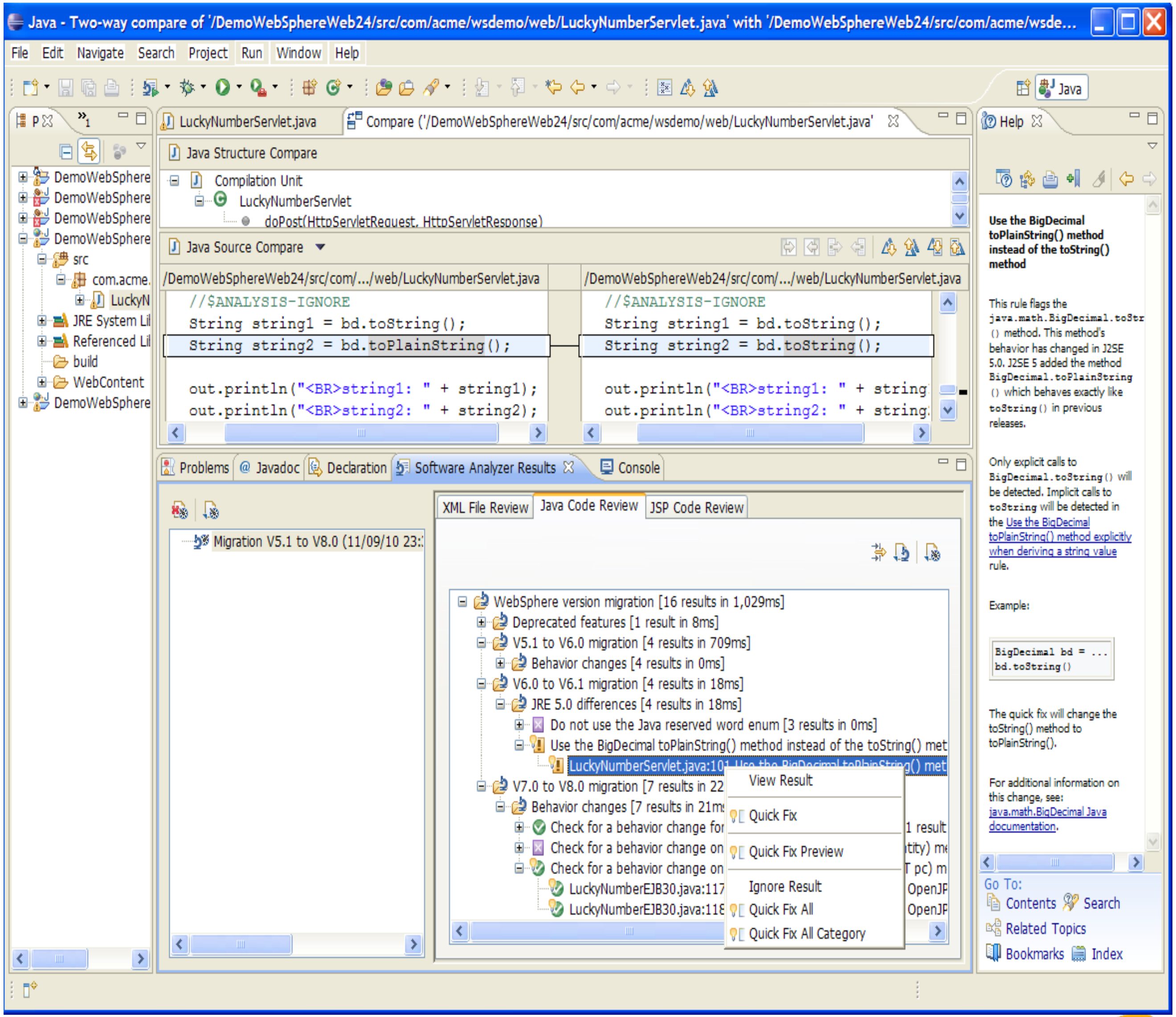
For more information, see:

