WebSphere Version to Version Migration: Planning a Successful Upgrade

—

Liam Westby
WebSphere Development

Introduction

This presentation covers migrating traditional WebSphere profiles from one version to a later version.

It contains overall planning guidelines and migration concerns.

It does not prescribe one migration path

- Varies with customer policies
- Varies with versions involved
- Varies with chosen procedures

Migration Overview

Choosing which version to migrate to

Depends on what version you are currently running.

Default ”best” answer is v8.5.5.

For currently support releases:

- Both WebSphere v8.5 and v9.0 will be supported through at least 2030.
- You don’t need to migrate from v8.5 to v9.0 for support reasons. Migrate only if you want to take advantage of additional capabilities in v9.0 (Java EE 7, etc.)

For extended support or out of support releases:

- We recommend that you migrate to a mainstream supported release when you are able.
- Whether you migrate to v8.5 or v9.0 depends on a variety of factors.
  - v8.5.5 will have fewer differences than v9.0.
  - v9.0 has the more recent programming model (Java EE 7.)

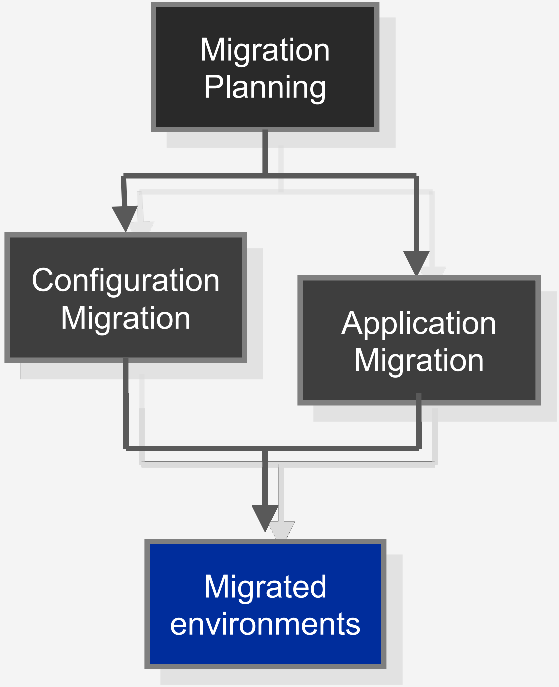
Migration process

The migration process involves a variety of steps.

- Plan the migration
- Analyze your applications
- Bring your configuration up to the new version
- Verify the environment is working

How long each of these steps takes depends on the size and complexity of your environment.

Planning is essential for coordinating between teams responsible for each step.



