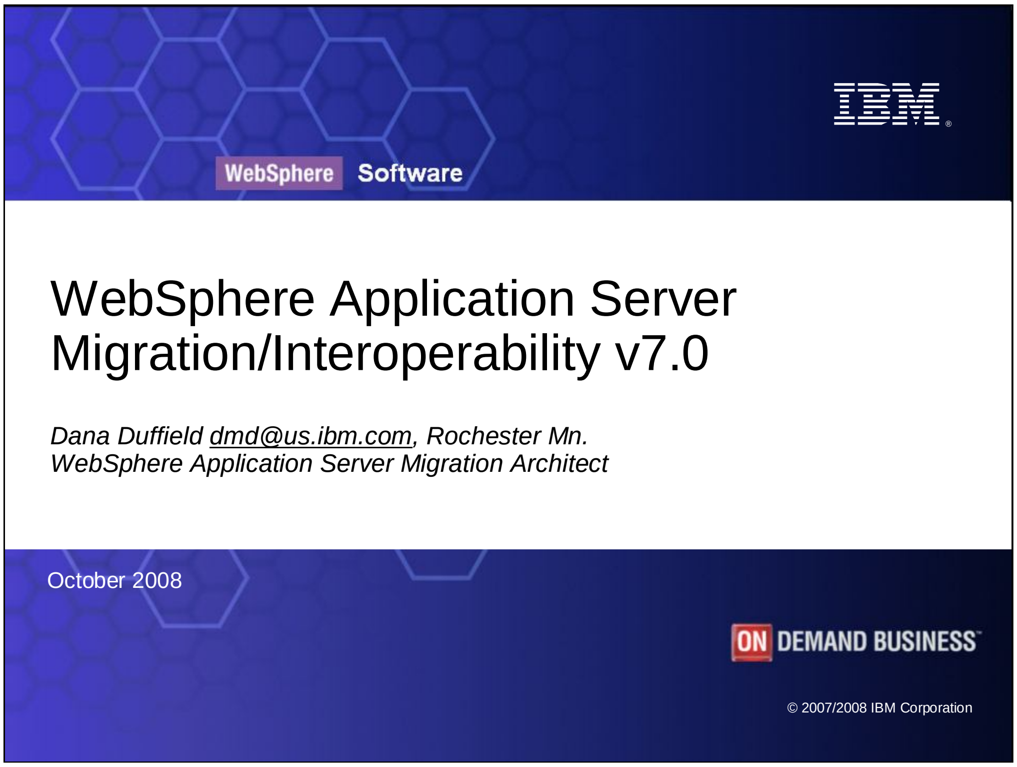


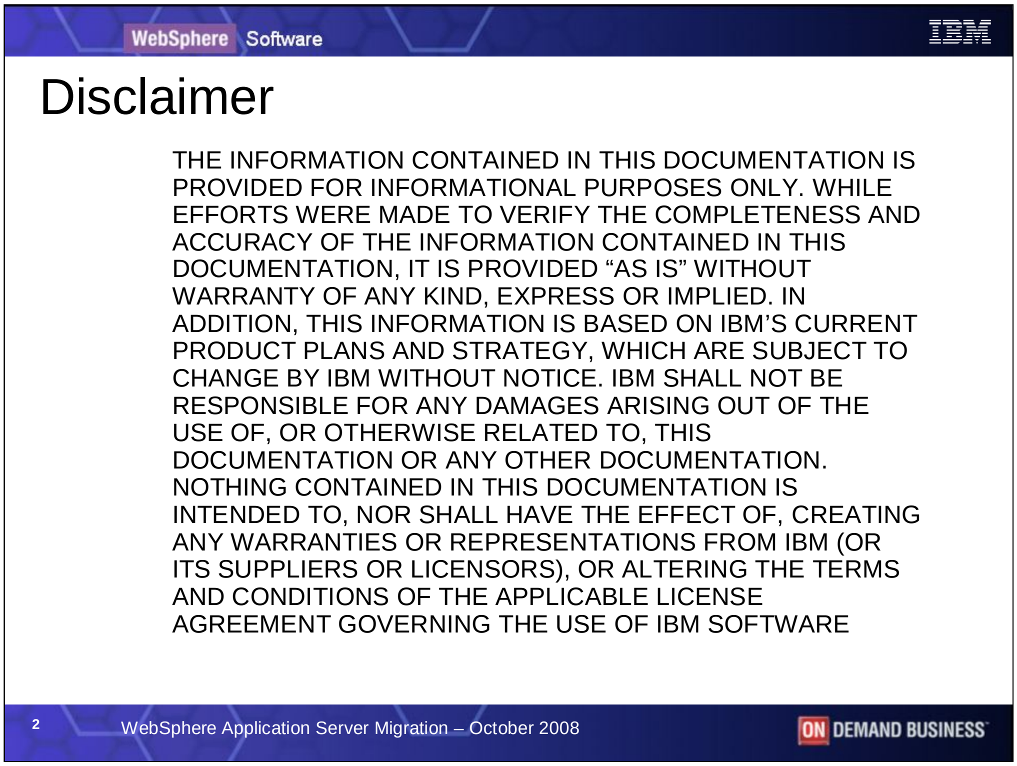
WebSphere Software

WebSphere Application Server Migration/Interoperability v7.0

Dana Duffield dmd@us.ibm.com, Rochester Mn.
WebSphere Application Server Migration Architect

October 2008

ON DEMAND BUSINESS



Disclaimer

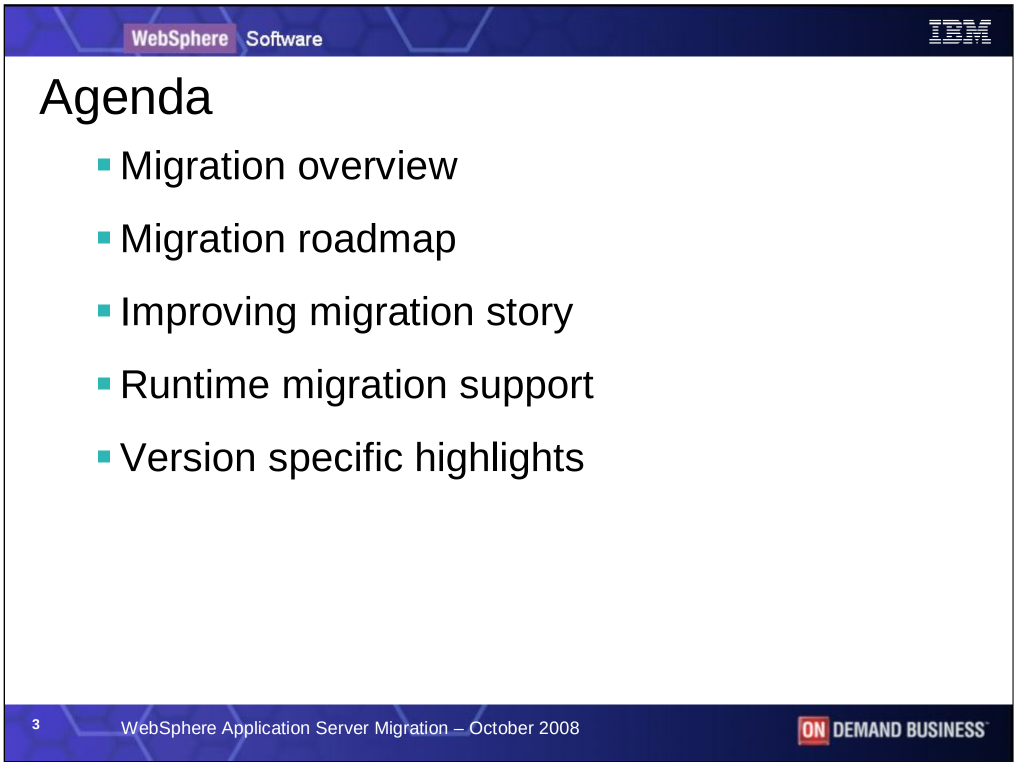
THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, IT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS DOCUMENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE

This Disclaimer does not need to be read by IBM WebSphere Application Server customers who have signed Non-disclosure agreements nor by IBM employees.

It must be read in all other cases with a summary of something like the following:

“Everything provided here is accurate to the best of my knowledge, however we are required to show this for legal reasons”

You can hide this by using the “Hide slide” function on either the “Outline” of the Normal view or the “Slide Sorter” view



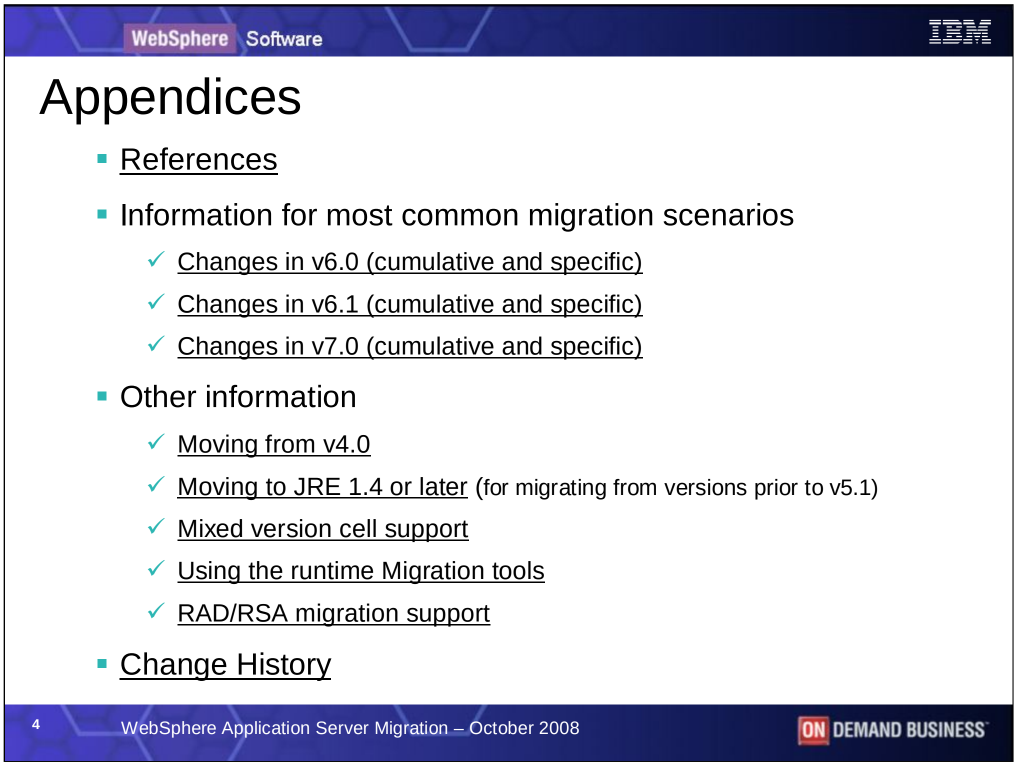
WebSphere Software

Agenda

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

This introduces the agenda

- The “Migration overview” section clarifies what we will be discussing
- The “Migration roadmap” describes details of what to consider before undertaking WebSphere Application Server Migration and provides how to build and execute a Migration plan
- The “Improving Migration Story” describes the improvements that have been made in WebSphere Application Server Migration over time. This is very important to understand when building a Migration plan because there are major differences in degree of difficulty based on the combination of
  - ➢Which WebSphere Application Server you are migrating **from**
  - ➢Which WebSphere Application Server you are migrating **to**
- The “Runtime Migration support” section describes the tools and techniques that can be used to migrate the WebSphere Application Server Migration between versions
- The “Migration Strategies” shows the typical strategies that can be chosen, based on each individual situation
- The “References” section contains a set of pointers to other materials that can be referenced for more information
- Change History is a log of what changes have occurred at different times in the deck. Since customers can get updates of the deck on a periodic basis we have found it is a good idea to have this section so those people can narrow down more easily on the latest changes.
- The remainder of the appendices include specific detailed information on various WAS versions of interest. This information is used during the planning cycle to determine specific impacts based on the **From** and **To** scenarios.
  - Each appendix has information on “Administrator, Developer and Misc topics”
  - For the currently most popular scenario of v5.1 to v6.1 see the Appendices on “Moving to v6.1” AND “Moving to v6.0 or v6.1”



