WebSphere Application Server
Planning for Version To Version Migration

Dana Duffield
WebSphere Application Server Migration Architect

WebSphere® Support Technical Exchange
Agenda

- Migration overview
- Migration roadmap
- Improving migration story
- Runtime migration support
- Version specific highlights
- Additional resources
Introduction

- This presentation is intended to educate and assist in performing WebSphere® Application Server Migrations.
- It contains overall planning guidelines as well as detailed migration concerns 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 as a guide to build your own plan.
- Get assistance if needed.
Overview

- Migration considerations
  - More than just software development
  - Must consider the applications, infrastructure, education and culture
  - Migration process should not compromise day-to-day business
  - Manage complexity, expectations, expense and risk

- Careful planning is required
  - Each situation is unique
  - There is no one standard plan
Migration Plan Roadmap

- Assessment
- Planning
- Skills
- Development Environment
- Application Code Migration
- Runtime Environment Migration
- Test
- Production
- Review the results
Assessment

- Gather the people
  - Consider a core Migration team
- Identify education requirements
  - Developer, Administrator…
- Hardware requirements
  - Possible Upgrades, All levels
- Topology assessment
  - Downtime tolerance, Failover support
- Application architecture
  - Tightened specifications
  - Dependencies between apps
  - API removal, JDK changes
- Review Testing practices
  - Standard practices and automation
- Vendor apps and WebSphere products
  - J2EE/JDK/WebSphere version requirements
Planning

- Build a plan based on assessment
  - Hardware requirements
  - Prerequisite software upgrades
  - Educational needs
  - Identify early adopters
  - Identify Pilot projects
  - Consider risk factors
  - Create an execution timeline
  - Include a rollback plan
Skills

- Plan for education
  - New development tooling
  - Changes in WebSphere administration model
  - Changes in the latest WebSphere version
  - New standards
Runtime Environment

- Most likely will need to support parallel development
- Migrate some test systems iteratively
  - Integration
  - System test
  - Performance
  - Pre-Production
- Based on your environment
Development Environment

- Likely require a change in IDE
  - Progress iteratively, expand outward

- Assume application compatibility
  - Assess apps, based on known issues
  - If no changes required, perform standard regression

- If development is required do it iteratively
  - Initially make changes that are required to support version migration
    - Reduces complexity of planning, diagnosis and debug - “Keep it Simple”
  - Test to the depth of test environment that fits your comfort level
  - Then do any necessary new code development and iterate following your standard practices

- Address Deprecations at some point
  - Ideally later as part of application updates
Test/Production/Review

- Run your standard test processes
  - Progress applications normally through the test environments
- Ensure Performance is measured
  - Differences exist between WebSphere versions
  - JDK changes may have occurred
- Have a rollback plan for production
  - Practice on another system earlier in the cycle
- Review the results of the Migration
  - Update the plan for next time
Section

Improving migration story
Administration

- Administration model changed starting in v5.0
  - Transition from DB repository to file based
  - Administration scripting tools changed
- Scripting model changed starting in v5.0
  - Stable scripting strategy starting v5.0
  - No changes moving from v5.0 to v5.1
  - Small number of changes when moving to v6.0 and v6.1
- Used to differ in migration tooling and support between Distributed, iSeries and zOS platforms
  - Improvement starting in v5.0, with same underlying migration tooling, support and experiences
## Administration impacts

<table>
<thead>
<tr>
<th>From\To</th>
<th>3.02</th>
<th>3.5</th>
<th>4.0</th>
<th>5.0</th>
<th>5.1</th>
<th>6.0</th>
<th>6.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Easier</td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Easier</td>
</tr>
<tr>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Easier</td>
<td>Easier</td>
</tr>
<tr>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>3.5</td>
<td></td>
<td>Easier</td>
<td></td>
<td></td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>3.02</td>
<td>Easier</td>
<td>Easier</td>
<td></td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>2.03</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
</tbody>
</table>
Administration scripting choices

- JACL or Jython?
- JACL is currently used more often
- Jython is the preferred direction
  - More industry support
  - Better performance
  - Priority in future deliverables
- No immediate need to convert from JACL
  - Deprecated in v6.1 but is still fully supported
  - Conversion tool included in v6.1 and available in Application Server Toolkit (AST) or via download (see speaker notes)
- Suggest an evolution
  - Create new scripts using Jython
Application Migration