Application Migration

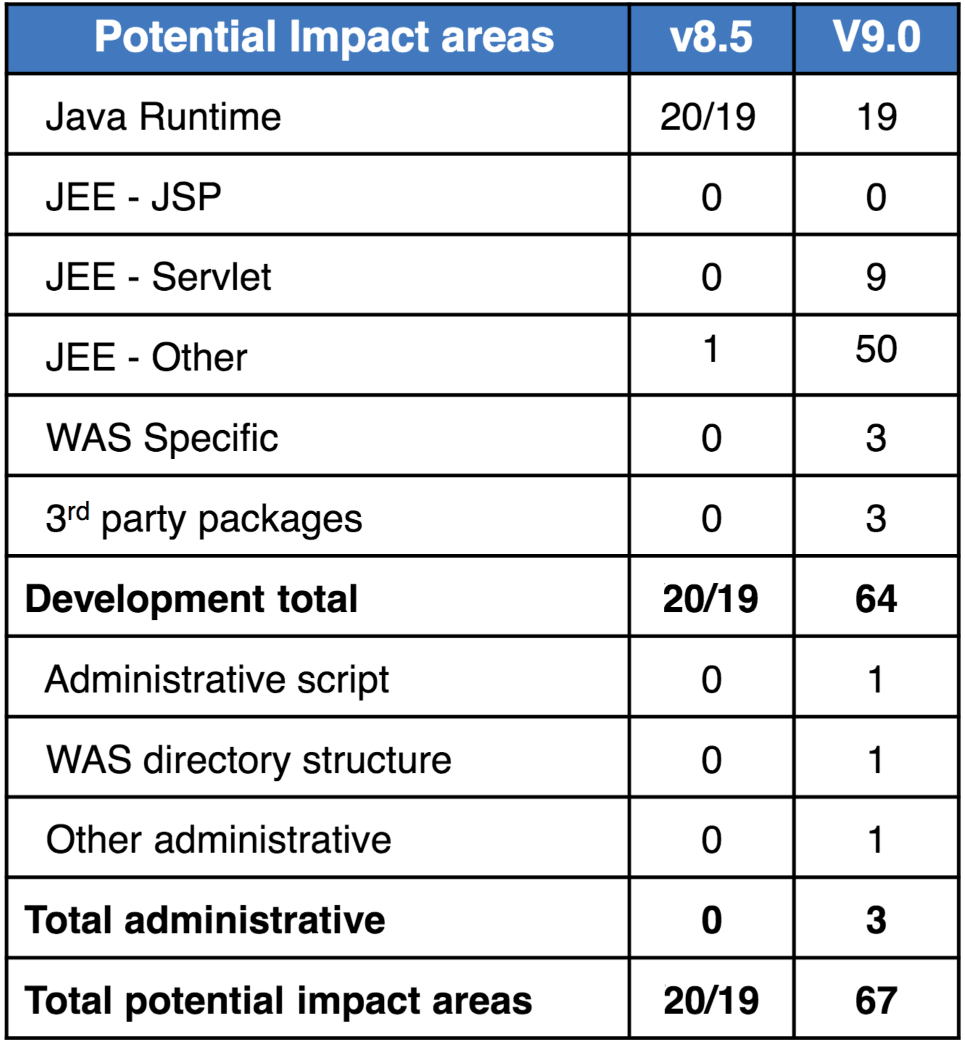
Application migration

When you adopt a new version of WebSphere, you may also be adopting a new level of Java SE or Java EE as well.

Application Migration means determining compatibility changes that might affect your application.

These changes typically require application code changes to resolve.

While development teams are needed to make these changes, detecting possible issues is something that operations can perform.



Potential Impact areas	v8.5	V9.0
Java Runtime	20/19	19
JEE - JSP	0	0
JEE - Servlet	0	9
JEE - Other	1	50
WAS Specific	0	3
3rd party packages	0	3
Development total	**20/19**	**64**
Administrative script	0	1
WAS directory structure	0	1
Other administrative	0	1
Total administrative	**0**	**3**
Total potential impact areas	**20/19**	**67**

Application changes can result from

JRE version differences

Java EE spec enforcements

Java EE spec clarifications

Deprecated APIs and features

Removed APIs and features

Behavior changes

Causes of application changes

Java Runtime compatibility (JRE)

- JREs focus on binary compatibility and are normally low impact
- New APIs and behavior changes can cause migration work

Java EE (JEE) compatibility

- Newer JEE versions intend to support older JEE versions
- But in some cases breaking clarifications do exist
- JEE supports incremental upgrade
- Modules within an application can be earlier versions

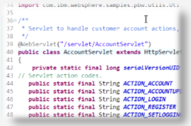
In general WebSphere APIs are very compatible

- Most interfaces after v5.0 are unchanged, some deprecated, fewer removed

However, no guarantee that applications will run unchanged

Migration tools for developers

Source

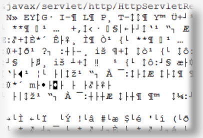


```
import com.ibm.websphere.samples.pbw.utils.Uti

/**
 * Servlet to handle customer account actions,
 */
@WebServlet("/servlet/AccountServlet")
public class AccountServlet extends HttpServlet
{
    private static final long serialVersionUID
// Servlet action codes.
  public static final String ACTION_ACCOUNT
  public static final String ACTION_ACCOUNTUPD
  public static final String ACTION_LOGIN
  public static final String ACTION_REGISTER
  public static final String ACTION_SETLOGGIN
```