Appendices

- References
- Information for most common migration scenarios
  - ✓ Changes in v6.0 (cumulative and specific)
  - ✓ Changes in v6.1 (cumulative and specific)
  - ✓ Changes in v7.0 (cumulative and specific)
- Other information
  - ✓ Moving from v4.0
  - ✓ Moving to JRE 1.4 or later (for migrating from versions prior to v5.1)
  - ✓ Mixed version cell support
  - ✓ Using the runtime Migration tools
  - ✓ RAD/RSA migration support
- Change History

This introduces the agenda

•The “Migration overview” section clarifies what we will be discussing

•The “Migration roadmap” describes details of what to consider before undertaking WebSphere Application Server Migration and provides how to build and execute a Migration plan

•The “Improving Migration Story” describes the improvements that have been made in WebSphere Application Server Migration over time. This is very important to understand when building a Migration plan because there are major differences in degree of difficulty based on the combination of

  ➢Which WebSphere Application Server you are migrating **from**

  ➢Which WebSphere Application Server you are migrating **to**

•The “Runtime Migration support” section describes the tools and techniques that can be used to migrate the WebSphere Application Server Migration between versions

•The “Migration Strategies” shows the typical strategies that can be chosen, based on each individual situation

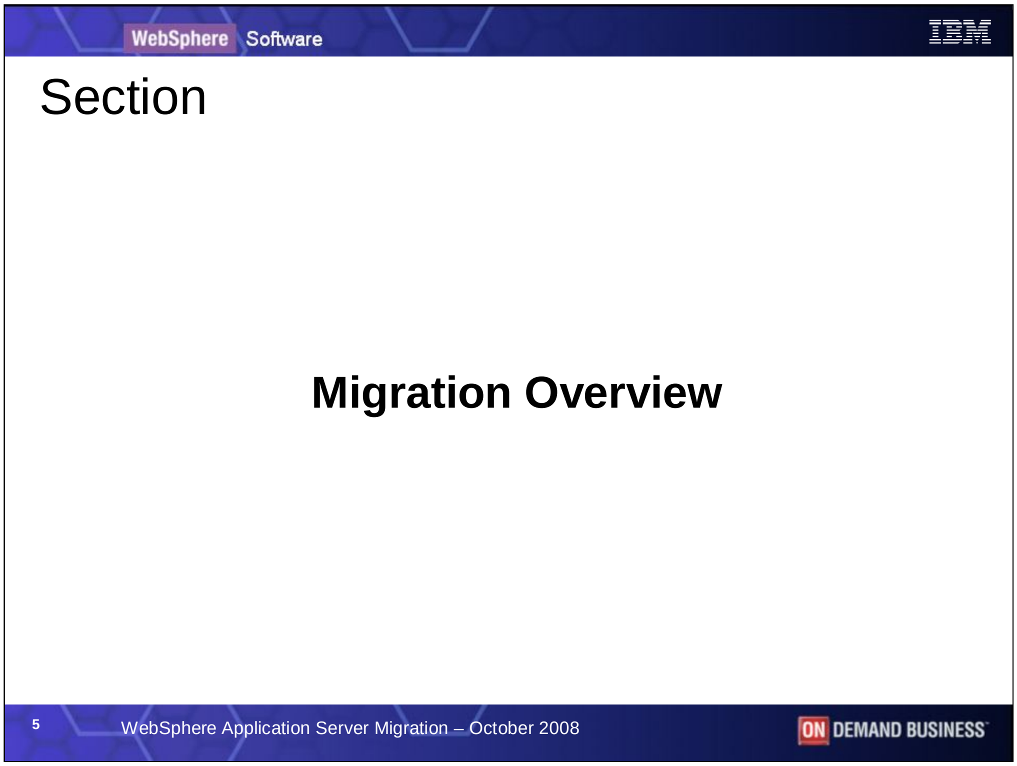
•The “References” section contains a set of pointers to other materials that can be referenced for more information

•Change History is a log of what changes have occurred at different times in the deck. Since customers can get updates of the deck on a periodic basis we have found it is a good idea to have this section so those people can narrow down more easily on the latest changes.

•The remainder of the appendices include specific detailed information on various WAS versions of interest. This information is used during the planning cycle to determine specific impacts based on the **From** and **To** scenarios.

  •Each appendix has information on “Administrator, Developer and Misc topics”

  •For the currently most popular scenario of v5.1 to v6.1 see the Appendices on “Moving to v6.1” AND “Moving to v6.0 or v6.1”

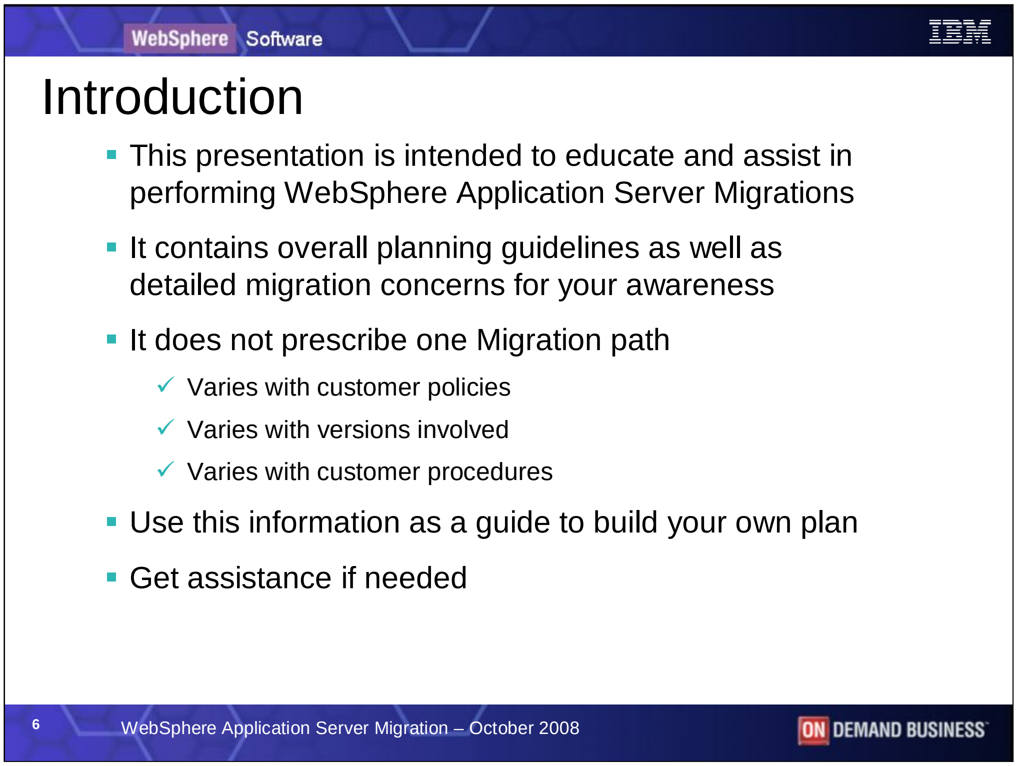


Section

Migration Overview

ON DEMAND BUSINESS

The “Migration overview” section clarifies what we will be discussing


WebSphere Software
IBM

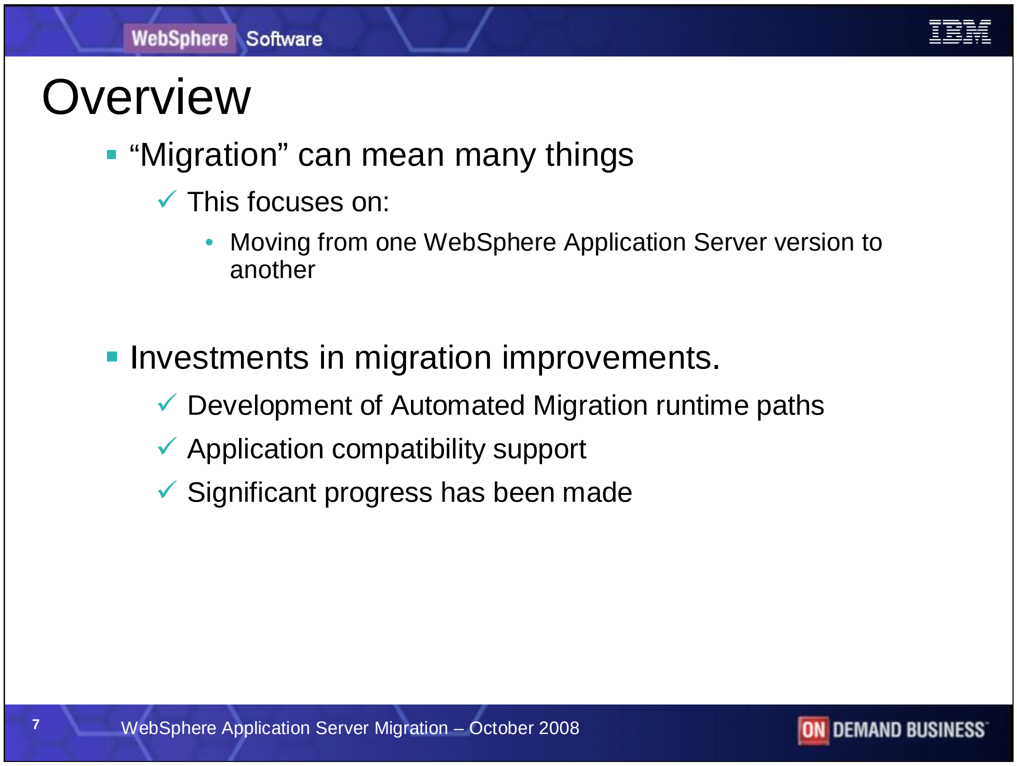
Introduction

- 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

The purpose of this material is to provide information required to plan and execute a WebSphere Application Server Migration. The goal is to enable customers to do this Migration themselves without requiring assistance. There has been significant improvement over time in the ability to more easily make this happen. The deck covers the overall planning process as well as details that can be used to determine specific details for the Migration scenarios. It does not prescribe one and only one Migration path. In our opinions and experiences there is not one and only one path. There are too many variables to prescribe one path.

The alternative is to provide the information and let each customer make informed decisions on what path is the best for their specific situation.

In not all cases is this possible, in that case it is wise to either augment the plan with Migration experts such as IBM Business Partners or IBM Services.



Overview

- “Migration” can mean many things
  - ✓ This focuses on:
    - Moving from one WebSphere Application Server version to another
- Investments in migration improvements.
  - ✓ Development of Automated Migration runtime paths
  - ✓ Application compatibility support
  - ✓ Significant progress has been made

As it says, we’re specifically talking about WebSphere Application Server Migration. We are not going to describe other types of Migrations like:

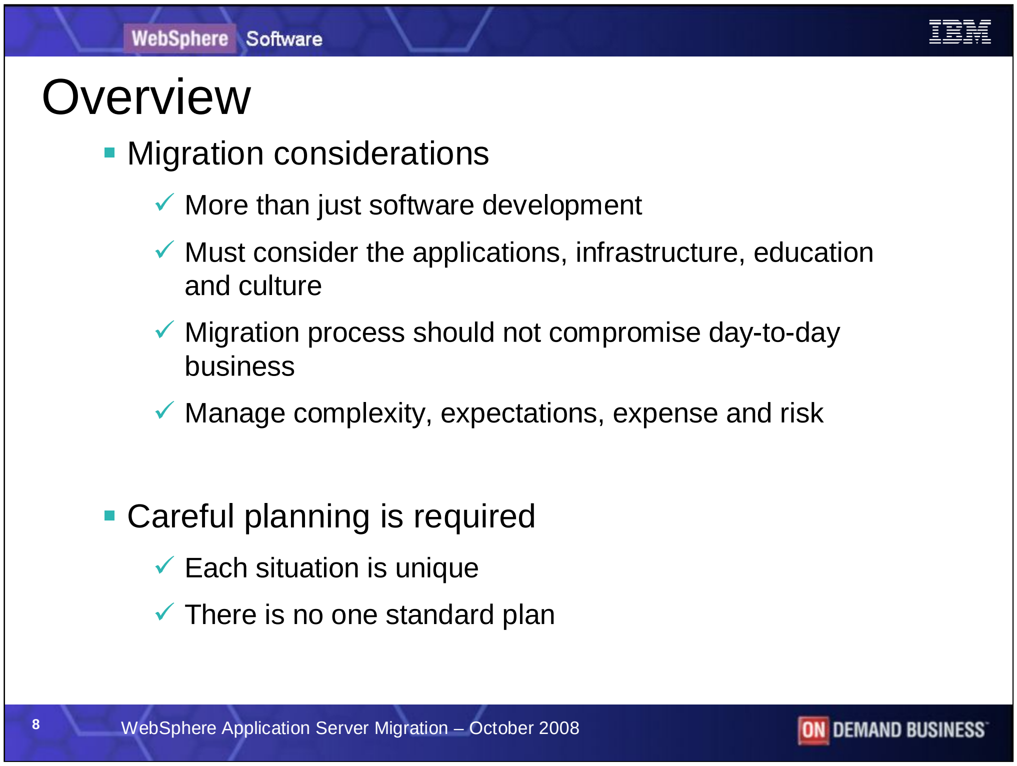
•Migrating between different vendor application server products

•Migrating other WebSphere products, we have some experience in this space but are not experts.