- JEE 6 EntityManager APIs
- OpenJPA 2.x EntityManager APIs
- OpenJPA 1.x APIs
- OpenJPA 2.x documentation *See section 1.1.2 Detach Behavior And section 6.20 Compatibility with Specification.*

Application Migration

Two options for Quick Fix All

- ✓ Rule Level
- ✓ Category Level



```
//$ANALYSIS-IGNORE
String string1 = bd.toString();
String string2 = bd.toPlainString();

out.println("<BR>string1: " + string1);
out.println("<BR>string2: " + string2);
```

```
//$ANALYSIS-IGNORE
String string1 = bd.toString();
String string2 = bd.toString();

out.println("<BR>string1: " + string
out.println("<BR>string2: " + string
```

```
WebSphere version migration [16 results in 1,029ms]
  Deprecated features [1 result in 8ms]
  V5.1 to V6.0 migration [4 results in 709ms]
    Behavior changes [4 results in 0ms]
  V6.0 to V6.1 migration [4 results in 18ms]
    JRE 5.0 differences [4 results in 18ms]
      Do not use the Java reserved word enum [3 results in 0ms]
      Use the BigDecimal toPlainString() method instead of the toString() met
        LuckyNumberServlet.java:101 Use the BigDecimal toPlainString() met
  V7.0 to V8.0 migration [7 results in 22
    Behavior changes [7 results in 21ms
      Check for a behavior change for                    1 result
      Check for a behavior change on                     tity) me
      Check for a behavior change on                     T pc) m
        LuckyNumberEJB30.java:117                        OpenJP
        LuckyNumberEJB30.java:118                        OpenJP
```

Use the BigDecimal toPlainString() method instead of the toString() method

This rule flags the `java.math.BigDecimal.toString()` method. This method's behavior has changed in J2SE 5.0. J2SE 5 added the method `BigDecimal.toPlainString()` which behaves exactly like `toString()` in previous releases.

Only explicit calls to `BigDecimal.toString()` will be detected. Implicit calls to `toString` will be detected in the Use the BigDecimal toPlainString() method explicitly when deriving a string value rule.

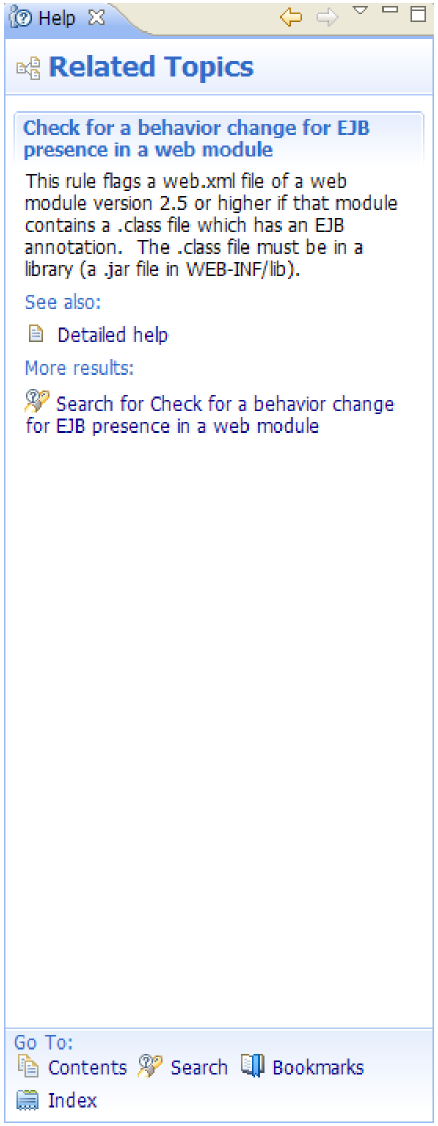
Example:

```
BigDecimal bd = ...
bd.toString()
```

The quick fix will change the toString() method to toPlainString().