- J2EE compatibility
  - J2EE 1.3 Implicitly supports J2EE 1.2
  - J2EE 1.4 Implicitly supports J2EE 1.2 and J2EE 1.3

- J2EE supports incremental upgrade
  - Modules within an application can be earlier versions

- In general WebSphere APIs very compatible
  - Many v4.x Applications run unchanged (except for zOS)
  - Most v5.0/v5.1 Applications run unchanged
Deprecation policy

- Deprecation policy in place for all WebSphere Application Server APIs
- API supported for 2 full releases from the point of deprecation or 3 full years whichever is longer
- API may be removed at such point
- Deprecations and Removals are documented in InfoCenter
- 5+3 support statement as of March 2007
  - Five for free, three for fee.
  - Updated April 2008 to include selected products within the Information Management, IBM Lotus, IBM Rational, IBM Tivoli and WebSphere
# Development impacts

<table>
<thead>
<tr>
<th>From\To</th>
<th>3.02</th>
<th>3.5</th>
<th>4.0*</th>
<th>5.0</th>
<th>5.1</th>
<th>6.0*</th>
<th>6.1*</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Easier</strong></td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Easier</strong></td>
</tr>
<tr>
<td>5.0</td>
<td></td>
<td></td>
<td>Easier</td>
<td></td>
<td></td>
<td>Easier</td>
<td><strong>Easier</strong></td>
</tr>
<tr>
<td>4.0&lt;sub&gt;z&lt;/sub&gt;</td>
<td></td>
<td></td>
<td>Varies</td>
<td>Varies</td>
<td>Varies</td>
<td>Varies</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td></td>
<td></td>
<td></td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>3.02</td>
<td>Easier</td>
<td></td>
<td></td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>2.03</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
</tbody>
</table>

*Note: 
- "Easier" for all other platforms for platform "z".
- Scenarios are difficult for this row."
# Impacts summary

<table>
<thead>
<tr>
<th>Development</th>
<th>v3.5</th>
<th>v4.0</th>
<th>v5.x</th>
<th>v6.0</th>
<th>v6.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisualAge for Java™ WebSphere Studio “Classic”</td>
<td>WebSphere Studio 5.0</td>
<td>WebSphere Studio 5.1</td>
<td>Rational Application Developer v6.0</td>
<td>Rational Application Developer v7.0</td>
<td></td>
</tr>
</tbody>
</table>

| Code | Pre-J2EE JDK 1.2 | J2EE 1.2 JDK 1.3 | J2EE 1.3 JDK 1.3 (v5.0) JDK 1.4 (v5.1) | J2EE 1.4 JDK 1.4 | J2EE 1.4 JDK 5 |

| Packaging and Deployment | Ad hoc Manual or scripted | J2EE packaging EAR deploy | J2EE packaging EAR deploy | J2EE packaging EAR deploy Configuration in the EAR | J2EE packaging EAR deploy Configuration in the EAR |

| Administration and Operations | Plug-in OSE Administration db WSCP, XMLConfig | Plug-in HTTPx Admin db for AE WSCP, XMLConfig | Plug-in HTTPx XML configuration JMX/TCL | Plug-in HTTPx XML configuration JMX/TCL | Plug-in HTTPx XML configuration JMX/TCL |
Choosing the “right” version

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

- **Answer:** “It depends”
  - End of Service date for your current version
    - For example [WebSphere Application Server v5.1 is September 2008](#)
  - End of Service date for the targeted version
  - Stability in lifecycle of targeted version
  - J2EE/JDK levels relative to targeted version
  - New WebSphere Application Server features you want
  - Version requirements of vendor or other WebSphere products that you have or want
  - The Version your Enterprise has committed towards
  - What about fixpack level? – do what makes the most sense…
Section

Runtime migration support
Tools and techniques

- Cross version plug-in support
  - Web Server plug-in can work to multiple WebSphere versions
    - The URI for a machine must be unique in the routing rules for the plug-in
    - E.g. v6.1 supports v5.0, v5.1, v6.0 and v6.1

- Coexistence
  - Running different versions of WebSphere on a machine at the same time
    - Requires port conflict resolution of concurrently running servers
    - Requires hardware capable of running all images
    - E.g. v6.1 supports v5.0, v5.1, v6.0 and v6.1
Tools and techniques...