•Migrating an application to new architecture

•Applying fixpacks

It is worth saying that we have made significant investment in improving the migration experience. In that past this has been a difficult and costly task for many of our customers. Our goal is to reduce the cost and time required to perform WebSphere Application Server Migrations



WebSphere Software

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

8 WebSphere Application Server Migration – October 2008

It is easy to think of a migration in narrow terms, but it is dangerous to do so. Application developers often tend to think of migration only in terms of changes to the application code, and administrators think primarily of the production run time environment

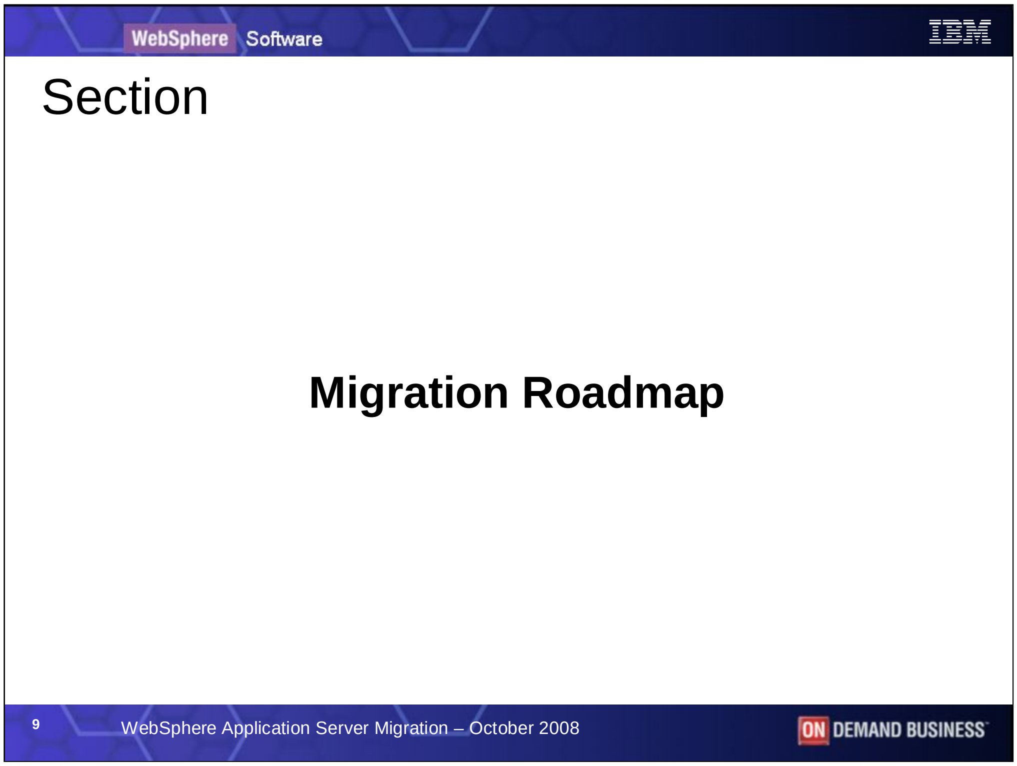
The point here is that there are many factors to consider when planning for a WebSphere Application Server Migration. Different factors and participants will be involved in the Migration process and plan. You should account for all of these.

Different perspectives

- •Product architect(Design and features)
- •Developer(code changes)
- •Administrator(production runtime)
- •Technical support(capacity planning)
- •Development managers(resources and deadlines)
- •Executive management(cost, risk, impact)

Careful planning is required. Many of the customer successes or failures can be directly related to the strength (or even existence) of a comprehensive plan. Many times Project Management becomes as important as the technical aspects when Migrating.

Each customer situation is unique based on the WebSphere Application Server version involved, use of 3rd party applications and previous Migration experiences: just to name a few.



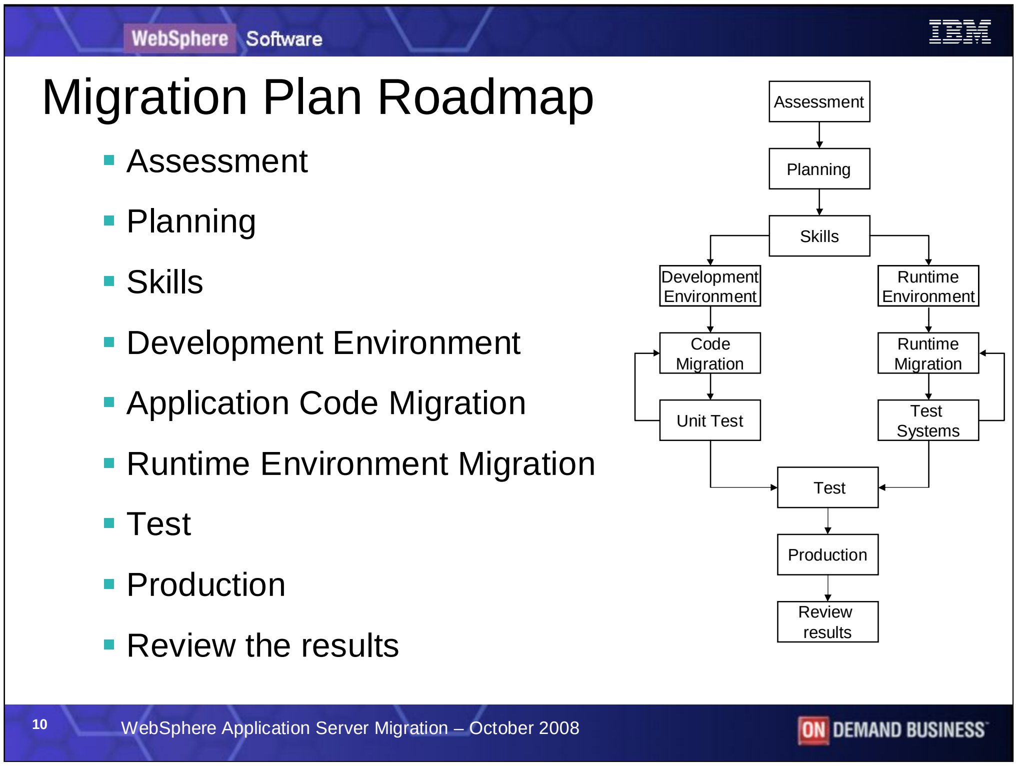
WebSphere Software

Section

Migration Roadmap

9 WebSphere Application Server Migration – October 2008

The “Migration roadmap” describes details of what to consider before undertaking WebSphere Application Server Migration and provides how to build and execute a Migration plan



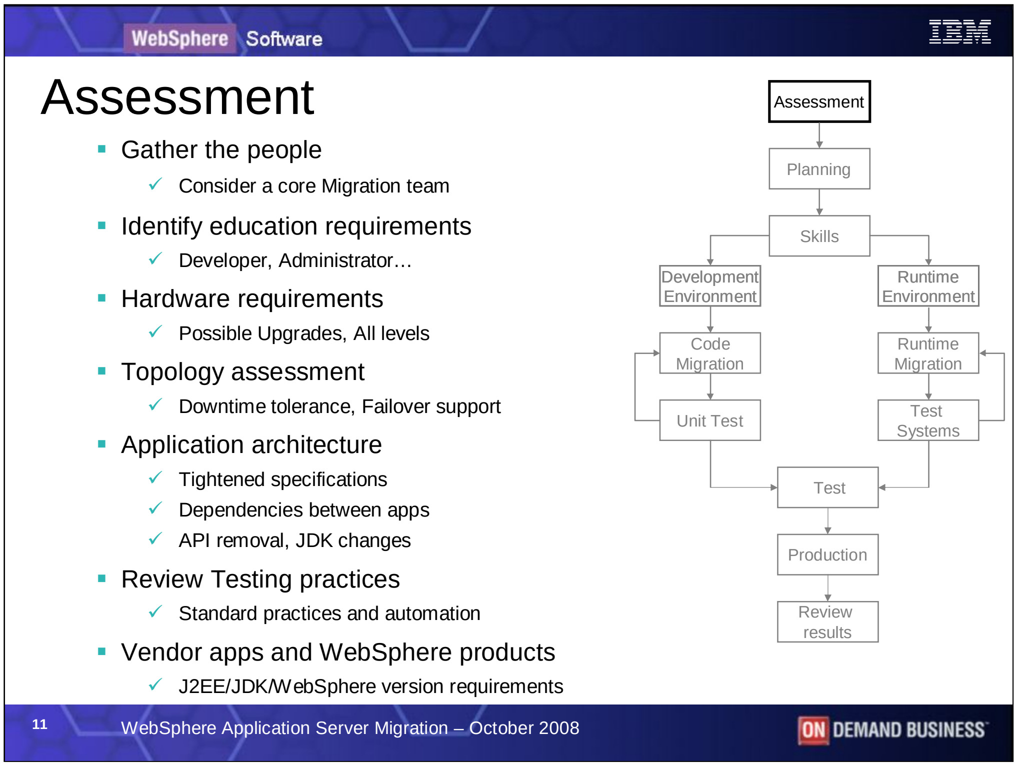
Migration Plan Roadmap

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

Assessment
Planning
Skills
Development Environment
Runtime Environment
Code Migration
Runtime Migration
Unit Test
Test Systems
Test
Production
Review results

This shows the overall process of building a Migration Plan. Each of the items will be covered in more detail later in this document.

1. The first step is Assessment. This step is very important as it identifies each of the factors that will have an impact on the Migration plan. It is effectively Requirements gathering based on each individual company's situation
2. Planning: takes the factors identified in the Assessment step and builds a specific set of actions and a proposed timeline for those actions. The plan may have to be readdressed as more factors are understood or one or more changes in timeline are required.
3. Skills: This step addresses upgrading any skill gaps that have been identified. Many alternatives for this education exist including formal classes, online courses, IBM Education Assistant, etc.
4. At this point there are two paths that can be worked in parallel. I typically talk about the "Runtime Environment" first because that is how it flows later in the materials. This path describes an iterative scenario to apply the runtime part of the Migration Plan to the various test environments that may exist in each company's environment. It provides a mechanism to test the runtime migration portion through the various test systems before performing the task on the Production system
5. Development Environment: Takes the standard iterative cycle of development to one more suggested degree when doing a WebSphere Application Server Migration
6. Test: Roll the applications or set of applications through the certification test cycle that is in place for the company.
7. Production: Roll the migration into the Production environment with hopefully no surprises
8. Review the results: Now is a good time to do a self-assessment of the success or failures of the different steps of the Migration. Now is an excellent time to feedback these changes into the Plan so the Plan can be re-used when it is needed next.