For additional information on this change, see: java.math.BigDecimal Java documentation.

Application Migration



Help

Related Topics

Check for a behavior change for EJB presence in a web module

This rule flags a web.xml file of a web module version 2.5 or higher if that module contains a .class file which has an EJB annotation. The .class file must be in a library (a .jar file in WEB-INF/lib).

See also:

Detailed help

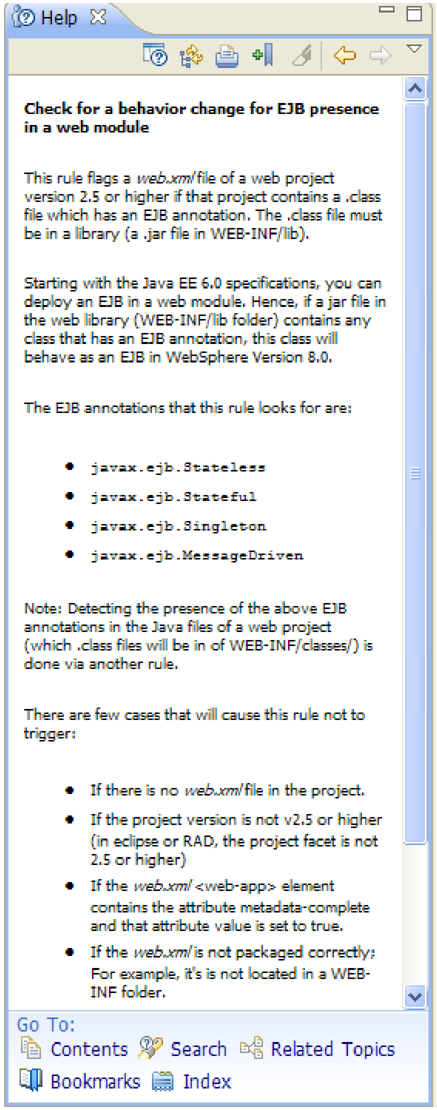
More results:

Search for Check for a behavior change for EJB presence in a web module

Go To:

Contents Search Bookmarks Index

- Help provided via eclipse help system
- With the rule selected hit F1
- Detailed help also provided for additional information



Help

Check for a behavior change for EJB presence in a web module

This rule flags a *web.xml* file of a web project version 2.5 or higher if that project contains a .class file which has an EJB annotation. The .class file must be in a library (a .jar file in WEB-INF/lib).

Starting with the Java EE 6.0 specifications, you can deploy an EJB in a web module. Hence, if a jar file in the web library (WEB-INF/lib folder) contains any class that has an EJB annotation, this class will behave as an EJB in WebSphere Version 8.0.

The EJB annotations that this rule looks for are:

- `javax.ejb.Stateless`
- `javax.ejb.Stateful`
- `javax.ejb.Singleton`
- `javax.ejb.MessageDriven`

Note: Detecting the presence of the above EJB annotations in the Java files of a web project (which .class files will be in of WEB-INF/classes/) is done via another rule.

There are few cases that will cause this rule not to trigger:

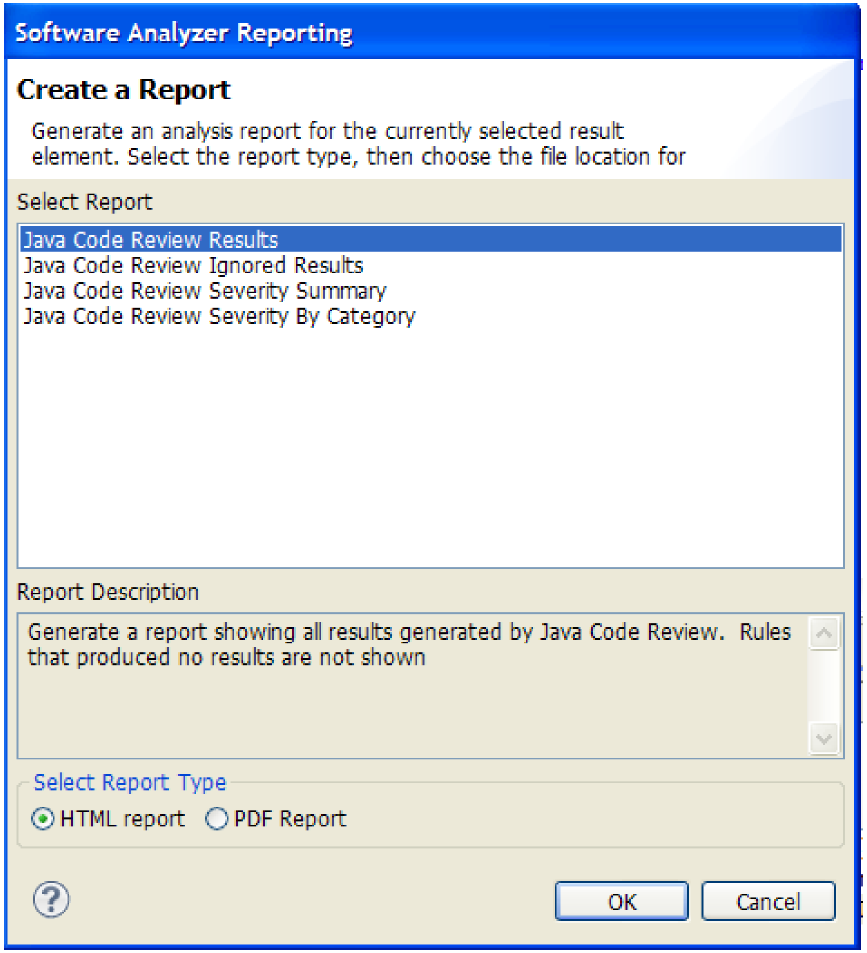
- If there is no *web.xml* file in the project.
- If the project version is not v2.5 or higher (in eclipse or RAD, the project facet is not 2.5 or higher)
- If the *web.xml* <web-app> element contains the attribute metadata-complete and that attribute value is set to true.
- If the *web.xml* is not packaged correctly; For example, it's is not located in a WEB-INF folder.

Go To:

Contents Search Related Topics Bookmarks Index

Application Migration

- **Generate Reports**
  - ✓ Java Code Review
    - PDF
    - HTML
    - XML
  - ✓ Other Code Reviews
    - XML



Software Analyzer Reporting

Create a Report

Generate an analysis report for the currently selected result element. Select the report type, then choose the file location for

Select Report

- Java Code Review Results
- Java Code Review Ignored Results
- Java Code Review Severity Summary
- Java Code Review Severity By Category

Report Description

Generate a report showing all results generated by Java Code Review. Rules that produced no results are not shown

Select Report Type

● HTML report ○ PDF Report

Agenda

- WebSphere Runtime overview
- Migration overview
  - Timing
  - Motivation
- Migration planning roadmap
  - Runtime contribution
- Overview of WebSphere changes by version
- **Application (Code) Migration**
  - **Thematic Overview of Java SE changes by version**
    - Java 6.0, Java 7.0
- Tuning / Performance
- Summary
- References

Java 5.0 / WAS v6.1 – 'Migration baseline'

- Java 5.0 was a very significant language revision, the most significant since Java 1.2 (i.e. J2SE definition)
    - Enumerations, Generics, Annotations
    - Concurrency package
    - Enhanced for-loops, Var-Args..., *etc.*
  - Significant for IBM Java releases, new J9 JVM also.
- Sun's declared intention: there would be no more 'minor' Java releases, i.e. no (1.)5.1
    - no repeat of 1.4.0, 1.4.1, 1.4.2
    - Java / J2SE new naming convention; 1.5.0 now '5.0'
- Java 8.0 was looking like the next significant language revision, of which Java 7.0 is the first (minor) installment...
  - … we'll see!