- WebSphere Application Migration tool (http://ibm.biz/DownloadWASMigTools)
  - Eclipse-based tooling
  - Version to Version – including Liberty
  - Third-party migration support from JBoss, WebLogic, Oracle, Tomcat
  - Cloud Migration Tool

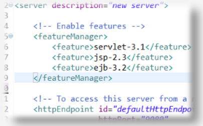
Binaries



```
javax/servlet/http/HttpServletRe
```

- Migration Toolkit for Application Binaries (http://ibm.biz/DownloadWASMigTools)
  - Command line tooling
  - Version to Version (V85+) and Liberty
  - Cloud migration rules
  - WebSphere configuration to Liberty server config or WebSphere Base edition wsadmin scripts

Aggregation and Modernization



```
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>servlet-3.1</feature>
        <feature>jsp-2.3</feature>
        <feature>ejb-3.2</feature>
    </featureManager>

    <!-- To access this server from a
    <httpEndpoint id="defaultHttpEndpo
                  httpPort="9080"
```

- Transformation Advisor (http://ibm.biz/cloudta)
  - Powered by the Migration Toolkit for Application Binaries
  - Focus on entire cell environment
  - Get estimated developer time cost to modernize
  - More concerned with application modernization than just version to version

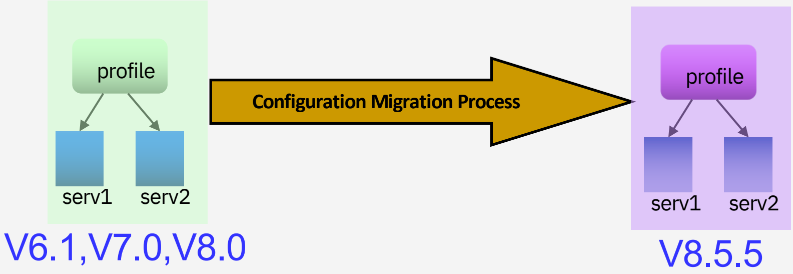
Configuration Migration

Configuration migration

Adopting a new major version of WebSphere requires creating a new installation and new profiles on each host machine.

Configuration migration is the process of recreating each profile from the current environment in the newer installation.

- Servers and topology
- Security
- Resources
- Variables
- Shared libraries
- Drivers
- … everything!



How to migrate configuration

If you have automated scripting, keep using that.

If you have a very small environment, set up the new environment manually.

Some of the advice on how to transition to the new environment still applies here, so stay tuned.

For most other circumstances, we recommend using migration tooling to assist you.

Specifically, we recommend using WASPreUpgrade and WASPostUpgrade.

WASPreUpgrade and WASPostUpgrade

Included as part of WebSphere.

Run from the new version of WebSphere that you are migrating to.

Focus on migrating one profile at a time:

- Each profile is saved into a backup directory by WASPreUpgrade, then merged from the backup directory into the destination profile by WASPostUpgrade
- To migrate an entire Network Deployment cell, migrate each profile individually starting with the deployment manager.

Allows for different approaches to migration:

- Choose on a profile by profile basis whether to migrate to a new install on the same host machine or to a new host machine altogether.
- (migrations to version 9.0 only) Choose whether to clone the cell so it can coexist with the older environment or replace the existing environment.