WebSphere Software IBM

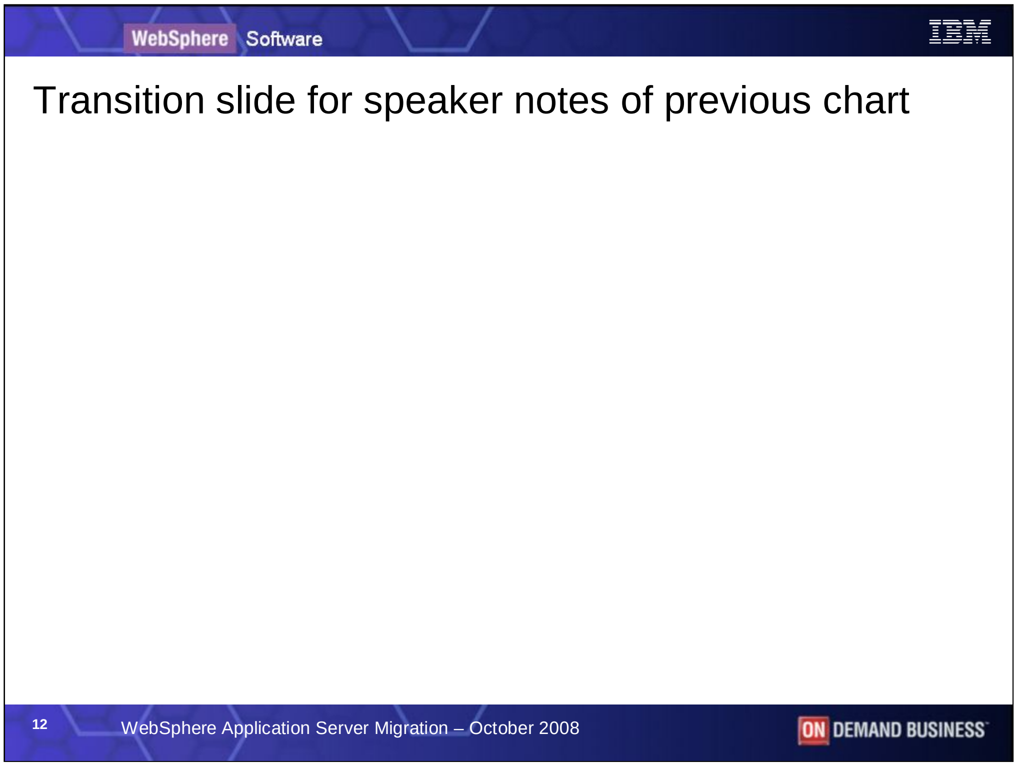
Assessment

- 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

11 WebSphere Application Server Migration – October 2008 ON DEMAND BUSINESS

This step identifies each of the factors that will have an impact on the Migration plan and the actions that will be required. It is effectively requirements gathering based on each individual company's situation.

- Gather the people – this is an important step for understanding all the requirements and to get everyone that is involved engaged. Typically you see new information surface because some team has a particular aspect to consider that no one else had yet identified. The aspect of a core Migration team has proven to be very effective in many cases. Some companies expand current duties of an Advanced technology team to be the first to explore the issues and own the plan. Others assign someone to be in charge of the overall process as part of a new job definition. In general the best results are achieved when someone is in charge of the technical details and rollout as opposed to providing information to the team without an overall plan.
- Identify education requirements – typically the education requirements revolve around two primary roles (Developer and Administrator). Although other roles or related roles can also be impacted.
  - Developer – What changes can impact the Application Developer? One area that has had a large impact in the past is the education required to learn details of J2EE development. This is a major factor when migrating to WebSphere Application Server 4.0 and later. This is when the true Java 2 Enterprise Edition (J2EE) programming model was supported by WebSphere Application Server. Another major factor for education is when changing Integrated Development Environments (IDEs). This happened when moving to WebSphere Application Developer (WSAD) in WebSphere Application Server 4.0 and later as well as the Rational tool suite in WebSphere Application Server 6.0 (primarily Rational Application Developer (RAD)) and later. New features and function may also be available to be understood in each WebSphere Application Server version New features and function may also be available to be understood in each WebSphere Application Server version.
  - Administrator – The administrator may also have educational needs. The major point of concern is crossing the boundary from the WebSphere Application Server 3.x and 4.0.x database repository environment to the file/XML based system starting in WebSphere Application Server v5.0 and carrying forward into later versions. New features and function may also be available to be understood in each WebSphere Application Server version. New features and function may also be available to be understood in each WebSphere Application Server version. Once the administrators are on WebSphere Application Server v5.0 and later there are fewer concerns although focus on what has changed and new features in each version is important to understand.
- Hardware requirements Version upgrade is a good time to understand if it is a reasonable time to upgrade hardware. It may coincide with planned sun setting of hardware and should be factored into the planning. A choice of hardware upgrade during WebSphere Application Server version migration may impact the Migration Strategy that can be selected.
  - Capacity planning – Upgrades to processor, disk capacity or memory may have to be factored into the plan. The latest WebSphere Application Server versions typically does not force a hardware upgrade but in many cases an increases memory consumption may be assumed.
  - One hardware assessment that can be forgotten is the hardware beyond the Production environment. The other Test systems in the development cycle, including Developer hardware should be considered.



WebSphere Software

IBM

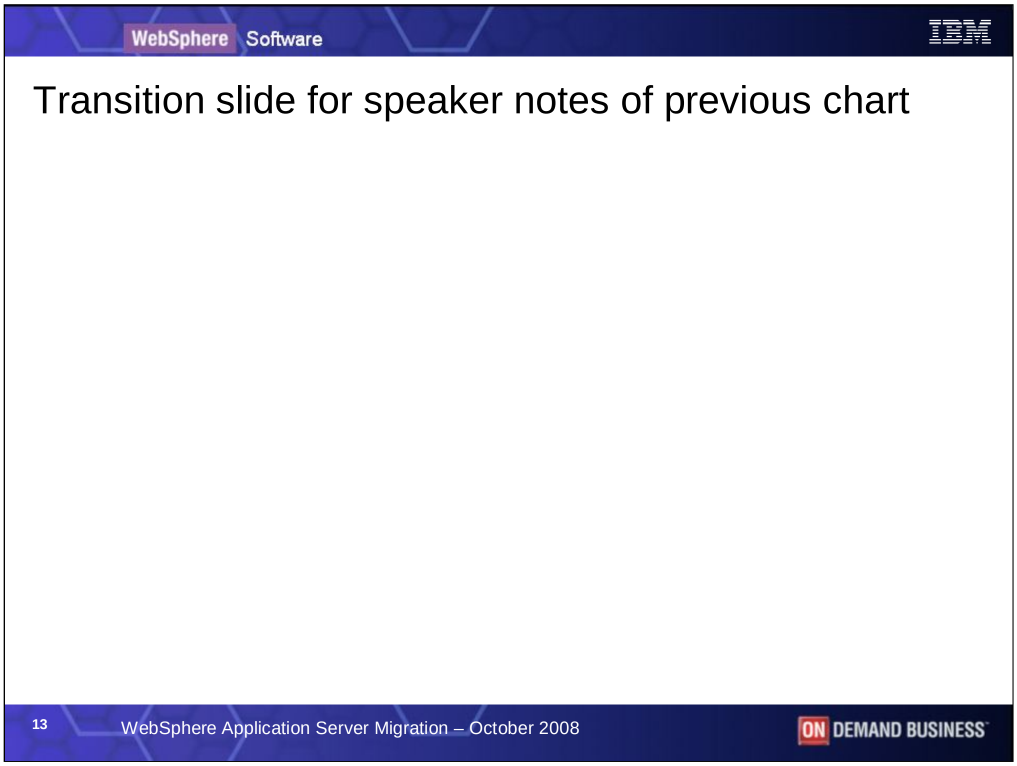
Transition slide for speaker notes of previous chart

12

WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

- Topology assessment – The topology can factor into the plan for how to Migrate the production systems.
  - Downtime tolerance – Different Qualities Of Service (QOS) requirements for different applications and systems have to be understood. This may impact the ability to upgrade during a specified maintenance window and will have to be factored into the plan. This could drive an additional step of prototyping a Migration production to determine if the maintenance window can be satisfied. Additional capabilities are being provided as part of the WebSphere Application Server Migration package (Mixed Node support) that can be used for minimizing this risk.
  - Failover support – Failover support requirements during a Migration are important to understand. This will determine another QOS aspect if there is not a failover topology already in place to address the availability needs of applications.
- Application architecture – Various aspects of Application architectures also need to be understood. This is much more of a variant based on the WebSphere Application Server release and associated J2EE level than some of the other Assessment factors already described.
  - Tightened specifications – When migrating to later WebSphere Application Server versions there may be cases where some applications that used to load and run on previous WebSphere Application Server versions will have some issue. One such area is tightening of J2EE specifications to clarify some areas that were not totally closed in prior specifications. An example is in the area of deployment descriptors. In both WebSphere Application Server v5.1 and v6.0 there are cases where applications will no longer be installable in those new versions that could be installed in earlier versions. This is due to tightened checking of the deployment descriptors during application installation. In other cases there may need to be changes made because of behavioral differences in Application Programming Interfaces (APIs) between the specification levels.
  - Dependencies between apps – If there are dependencies between applications or on shared libraries these have to be understood. These factors can determine the order of migration of these applications.
  - API removal – If APIs have been removed that are in use by existing applications then these have to be modified before migration to the new WebSphere Application Server version. These API removals are tightly controlled and documented in the WebSphere Application Server InfoCenter. This documentation also includes deprecation information.
  - JDK changes – It is important to know when Java Development Kit (JDK) boundaries are crossed when planning to Migration. In most cases the JDK compatibility is very strong. In some cases there are documented breakages that need to be understood. More information on this is provided on the JDK compatibility site as noted in the References section.
- Review Testing practices
  - The existing testing practices should be reviewed to determine whether or not they provide adequate coverage. If they do not, then some augmentation of testing practices may be required. When a problem is encountered during migration and well-defined, robust testing practices are not in place, it is often difficult to know if the migration is a catalyst that exposed a pre-existing problem, or if the migration is the actual source of the problem. This makes it difficult to gauge the progress and the ultimate success of the migration. A comprehensive testing strategy is a very important part of enterprise application ownership. The amount of automated testing is also an important factor. Automation ensures repeatable success.