Java 5.0 → Java 6.0
WAS 6.1 → WAS 7.0 / 8.0 / 8.5

- Java 6.0 was primarily a 'client' computing release:
  - GUI improvements & JavaFX – the first attempt!
  - New-generation plugin
  - Core performance

- Java 6.0 delivered some 'server' computing related:
  - Enterprise / JEE significant...
    - JAX* 2.0 (JAXB, JAX-WS, etc.)
    - JDBC 4.0 revision
    - More Annotations (and their processing)
  - Lesser / Minor enhancements?
    - Programmatic API for Java Compiler, ...

Java Migration issues

- New functionality is just that.... New! (and often Immature)
  - It is likely not a cause/issue for older code that is migrating, except...
  - New language keywords – may conflict with existing variable names; fundamental conflict for compiling

- However, any issue that could require some kind of human intervention to analyse/resolve could be a *migration hurdle*
- In supporting migration, we are trying to minimise the efforts needed for each migration hurdle, if we cannot eliminate them.
- The “devil is in the detail”...

Java 6.0 Migration, an example

- Consider “JavaFX”*
  - Its first introduced with 6.0
  - Its intended as a UI replacement, but its an addition to existing UI support – Swing, *etc.* is still shipped
  - It will not prevent an existing Swing-based, say, application from still functioning if now executed on 6.0
- JavaFX is not a migration hurdle for continued use of your existing application code moving up to adopt Java 6.0 for execution

* *JavaFX is not a core component of Java SE.*
It was a proprietary extension shipped by Sun ; it is not implemented/delivered with IBM Java.

Java 6.0 Migration, an example

- Consider the New-generation plug-in
  - Its an implementation replacement for the previous plugin
  - It also shipped mid-service stream, as 6.0_10; not GA
    - Really Java 6.1?
    - Not discussed within major/minor version release notes
    - Not covered significantly in New Release documentation, or as a significant migration topic

- This is a potential source of J2SE Migration hurdle(s) for 'client' J2SE applications:
- It replaces a pre-existing capability/facility
- It (was) new, immature … source of risks!

WASMT Migration impacts (worst case scenario)

Potential Impact areas	v6.0	v6.1	v7.0	v8.0	v8.5
Java Runtime (v8.5 has JRE6 and 7)	n/a	6	2	n/a	0/20note
JEE - JSP	8	n/a	1	1	0
JEE - Servlet	5	n/a	0	1	0
JEE - Other	3	n/a	5	6	0
WAS Specific	1	6	0	4	0
3rd party packages	2	0	1	0	0
Development total	**19**	**12**	**9**	**12**	**0/20**
Administrative script	4	3	2	0	0
WAS directory structure	1	1	0	0	0
Other administrative	5	2	6	7	0
Total administrative	**10**	**6**	**8**	**7**	**0**
Total potential impact areas	**29**	**18**	**17**	**19**	**0/20**

Note: Java7 introduces a number of potentially breaking changes. “0” represents Java6 and “20” is for Java7. Not all breaking changes will impact all applications

Java 7.0

- Java 8.0 was looking like the next significant language revision, of which Java 7.0 is the first (minor) installment...

- Oracle's take-over of Sun MicroSystems (2010, agreed 2009)
- Oracle consulted on the future of Java after 6.0 – already in the field now for 4 years:

  - Java 7.0 now (almost) ; Java 8.0 major revision (2 yrs); or
  - Java 7.0 major revision (1-2+ yrs)

- The Java community opted for the former...

Java 7.0

- Java release cadence – every 2-3 years - significantly disrupted
- Java 7.0 delivered the latest stable Java 6.0 future revision...

- No **major** language changes
  - invokedynamic – new JVM bytecode instruction
    - First real change in JVM instruction set in 20 years!
    - Supports the new generation of JVM-based dynamic languages (Jruby, Scala, Jython, *etc.*)
    - ***NOT*** *exploited by the Java 7.0 language at all!*

- But, some long-overdue API revisions...