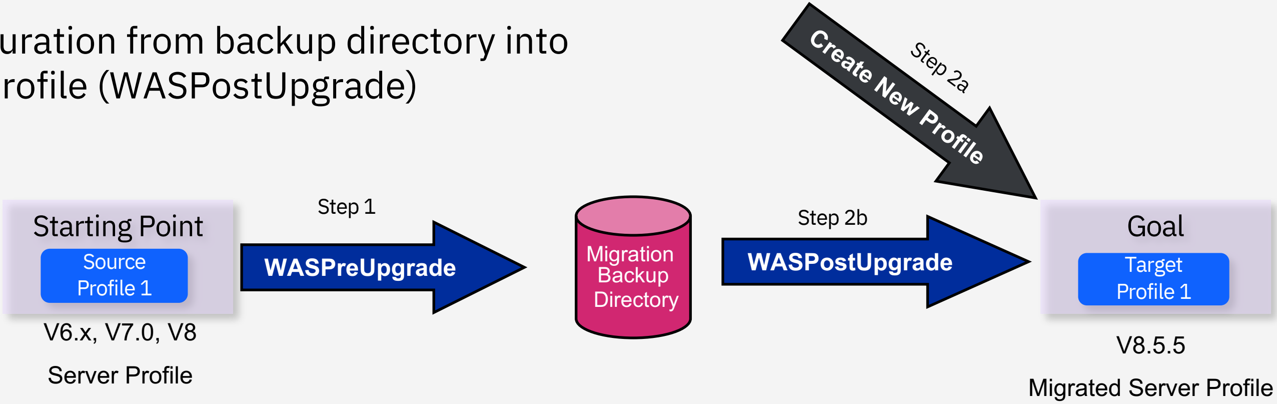
Goal is to get the new environment working the same as the old environment, while enabling new features where possible.

Local migration or remote migration?

Local migration: both the source and destination profile exist on the same host machine.

Two phases:

- Collect backup directory on local filesystem (WASPreUpgrade)
- Merge configuration from backup directory into destination profile (WASPostUpgrade)



Local migration or remote migration?

Remote migration: profile is upgraded and transferred to a new host machine via the backup directory.

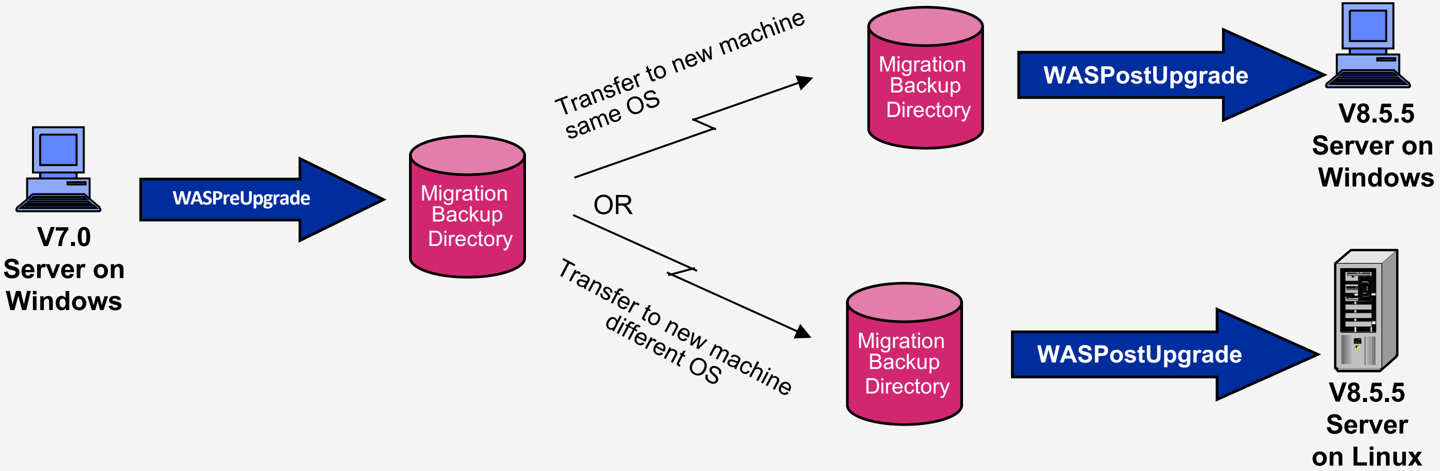
Cross-operating system and architecture is supported (excluding z/OS and IBM i.)