WebSphere Software

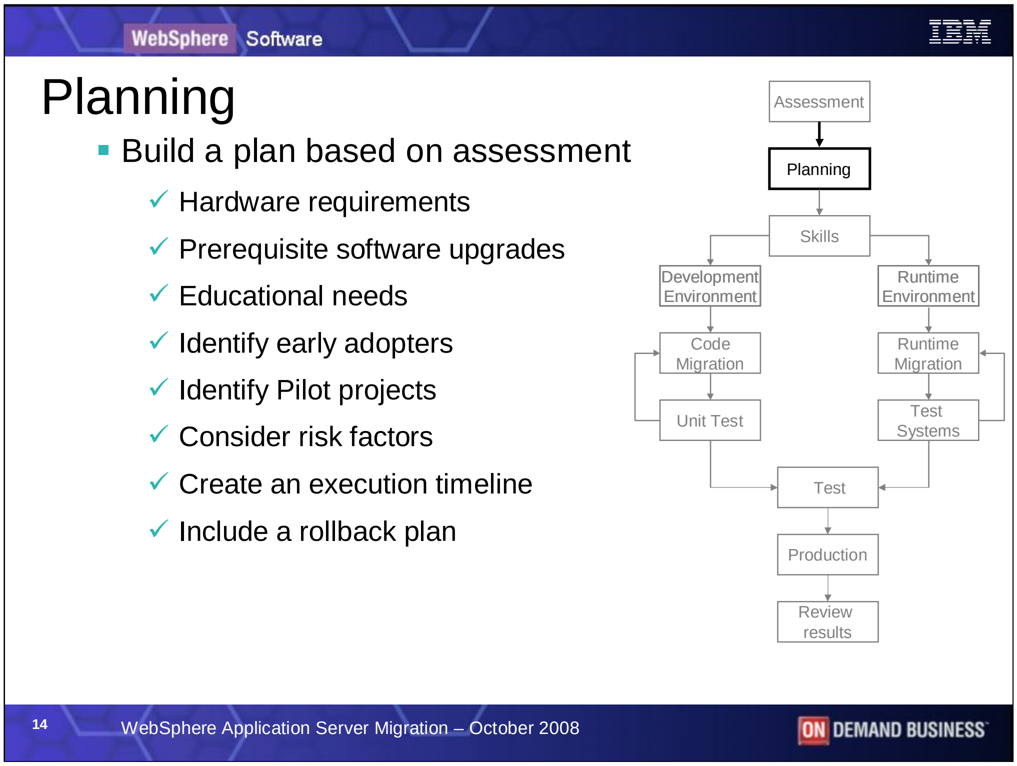
IBM

Transition slide for speaker notes of previous chart

13 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

- Vendor apps and WebSphere products – Supported specification levels can be a factor when determining which WAS version to migrate toward.
  - J2EE – Sometimes specific J2EE levels are required by companies/customers. Knowing which J2EE level is supported helps determine which WAS version to target.
  - JDK – Same for JDK level. One of the primary motivators for WebSphere Application Server v5.1 was JDK 1.4. Knowing what JDK level that is supported is more required information. It is also useful to know when a JDK boundary is crossed to help evaluate the Migration impact. More on both of these topics is provided later in this deck.
  - WebSphere level requirements - Third party and other WebSphere products (e.g. WCS, WPS, …) can have specific requirements for WebSphere Application Server versions that they support. This can cause complications on which WebSphere Application Server version to migrate to when and whether work is required to potentially line up support from vendors for the version that is actually desired by the customer.
  - WebSphere version changes
    - Architecture – Certain WebSphere Application Server versions had changes in architecture that impacted Developers and/or Administrators. Understanding these impacts is key to building the plan. More is covered on this later in this presentation.
    - Compatibility – Compatibility of applications and administration scripts is another factor to consider when building the plan. Various WebSphere Application Server versions have different levels of compatibility support. More is covered on this later in this presentation.
- Vendor application management – Many times this area requires a significant amount of work. In many cases there is not even a well known list of what vendor applications are being used in a company. This is the first order of business. The reason this is important is each of these applications have to be ready to run on the chosen version of WebSphere that you are migrating to next. Each vendor will have to be contacted for their statement of support of this version and some negotiation may be required to get this support. Failure to do this can lead to delays in the migration planning or require keeping older versions of WebSphere Application Server in service to run these applications. It is also worth considering laying out requirements for support of future WebSphere Application Server versions based on a timeline after general availability of WebSphere Application Server versions. This does not help the current situation that you may have to deal with but it does provide leverage for the future.


WebSphere Software

IBM

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

Assessment
Planning
Skills
Development Environment
Runtime Environment
Code Migration
Runtime Migration
Unit Test
Test Systems
Test
Production
Review results

WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

Given the Assessment and basic requirements, the next step is to build a plan taking each of the assessment results into account

•Hardware requirements – One of the first actions is to acquire and plan for upgrading hardware requirements. This can include Development machines (which should be considered an early upgrade) all the way through the Test environments (which would typically require a plan to upgrade in parallel. Remember that the Test and Production systems are not the only hardware platforms that need to be considered. Sometimes Development machines may also need to be upgraded if needed. For example, you require more memory for RAD, see reference in the back titled “Rational Application Developer Performance Tips” for some relief.

•Prerequisite software upgrades – In most cases the newer version of WebSphere Application Server will require upgrading some prerequisite software before you can upgrade. Primary examples of this are the Operating System and WebServer levels. See the references for a link of finding this information for the particular version of WebSphere Application Server that you will be using.

•Educational needs – Education needs should also be addressed fairly early in the cycle. There are a variety of techniques that can be used. Standard face-to-face classes are usually available. There are education modules available for more focused learning. See the references section for these “WebSphere Training and Technical Enablement” and “IBM Education Assistant”

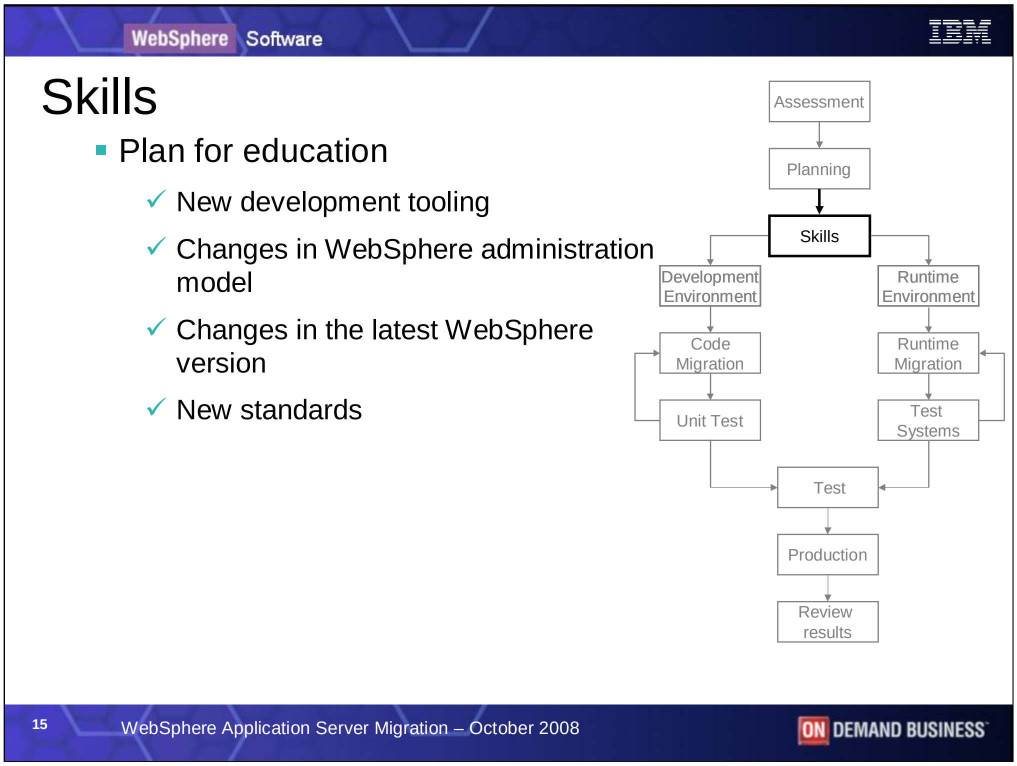
•Identify early adopters – If possible it is best to use iteration in working with staging the Migration with Development teams. Sometimes groups emerge that either want or need to be on the latest technologies. It is ideal to work with these teams as a first pass through Migrating applications and then apply lessons learned to further iterations with other teams. Communications is an important part of this process as lessons learned by the early teams can pay big benefits for the rest of the process.

•Identify Pilot projects – Work with the earliest adopters to identify Pilot projects that can be used to drive the Migration process. There are different benefits of selecting mission-critical versus “typical’ applications in building confidence in either the simplicity or the common changes that may be required for a successful Migration. The “typical” applications that are migrated can be used as a measuring stick for determining effort of similar applications later in the cycle. The overall plan can and should be adjusted as more information is earned.

•Consider risk factors – There can be a large variety of risk factors. The idea is to identify and minimize them. Several examples include impeding application updates too long while Migration is occurring and exceeding a window where it is critical to have stable and available Production environment.

•Create an execution timeline – From the previous information take a first pass at building an execution timeline. Include all the factors above and upgrade plan for each of the Test systems in the Test environment. This should be viewed as an iterative plan. Once some of the earlier steps are completed the timetable should be readdressed and refined. Most organizations have applications that are constantly changing in response to external business drivers. Working a migration plan around other deliverables can be a daunting task that requires special attention.

•Include a rollback plan – Always include a rollback plan. This is critical to the business. It is also important to practice the



WebSphere Software

Skills

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

Assessment
Planning
Skills
Development Environment
Runtime Environment
Code Migration
Runtime Migration
Unit Test
Test Systems
Test
Production
Review results

15 WebSphere Application Server Migration – October 2008

Skills building can take different forms and should be adjusted based on how people learn best.

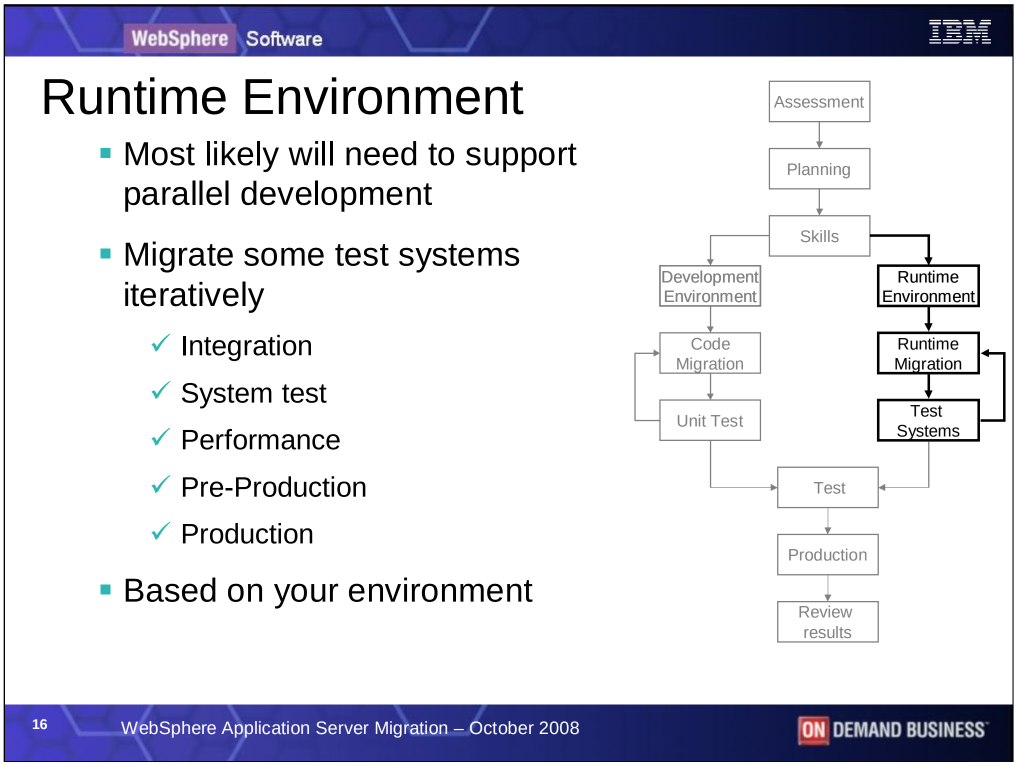
A variety of techniques can be used. Standard approaches include formal classes on many different levels. Other alternatives, such as online education can also be used to address this need. One online source worth considering is the IBM Education Assistant. See References for more information.

•New development tooling – Some of the WAS version boundaries include changes in IDE as well. The movement from VAJ to WSAD was a fairly significant change. The movement from WSAD to RAD has proven to be much less drastic. Both are based on Eclipse 3.0.x and appear to be an evolution.

•Changes in WebSphere administration model - – This is a significant step when moving from WebSphere Application Server 4.0 or earlier to WebSphere Application Server 5.0 and later. This is due to the change in how the administration model is architected. This also impacts the administration scripting model and education on making that transition should be included. As previously noted the administration model has been very stable since version 5.0. However, some changes in administration scripts may be required and it is always advisable to learn new features and capabilities.

•Changes in the latest WebSphere version – new features in the latest WebSphere Application Server versions provide an opportunity to take advantage of new capability. This can be an opportunity to Developers as well as Administrators

•New standards – New standards, such as J2EE, may be available in the latest WebSphere Application Server version. This provides additional features and support that needs to be understood.