- **Interoperability**
  - Different versions of WebSphere can communicate
    - Support for applications that are Secure, Transactional, EJB WLMable
    - E.g. v6.1 supports v5.0, v5.1, v6.0 and v6.1

- **Runtime migration tools**
  - Copy existing configurations between versions
    - zOS Customization or zMMT – create customized jobs
    - iSeries
      - Use commands directly
    - Distributed –
      - Migration Wizard
      - Or use commands directly

![Diagram](image-url)
Tools and techniques…

- v6.1 Mixed cell support
  - Nodes at different versions are supported in the same cell.
  - Different OS platforms are supported in the same cell.
  - Deployment Manager must always be at highest version and PTF level.
  - Must use Runtime migration tools to create mixed version cell for v5.0 and v5.1 nodes.
  - v6.0.2.x nodes can be added directly.
  - Supports v5.0, v5.1, v6.0 and v6.1.
  - Some limitations exist:
    - Cannot add v5.x nodes directly.
    - Cannot use some re-deployment when installing applications.
Three Network Deployment strategies

1. **Manual**
   - Ignores the existing configuration
   - Create a new cell and populate with administration scripts or manually

2. **Automated with whole node upgrade**
   - Recreates the *exact* v5.x/v6.0 configuration in v6.1
   - All applications on managed nodes are migrated at the same time

3. **Automated with mixed version utilization**
   - Recreates the *exact* v5.x/v6.0 configuration in v6.1
   - Add v6.1 nodes to the cell
   - Applications can be moved incrementally

More details on these…
Manual

- Ignores the existing configuration
- Create a new cell and populate with administration scripts or manually
- Best results with a comprehensive set of scripts

Pros
- No dependencies on tooling
- Least risk assuming existing scripts are comprehensive
- Can easily migrate applications singly

Cons
- Comprehensive set of scripts and ongoing maintenance of those scripts can be expensive
- Any required changes to these scripts must be done before migrating
- Any tuning of the old configuration is not carried forward
Automated with whole node upgrade

- Use Runtime migration tools on DMgr
  - Recreates the **exact** v5.x/v6.0 configuration in v6.1
- Later migrate the existing v5.x/v6.0 nodes using the runtime migration tools
  - All applications on a managed node are migrated at the same time

**Pros**
- Does not require comprehensive set of scripts
- All configuration is moved forward

**Cons**
- Dependency on using the runtime migration tools
- Requires all applications on a node be ready to migrate at the same time
- Limited value if you are refactoring your topology
- Does not enable some upgraded features
Automated with mixed version utilization

- **Use Runtime migration tools on DMgr**
  - Recreates the **exact** v5.x/v6.0 configuration in v6.1

- **Add new v6.1 nodes**
  - Applications can be migrated singly when ready
  - Remove old nodes when no longer needed

- **Pros**
  - Does not require comprehensive set of scripts
  - All configuration is moved forward
  - Cell level accessible by v6.1 nodes

- **Cons**
  - Dependency on using the runtime migration tools
  - Limited value if you are refactoring your topology
  - Does not enable some upgraded features
Section

Version specific Highlights
Moving to v6.1 highlights

- Unique to v6.1
  - Development tool overview
  - WebSphere removed features
  - JDK 5 impacts
  - Administration script required changes
  - Migration and Feature Packs

- Common to v6.0 and v6.1
  - Development tool overview
  - J2EE 1.4 impacts
  - V6.0 WebSphere API migration details
  - Administration script required changes
  - Profiles
  - Port usage
  - CoreGroup considerations
Migration and Feature Packs

- 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
Enablement

- This information and more is now available online!
- Includes general planning with detailed notes and WebSphere AppServer version specific information
- Updated with timely information
- IBM Migration Knowledge collection
Summary

- Migration needs to be a pragmatic, well designed and repeatable process

- WebSphere Application Server migration is becoming easier!
  - More tools
  - More techniques
  - Minimizing required Application changes
Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/

- Learn about other upcoming webcasts, conferences and events: http://www.ibm.com/software/websphere/events_1.html

- Join the Global WebSphere User Group Community: http://www.websphere.org

- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant


- Sign up to receive weekly technical My support emails: http://www.ibm.com/software/support/einfo.html
Questions and Answers