Java 7.0

- Some **minor** language changes (Project Coin)
  - Mostly, cosmetic programmer productivity:
  - <> Diamond Operator

```
Map<String, List<Trade>> trades = new TreeMap<String, List<Trade>> ();
```

becomes

```
Map<String, List<Trade>> trades = new TreeMap <> ();
```

  - Strings in switch statements
  - Automatic resource management
  - Multiple Exception types in `catch` blocks
- These changes, and others, implemented in revised '`javac`' compilation tool

Java 6.0 → Java 7.0
WAS 7.0 / 8.0 / 8.5 → WAS 8.5

- But, some long-overdue API revisions...
- JDBC 4.1 - Java may not have evolved, but the JDBC standard has!
- NIO 2.0
  - Specifically, assistance navigating FileSystems, *etc.*
  - Changes to some existing NIO behaviour
- AWT/2D/... UIs
  - Always a candidate for some tweaks! Metro support, *etc.*
- Other Misecellaneous...

WASMT Migration impacts (worst case scenario)

Potential Impact areas	v6.0	v6.1	v7.0	v8.0	v8.5
Java Runtime (v8.5 has JRE6 and 7)	n/a	6	**2**	n/a	0/2
JEE - JSP	8	n/a	1	1	0
JEE - Servlet	5	n/a	0	1	0
JEE - Other	3	n/a	5	6	0
WAS Specific	1	6	0	4	0
3rd party packages	2	0	1	0	0
Development total	**19**	**12**	**9**	**12**	**0/20**
Administrative script	4	3	2	0	0
WAS directory structure	1	1	0	0	0
Other administrative	5	2	6	7	0
Total administrative	**10**	**6**	**8**	**7**	**0**
Total potential impact areas	**29**	**18**	**17**	**19**	**0/20**

20

Note: Java7 introduces a number of potentially breaking changes. "0" represents Java6 and "20" is for Java7. Not all breaking changes will impact all applications

Java 6.0 → Java 7.0
WAS 8.5 → WAS 8.5

Developer Experience

- 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

- Shipping two fully supported JDKs:
  - Java 6.0
  - Java 7.0
- Available to facilitate your apps. migration at a time to suit you
- Its a one-way transition...

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

Most Java 7 (Project Coin) language benefits require re-expressing your code.

Java 7.0 → Java 8
WAS 8.5 → WAS <*next?*>

- Oracle's consultation means we actually know more about their plans for the next 'significant' Java release...
  - Due September 2013!
- Significant language developments:
  - Anonymous Function Literals / “Lamda” expressions
    - Closures – significant extensions for existing Runtime components, such as the Collections
  - Date & Time API revision
- … And, also some significant casualties:
  - Modularity,... <*other immature developments?*>

Agenda

- WebSphere Runtime overview
- Migration overview
    - Timing
    - Motivation
- Migration planning roadmap
    - Runtime contribution
- Overview of WebSphere changes by version
- Application (Code) Migration
    - Thematic Overview of Java SE changes by version
        - Java 6.0, Java 7.0
- **Tuning / Performance**
- Summary
- References

Tuning (aka Performance)

- The other significant phase concerning the Runtime

- Having migrated, you likely should consider re-Tuning

- You are crossing Major releases in multiple components (WAS, JDK, etc.)
  - Whilst much of the behaviour is the same/similar, we are always trying to improve it!