WebSphere Software
IBM
Assessment
Planning
Skills
Development Environment
Runtime Environment
Code Migration
Runtime Migration
Unit Test
Test Systems
Test
Production
Review results
ON DEMAND BUSINESS

Runtime Environment

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

16 WebSphere Application Server Migration – October 2008

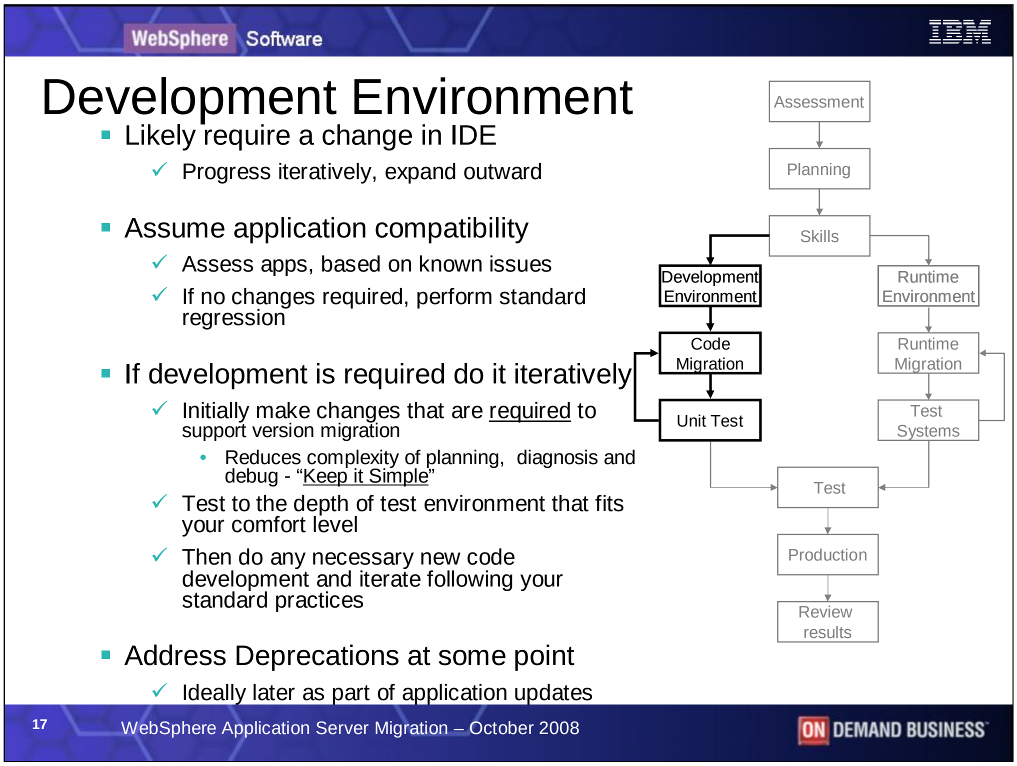
Testing should already be part of your application development and delivery lifecycle. Migration is just a part of that lifecycle process, and therefore should involve rigorous testing at each stage of the project.

Most organizations have more than one run time environment, including development test, system test, performance, and pre-production. Not all companies approach the completeness of testing support as shown in this example. Other environments with other purposes may also exist. Rarely is the migration of a production run time environment a simple matter of shutting everything down, installing new hardware and starting everything back up. This may be possible for development test and system test environments, but other run time environments have restrictions that make such practices impractical, if not impossible.

For the migration period you will need parallel test environments in most if not all of the test environments so that you can maintain applications that have not been migrated but also are able to support testing for the new version. It is also important to use some of these test systems as a test of your migration plan, especially those systems that are the closest to representing the production environment.

The other aspect to understand is that the risk of losing access to these test systems typically increases as you approach production systems. For example, losing a development environment is not typically as catastrophic as losing pre-production or of course production.

In nearly every case, the production run time environment must stay in service while it is migrated. Often, very limited downtime is acceptable for even the pre-production environment. Migration of a production run time is made more complex if there are multiple applications with different development teams and different delivery schedules. In this kind of situation, you may need to run both application servers concurrently for a period of time.



WebSphere Software

IBM

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 <u>required</u> to support version migration
    - Reduces complexity of planning, diagnosis and debug - "<u>Keep it Simple</u>"
  - ✓ 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

17 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

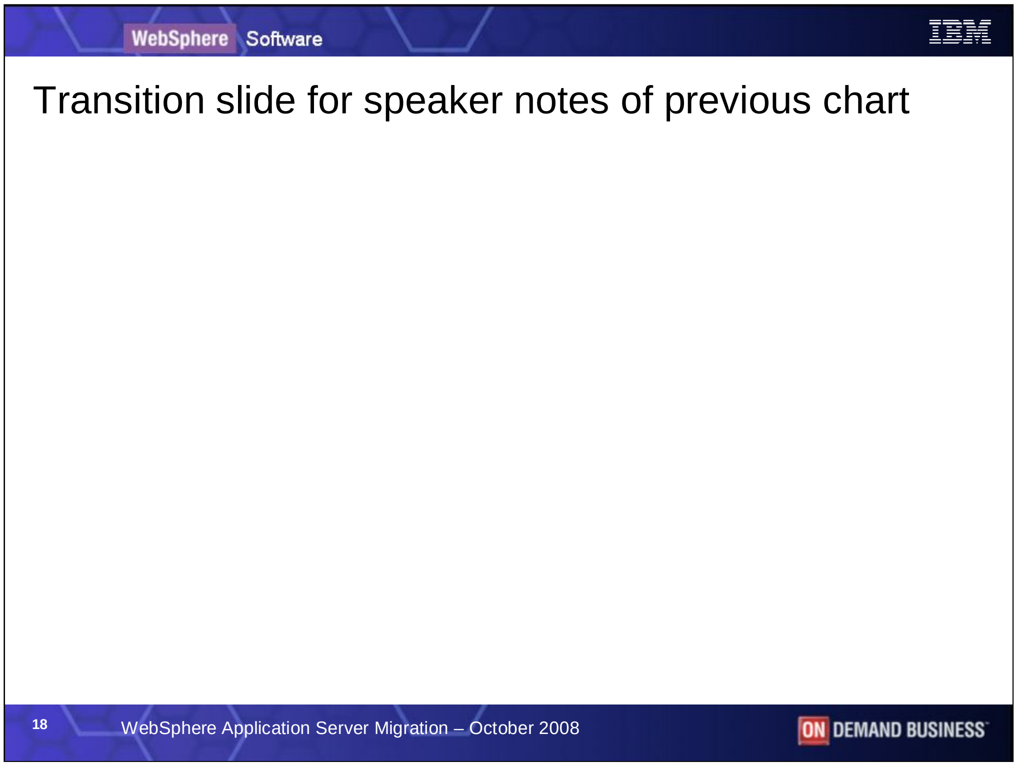
•Moving to the next version of WebSphere Application Server will more than likely require a change in IDE. This may require some significant work if you are migrating from the older versions of WebSphere Application Server. This can also be somewhat of an emotional issue if changes are significant between the toolsets. As time has gone by significant progress has been made in making these transitions easier. In the latest scenarios the migration support is very strong, including tools to migrate workspaces across different versions of the tools as well as tools to upgrade source code to newest J2EE specification levels.

•Iteration can be useful in this transition as well. Start with a smaller number of developers with their associated projects to migrate and gain experience in using and transitioning to the new toolset. As lessons are learned they can be shared. It is also important to understand if the new tooling will require new hardware or memory upgrades. Be sure to check these requirements as part of the overall planning process.

•Application compatibility – In older versions of WebSphere Application Server the compatibility support was not ideal. This is especially true when a transition was made from versions older than v4.0 were made to v4.0 or later. This required a significant application re-engineering. WebSphere Application Server versions since that time have greatly improved compatibility. The view should be to start with an optimistic view that applications should be highly compatible between versions. Of course there are known cases of incompatibility. Use the details of the end of this material based on the migration scenario that you are performing to understand potential incompatibilities and plan for these changes as part of the migration. If it is determined that no changes are required for specific applications then plan on standard test processes and procedures for those applications.

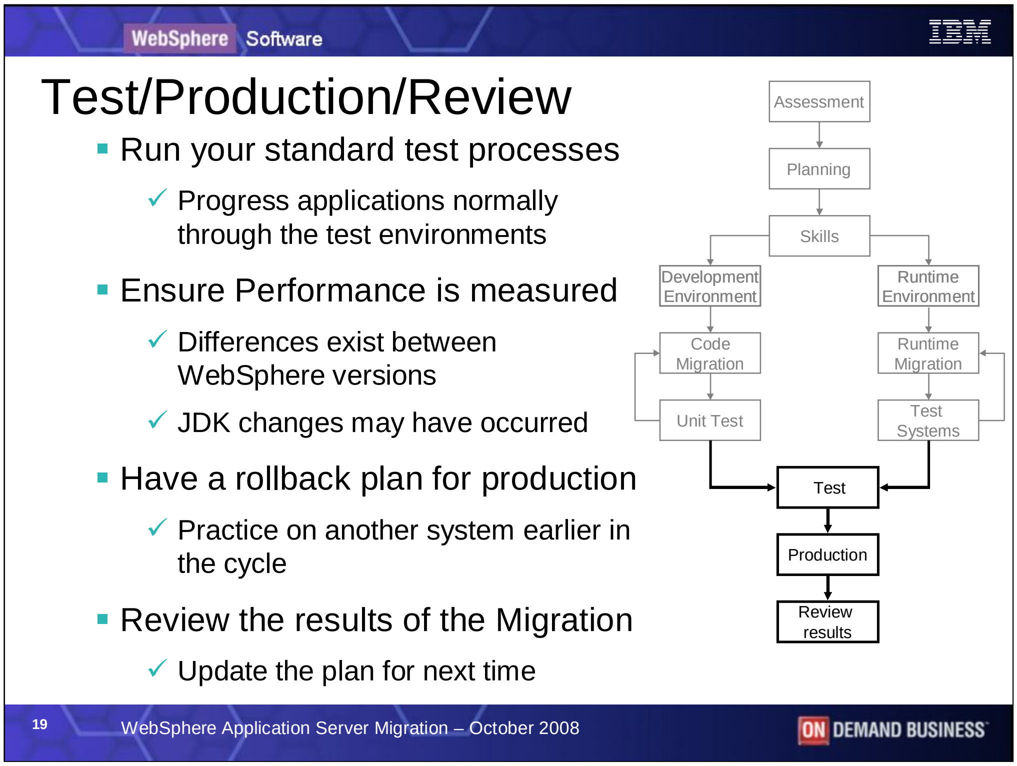
•It is also important ro understand that there is no need to upgrade applications to the latest J2EE specification level supported by the version of WebSphere Application Server that you will be using. Support is also provided to run applications written to these earlier level J2EE specifications.

•Hopefully no development is required to change existing applications, however this is not always the case. If changes are required it is important to try and keep the changes minimal and only to those that are <u>required</u> to just get the application migrated and running on the new version. This is just simple logic in order to reduce complexity. The more changes that are introduced, the more problems can be introduced. If there is some problem to resolve it is easier to narrow down on the problem when the fewest number of changes are introduced. We have seen many cases where changes were introduced into an application at the same time as version migration. If a problem occurs there is a tendency to blame the problem on the new WebSphere Application Server migration. Sometimes this is true and sometimes it is not. The point is it is much more difficult to debug the problem and do the problem analysis and is exponentially difficult based on the number of changes that have been made. It is not always possible to manage to this ideal. There are cases where application upgrades must be made at the same time as version upgrade in order to satisfy business needs. In that case it is minimally a good idea to logically separate the source code changes into migration-related and business related and checkpoint the source code. In this way you can compromise the Keep-it-Simple approach championed here with the business needs and have some way of logically separating changes into testable units.



Transition slide for speaker notes of previous chart

- The degree of testing effort required can be controversial. Sometimes it is quite expensive to perform testing and doing extensive tests versus sensible amount of testing can be a difficult decision. There is always a tradeoff of risk versus expense. This will be addressed more fully in the next chart. In the end it is an decision each company may make differently based on the risk/reward criteria.
- After successful and stable migration to the WebSphere Application Server then it is safe to use the new features and functions of the latest version. Remember that once these new features are used they cannot be supported on older versions of WebSphere Application Server. This can be more complex if a cell is left in mixed-node environment for a longer period of time. This mixed-node topic will also be addressed more fully on a later chart.
- Similar to new code development, address Deprecations at some later point in time. WebSphere Application Server deprecated interfaces should be planned to be removed at a later cycle of development, typically as part of application maintenance or as on aspect of standard application upgrades.



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

•Once the development and runtime testing environments are ready to go there is a significant amount of work to be done before progressing to the point of rolling out migration to the production environment. The applications have to be tested and perhaps modified, the test environments have to be replicated and the runtime migration process for the production system must be tested on one or more test systems.