Decision to perform a local or remote migration is profile-by-profile.

Requires the new version of WASPreUpgrade to be present on the old host machine in order to collect the backup directory.

Can install new WAS on the old host temporarily, or

Can use “createRemoteMigrJar” command on the new host to produce a self-contained WASPreUpgrade bundle that can be used on the old host.


V7.0
Server on
Windows
WASPreUpgrade
Migration Backup Directory
Transfer to new machine same OS
OR
Transfer to new machine different OS
Migration Backup Directory
WASPostUpgrade
V8.5.5
Server on
Windows
Migration Backup Directory
WASPostUpgrade
V8.5.5
Server
on Linux

Replacing the cell or cloning it?

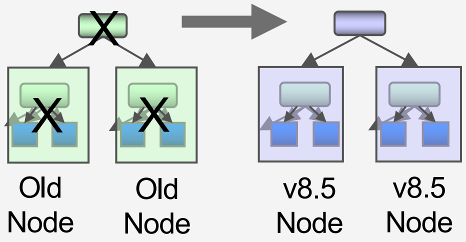
WebSphere v9.0 migration introduces the clone migration strategy.

Standard migration: each profile in the new release replaces its source from the old release within the cell as it is migrated. Cells can operate with nodes at different versions.

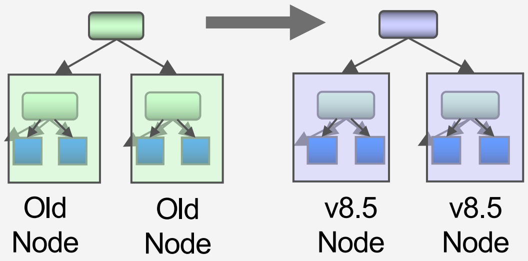
Clone migration: each profile is duplicated from the old release from the new release, resulting in two independent cells with the same configuration (except where necessary.)

All profiles in the cell must be cloned. Neither standard nor clone restrict choosing local or remote migrations on a per profile basis.

Standard migration



Clone migration



Planning and Best Practices

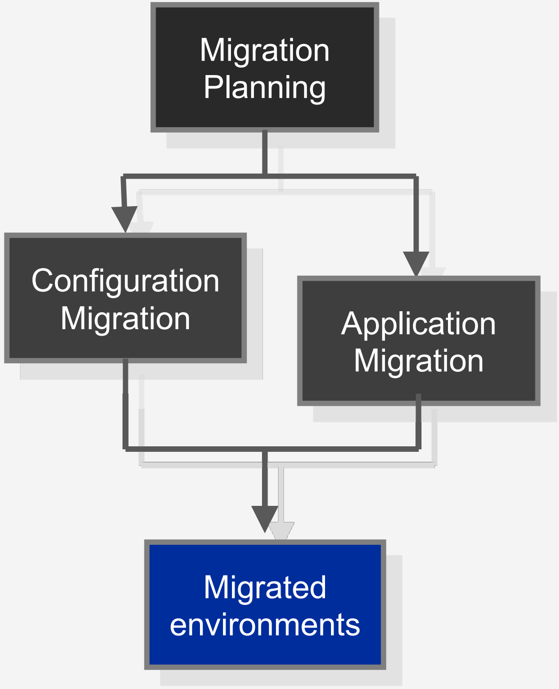
Planning your migration

Application and configuration migration can be planned and executed in parallel once the destination environment is known.

Use the migration tools to identify possible compatibility issues and implement fixes.

At the same time, you can begin developing your plan to migrate configuration in each of your environments.

Consider how you will verify your application works once the new environment is ready and the new code is deployed.