- Specifically, using later release JREs will include better (performance) JVMs, featuring improved JITs
- JVMs are co-developed with latest processor hardware, and hence able to benefit/exploit new capabilities

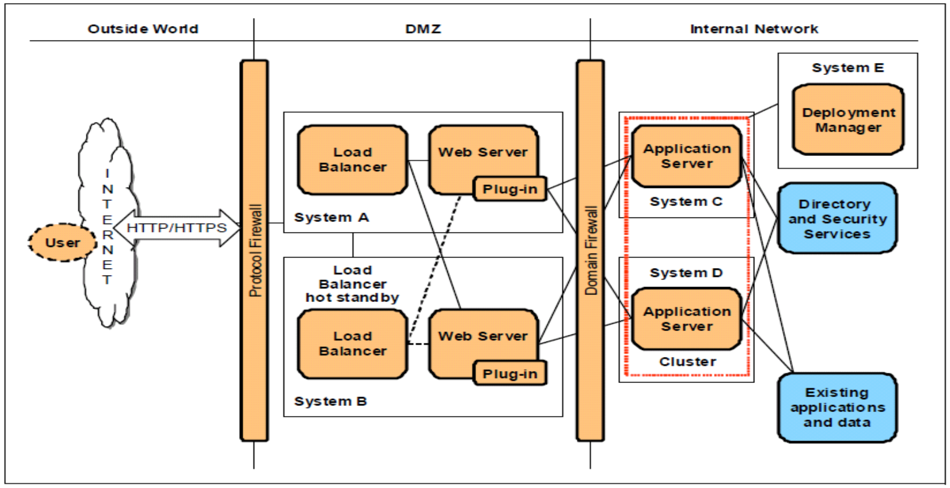
Tuning (aka Performance)

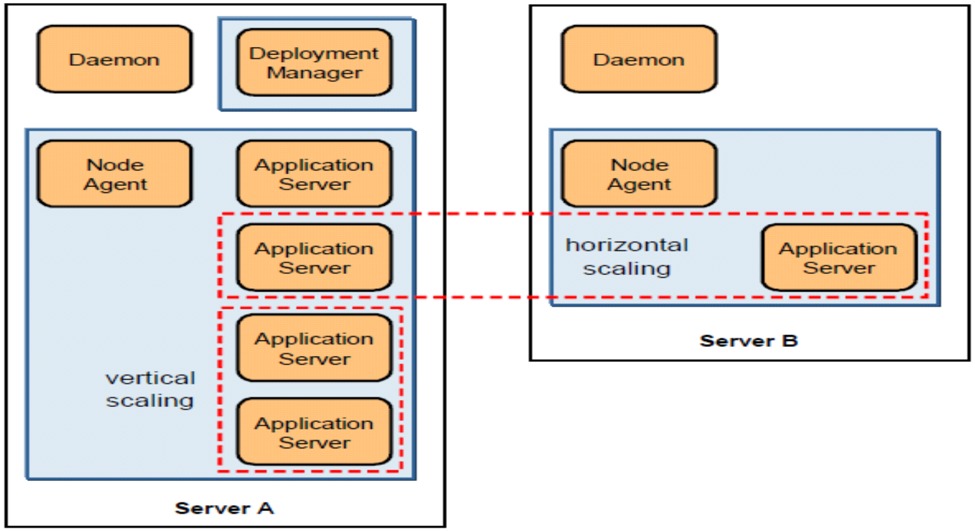
- Consider your 'platform'
  - What constitutes your WAS platform?
  - Server/OS? Changing?
  - User process type? (32-bit / 64-bit)
  - Clustered? Horizontal / Vertical

- For example: Migrating 32-bit Java 5.0 → 64-bit Java 6.0 w/ compressedrefs
  - Escape the restrictions of '2GB' heaps
  - Retain the benefit of 32-bit Java heap pointers/refs!
  - Retain benefits of L2 cache use!
  - Benefit from 64-bit hardware (x86_64, particularly)
  - Benefit from multi-core developments

WAS Deployment Topology

- How should applications be hosted in WAS?
  - Cell, Cluster, multi-tenant etc.
- Topology used is selected based on WAS Infrastructure design considerations
  - Workload Management
  - High-Availability
  - Scalability
  - Disaster Recovery
  - Security
  - Caching
  - Serviceability





Tuning (aka Performance)