•Part of this migration process will involve testing the applications themselves. This should ideally be performed using a set of standard regression tests that are typically used whenever an application has been modified for maintenance or for business upgrade. These same tests would be used for standard testing as well as version upgrade scenarios. Ideally these tests are automated and repeatable to reduce expense. The other question is How much testing is enough? Ideally all applications would be fully tested but this can be very expensive. In all cases allowances need to be made to account for practicalities, in the end every company has to make their own call based on quality goals, risk and expense. Some of the variables are

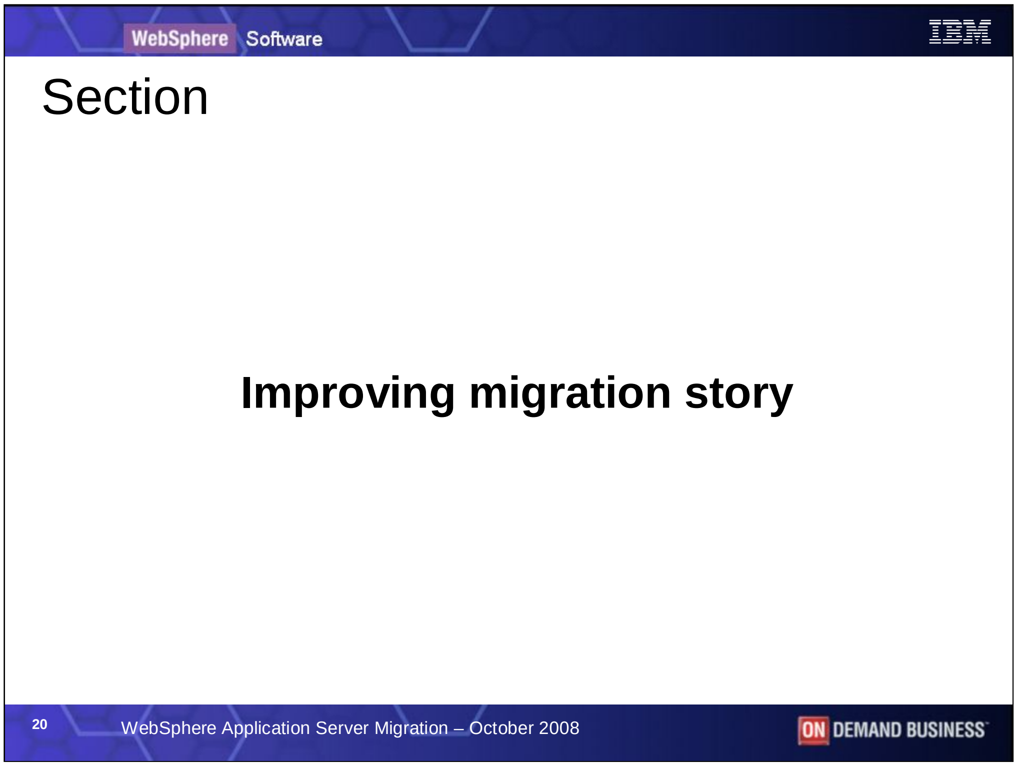
> •**Internal applications**, if slightly lower quality is allowed then reasonably a lesser amount of testing can be done. If this approach is taken hopefully some gains can be gained if more risky applications have some common characteristics with these internal applications, **Intranet** – typically the next level of quality required and higher risk **Internet** – typically the highest risk and highest required quality requirements.
>
> As you can see there can be varying degrees of risk with various applications and different testing philosophies can be used for these (or other) categories of applications. You can also choose the use the iterative approach here as well. Applications can be grouped by features that they use. Once these categories are established you can pick the highest risk application(s) and test those first and most thoroughly. Depending on how this testing goes you can determine how much effort to apply to the rest of the applications in this group. For example, if no or few problems were found with the initial tests you can consider a calculated risk of level of testing effort with the remaining applications in this group.

•It is also a good idea to establish a Performance baseline of your most critical applications before migrating. This can be used to determine if there is measurable Performance degradation if a problem occurs on the new WebSphere Application Server version. Many times the newer version will require more memory for the default application server configuration. If you have tightly tuned your existing environment then you may experience some performance degradation and the tuning will need to be revisited.

•Always have a rollback plan for the production environment. This is critical to the success of your business in case a drastic problem does occur. Also be sure to practice the plan before you may need to use it!

•Once this is accomplished you are ready for the final step of migration of the production environment.

•It is always a good idea to do a post-analysis of the migration process from beginning to end to see what went right and what could have gone better. You can then iteratively improve your migration experience with this knowledge and adjustments.

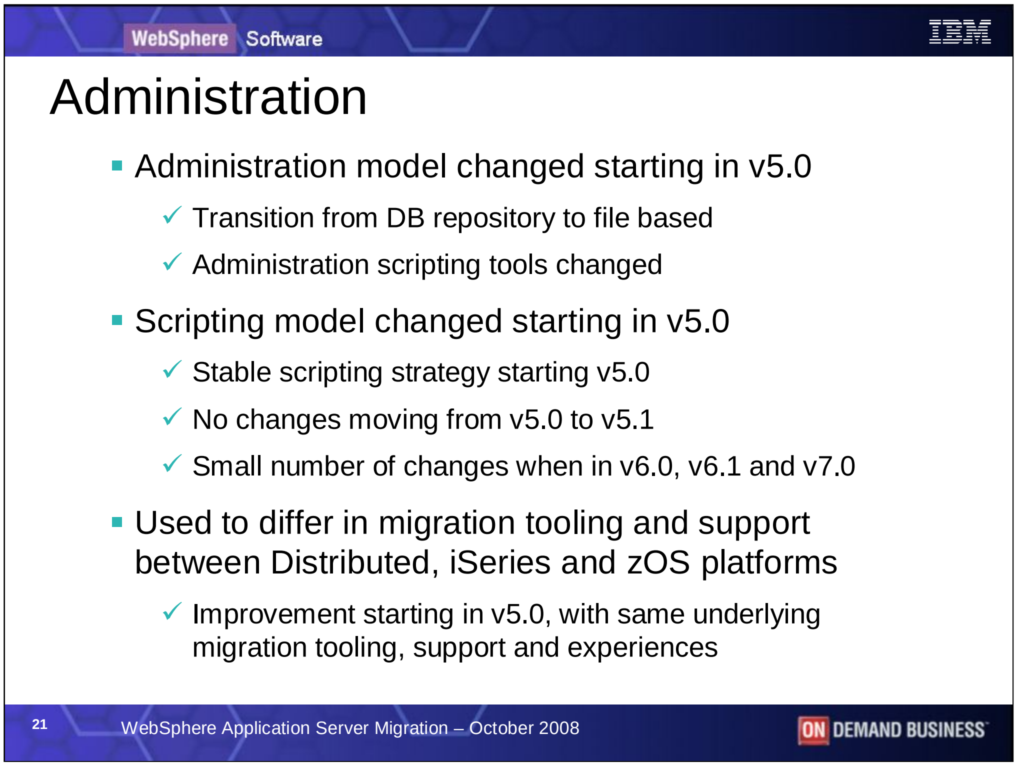


Section

Improving migration story

The “Improving Migration Story” describes the improvements that have been made in WebSphere Application Server Migration over time. This is very important to understand when building a Migration plan because there are major differences in degree of difficulty based on the combination of

- Which WebSphere Application Server you are migrating **from**
- Which WebSphere Application Server you are migrating **to**



WebSphere Software

IBM

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 in v6.0, v6.1 and v7.0
- 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

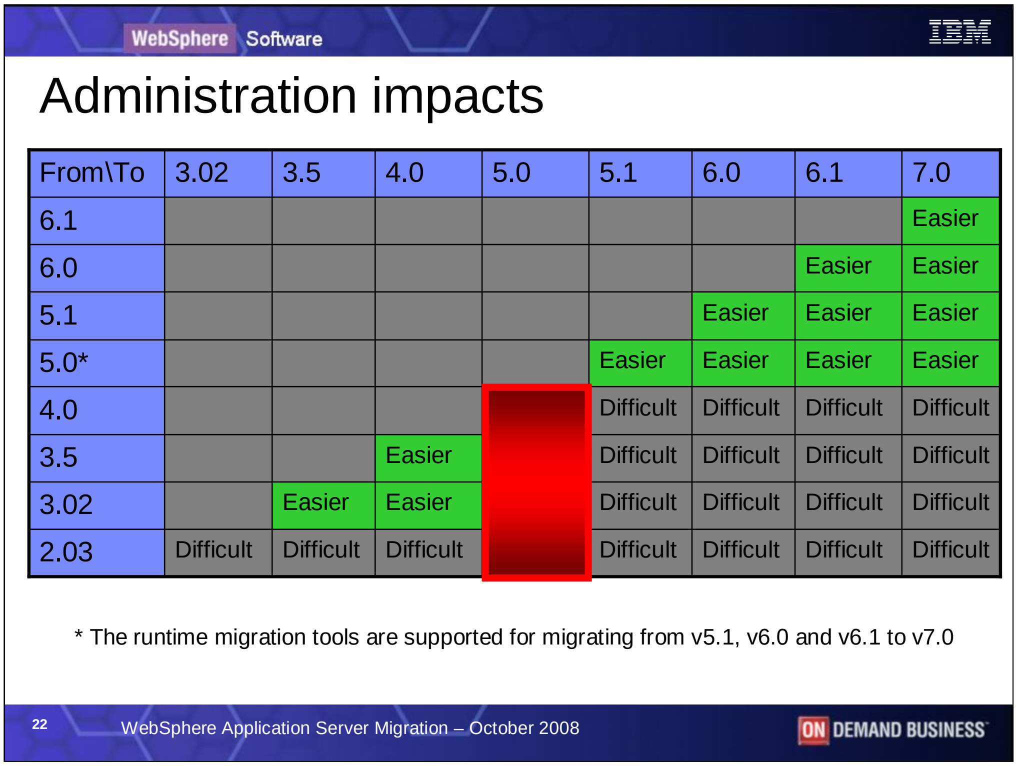
21 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

There are two areas of interest to understand when looking at the combination of versions that are involved in the migration scenario. These areas are Administration and Development. The first area that we are going to look at is administration.

The big change in administration occurs when migrating to WebSphere Application Server version 5.0 or later from version 4.0 or earlier. This is because prior to version 5.0 the administration data was stored in a database. This had both good and bad side effects. In version 5.0 the administration model was re-engineered to use a flat file system instead of a database repository. This had several side effects as well that made this transition more difficult. The administration tooling that was used in prior versions was no longer available on version 5,0 and later. The administration scripts that were written using the wscp support was not longer supported, as well there was not an automated migration path to convert these tools. Also, the XMLConfig command was not supported since the configuration was already in xml files in version 5.0 and later.

However, once this chasm is crossed the new scripting model has been very consistent. No changes were introduced in version 5.1 and few changes were introduced in versions 6.0 and 6.1. There also has been some differences in some of the platforms supported by WebSphere, primarily zOS. Over time there has been increased closure in common tooling and more consistent support arorss all the platforms supported by WebSphere Application Server.



Administration impacts

From\To	3.02	3.5	4.0	5.0	5.1	6.0	6.1	7.0
6.1								Easier
6.0							Easier	Easier
5.1						Easier	Easier	Easier
5.0*					Easier	Easier	Easier	Easier
4.0					Difficult	Difficult	Difficult	Difficult
3.5			Easier		Difficult	Difficult	Difficult	Difficult
3.02		Easier	Easier		Difficult	Difficult	Difficult	Difficult
2.03	Difficult	Difficult	Difficult		Difficult	Difficult	Difficult	Difficult

* The runtime migration tools are supported for migrating from v5.1, v6.0 and v6.1 to v7.0

This table shows some examples of Administration impacts for some of the common scenarios involved in Migration scenarios. This table is organized with the “from” scenarios arranged in rows and “to” versions arranged in columns. The various cells at the intersection of the rows and columns gives you a high level perspective of the effort involved in these various migration scenarios. Some of the more common scenarios are highlighted:

From version 3.5 to version 4.0

This was a simple scenario. The database was still used by both WebSphere versions. There were a few minor changes in the schema but there were few changes required.