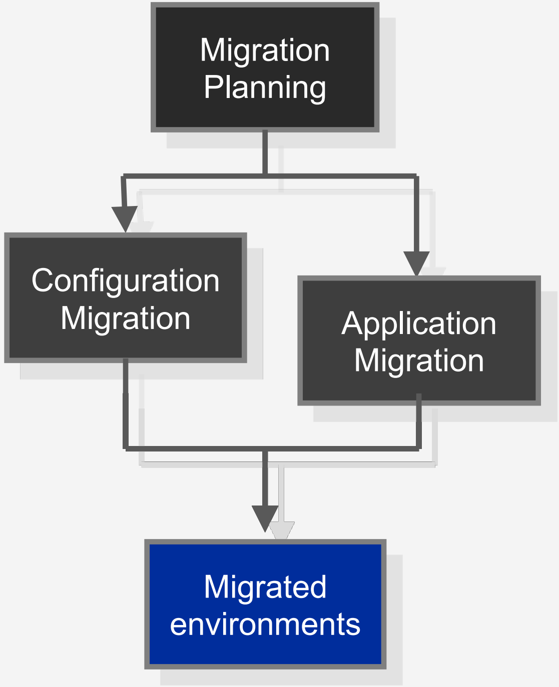
Configuration migration recommendations

If you have multiple stages of environments (development, test, production, etc.) begin with the least critical environment first.

Plan around any outage or change windows you may have to abide by. This may require doing only the critical tasks during the window and preparing everything else before or splitting the full migration over multiple windows.

Your plan should include the exact commands to run for each step of the migration procedure. Update this plan whenever you change any of the commands.

At a minimum, back up each profile in the cell before you begin migration. For best results, back up the deployment manager before migrating each node.



Problem avoidance and determination

Upgrade your new release to the latest fix pack before you begin migration.

- Fix packs include fixes for migration code.
- Upgrade before generating a remote migration jar, as well.

Migration can be very memory and file intensive depending on the topology.

Some general system problems are:

- Out of memory
- File handles
- Disk space

Other problems may include changes to the old WebSphere install and/or profile data.

- File permissions
- Network connectivity, TLS security, etc.
- Symlinks in the configuration
- Hand edited XML and properties files with invalid formats.
- Changes to the profile's setupCmdLine script.
- The list of profiles for the install is not correct.

Problem determination

The backup directory contains the logs for both WASPreUpgrade and WASPostUpgrade.

If a problem occurs during migration, or with servers or applications after migration, save the migration backup directory and engage IBM Support.

- Generally, support will need to see the backup directory and the destination profile after the migration attempt.
- Certain trace may also be needed; avoid setting “*=all” trace. That trace setting can cause performance issues.

– More helpful hints

- Become familiar with the tools and what they do. Run it, throw it away, start again.
- Migrate only one profile at a time when using the WASPreUpgrade and WASPostUpgrade commands.
- Migrate to a clean migration backup directory.
- Migrate into a clean target profile. Do not use the same profile already migrated into before.
- Separate the application installs from the WASPostUpgrade migration step by using the –includeApps script parameter.
- In a complex topology always perform backups of all cell profiles prior to migrating it - save as a recovery point.

Resources

Migration Knowledge Collection:

http://ibm.biz/WASMigPlanning

Especially “Migrating Traditional WebSphere Versions:”

http://ibm.biz/Mig2WAS9 (applies to both v8.5 and v9.0)

Knowledge Center for migration:

http://ibm.biz/MigratingWAS

Specific commands and procedures:

https://www.ibm.com/support/knowledgecenter/SSAW57_9.0.5/com.ibm.websphere.migration.nd.doc/ae/tmig_admin.html

Resources

Migration Toolkit (Source and Binary) downloads:

http://ibm.biz/DownloadWASMigTools

Transformation Advisor:

http://ibm.biz/CloudTA

WebSphere Application Server Versions: What’s Different?:

https://www.ibm.com/support/pages/sites/default/files/inline-files/$FILE/WebSphere%20Application%20Server%20Versions%20Whats%20Different.pdf

Thank you!

Special thanks to the panelists:

Cindy High
Michael Hill
John Juracek
Marika Joannnidis