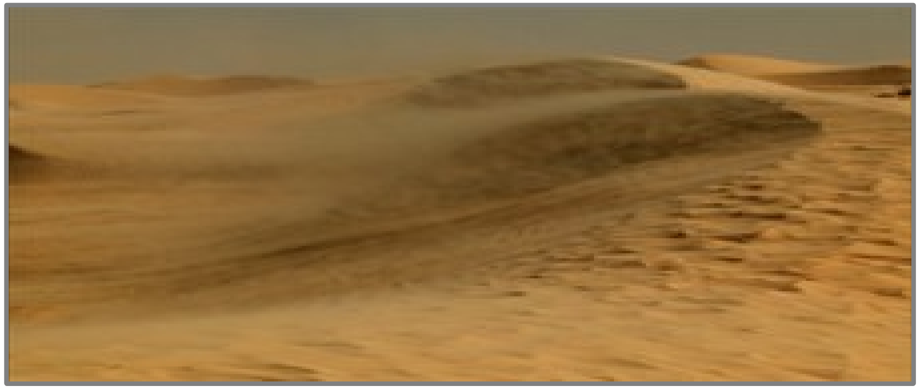
- Performance – not presently targetted for Migration focus
  - Too difficult to maintain the 'leap-frogging'
  - Too 'platform'-specific (server models, etc.)
  - Not suited to tooling based on IDEs

- Addressed by other tooling initiatives:
  - HealthCentre
  - Performance teams
  - Services
  - Support
- Nevertheless, don't ignore re-tuning... its an important phase to consider to get the best of your updated product!

Agenda

- WebSphere Runtime overview
- Migration overview
    - Timing
    - Motivation
- Migration planning roadmap
    - Runtime contribution
- Overview of WebSphere changes by version
- Application (Code) Migration
    - Thematic Overview of Java SE changes by version
        - Java 6.0, Java 7.0
- Tuning / Performance
- **Summary**
- References

Summary / Conclusions



- Java has excellent backwards compatibility

- Changes arising for major Java versions remain relatively few … causing **Migration concerns**

- We – IBM Java - are working with WebSphere Migration Toolkit developers delivering rules for base-level Java SE changes
  - Mostly concerned with WebSphere/JEE API changes
  - Application migration from other AppServers

- Your Application Code, and its dependencies, should be your only chief concern

Copyright and Trademarks

References

References

- IBM WebSphere Application Server Migration Toolkit
  - http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/
- WebSphere Application Server v8.5 Migration Guide
  - http://www.redbooks.ibm.com/abstracts/sg248048.html
- What's new in WebSphere Application Server v7.0
  - http://www.ibm.com/developerworks/websphere/library/techarticles/0809_alcott/0809_alcott.html
- What’s new in WebSphere Application Server v8.0
  - http://www.ibm.com/developerworks/websphere/techjournal/1106_alcott/1106_alcott.html
- What's new in WebSphere Application Server V8.5
  - http://www.ibm.com/developerworks/websphere/techjournal/1206_alcott/1206_alcott.html
- IBM Techdocs Whitepapers on WAS 7.0 (AIX) Migration case studies, including other IBM products
  - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101455

Resources

- WebSphere Application Migration Toolkit – Main Page on developerWorks
  - ✓ http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/index.html

- WebSphere Application Migration Toolkit Forum
  - ✓ http://www.ibm.com/developerworks/forums/forum.jspa?forumID=2106

- WebSphere Application Migration Toolkit Demo Video
  - ✓ http://www-01.ibm.com/software/webservers/appserv/wasmigrate.html

- Webcast replay: Simplifying Migrations with the WebSphere Application Server Migration Toolkit (focus on Version to Version Migration)
  - ✓ http://www-01.ibm.com/support/docview.wss?uid=swg27020784

- The product documentation (in PDF format) inside the toolkit downloaded archive
  - ✓ ApplicationMigrationTool_en_US.3.0.0.pdf