From version 4.0 or earlier to version 5.0 or later

This represents the chasm noted earlier, any scenario crossing the v5.0 boundary represents significant work to move the administration scripts to the new model as well as significant training for administrators.

From version v5.0 to version v5.1

This is the simplest of the scenarios because there were no changes in the administration model.

From version v5.0 or v5.1 to version 6.0

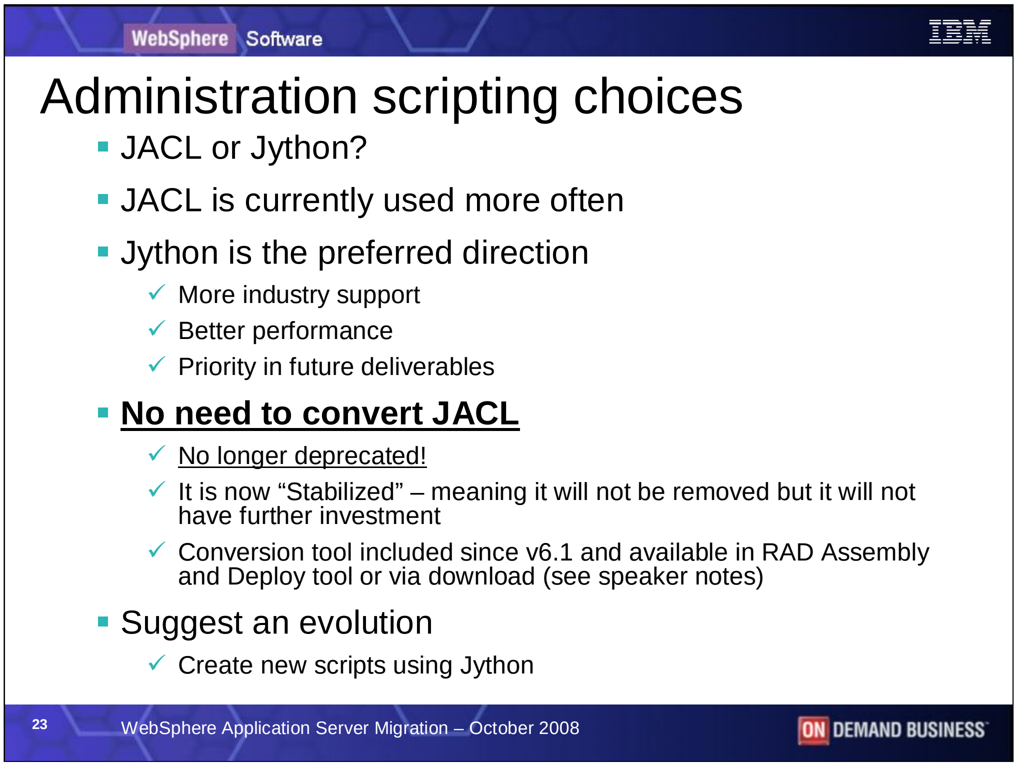
Version 6.0 did introduce a few changes in the administration model and these have to be considered when building a plan for moving to version 6.0 The details of changes to administration scripts are detailed in the appendices of this document.

From version v5.0, v5.1 or v6.0 to version 6.1

Version 6.1 also introduce a few changes in the administration model. These are also detailed in the appendices of this document.

From version v5.1, v6.0 or v6.1 to version 7.0

Version 7.0 is very compatible, new features have been added but existing scripts should still work correctly.



WebSphere Software

IBM

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 need to convert JACL**
  - ✓ No longer deprecated!
  - ✓ It is now “Stabilized” – meaning it will not be removed but it will not have further investment
  - ✓ Conversion tool included since v6.1 and available in RAD Assembly and Deploy tool or via download (see speaker notes)
- Suggest an evolution
  - ✓ Create new scripts using Jython

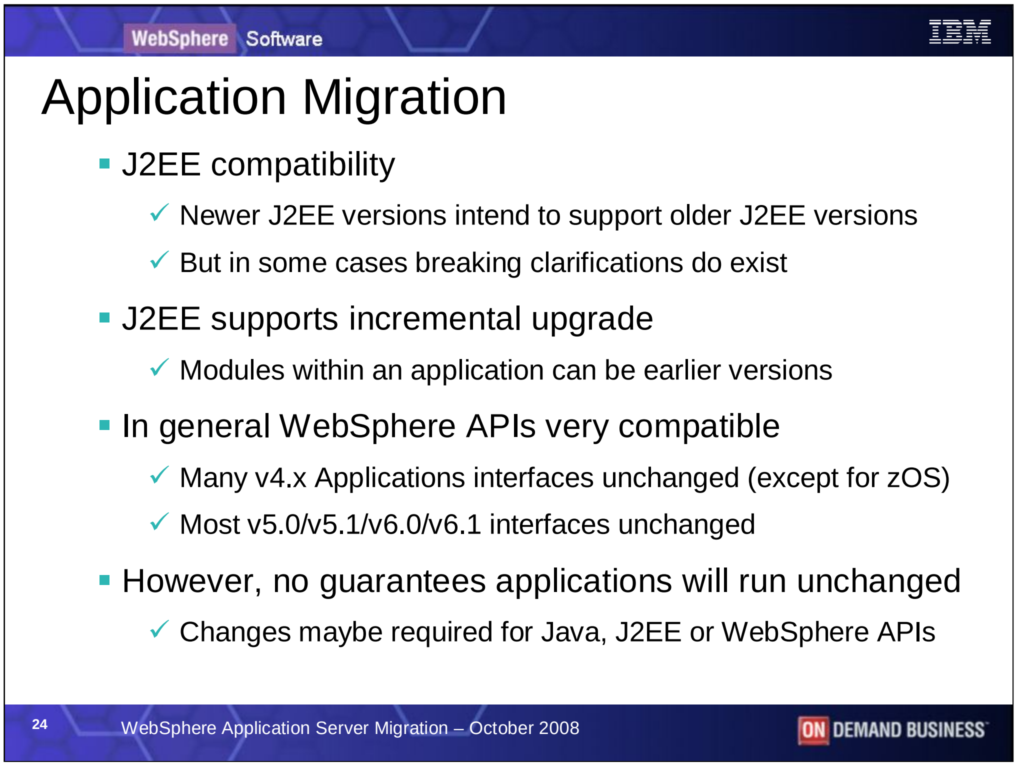
23 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

WebSphere Application Server is moving in the direction of Jython, so any future scripts you may write should be done in Jython and not Jacl. There currently are no plans to drop support for Jacl, so there is no need to rush out and change your scripts.

There are many reasons for the Jacl to Jython change, among those are performance and industry support. If you decide that you are going to change your existing scripts then it is worth looking into the “Jacl2Jython” tool which will automatically convert most of your Jacl code to Jython code and if it cannot convert something it will document that so that you can revisit later. This tool can also be used as a teaching aid for those that only know Jacl(ex: create your script in Jacl and run the tool and inspect/study the mappings). All in all this tool is pretty slick and worth trying out before manual attempts are made.

This tool can be found at the following link: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012144



WebSphere Software

IBM

Application Migration

- J2EE compatibility
  - ✓ Newer J2EE versions intend to support older J2EE versions
  - ✓ But in some cases breaking clarifications do exist
- J2EE supports incremental upgrade
  - ✓ Modules within an application can be earlier versions
- In general WebSphere APIs very compatible
  - ✓ Many v4.x Applications interfaces unchanged (except for zOS)
  - ✓ Most v5.0/v5.1/v6.0/v6.1 interfaces unchanged
- However, no guarantees applications will run unchanged
  - ✓ Changes maybe required for Java, J2EE or WebSphere APIs

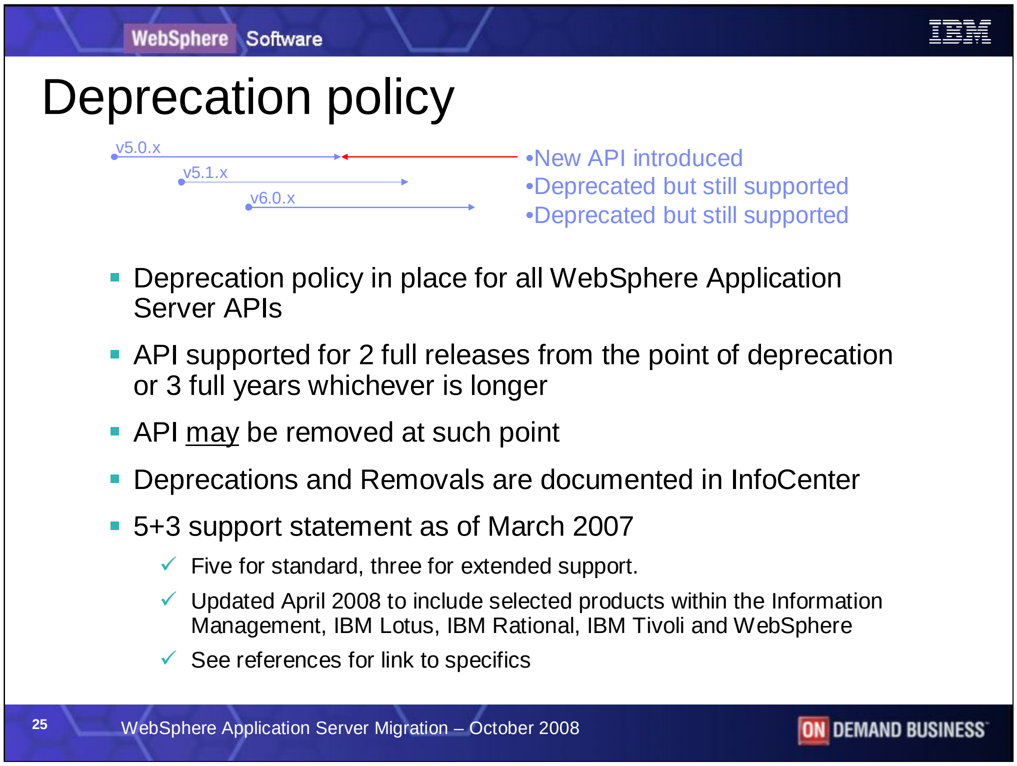
WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

Wherever possible the J2EE specification remains compatible with prior versions of the J2EE specification. This implies that applications written to J2EE v1.2 can be run without change on J2EE v1.4. For the most part this is true, but not always. You should check each individual J2EE specification for source and binary incompatibilities, they are few, but they can represent problems if left unchecked. The most troubling changes are when signatures do not change(can be found at compile time), but depended upon characteristics change(See Servlet v2.4 specification, section SRV.1.6.1 for an example). This is another example of why migration planning is essential: This issue has a good potential to not show it's ugly head until you are well into production.

J2EE offers the capability to incrementally upgrade your applications. J2EE does not force you into the position of migrate all or nothing(ex: In one application you can migrate two of your Web Modules to current specification level and leave three Web Modules at the old specification level. J2EE is flexible enough to allow you upgrade an Enterprise Java Bean (EJB) at a time within the same EJB Module). More information can be found in the following Redbook (Chapter3): http://w3.itso.ibm.com/abstracts/sg246369.html

A great deal of time and resources are invested in making sure WebSphere Application Server APIs remain compatible and in general we have succeeded in this effort.



WebSphere Software

Deprecation policy

v5.0.x
v5.1.x
v6.0.x

•New API introduced
•Deprecated but still supported
•Deprecated but still supported

- 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 standard, three for extended support.
  - ✓ Updated April 2008 to include selected products within the Information Management, IBM Lotus, IBM Rational, IBM Tivoli and WebSphere
  - ✓ See references for link to specifics

WebSphere Application Server has a policy in place to support the APIs for two full releases or three full years whichever is longer. An API “may” be removed at such point. I would take “may” as meaning we will remove the API. Always plan and take actions on the work items that you know about beforehand. Let’s face it, if you choose to gamble and not take preemptive action, the API may just disappear at the most inopportune time. However, as previously described, removing deprecated interfaces should not be done as part of the version migration process because it introduces additional risk. Deprecation mitigation should be done later in the cycle as part of standard application maintenance.

Deprecations and Removals are documented in the InfoCenter and we have included some direct links here:

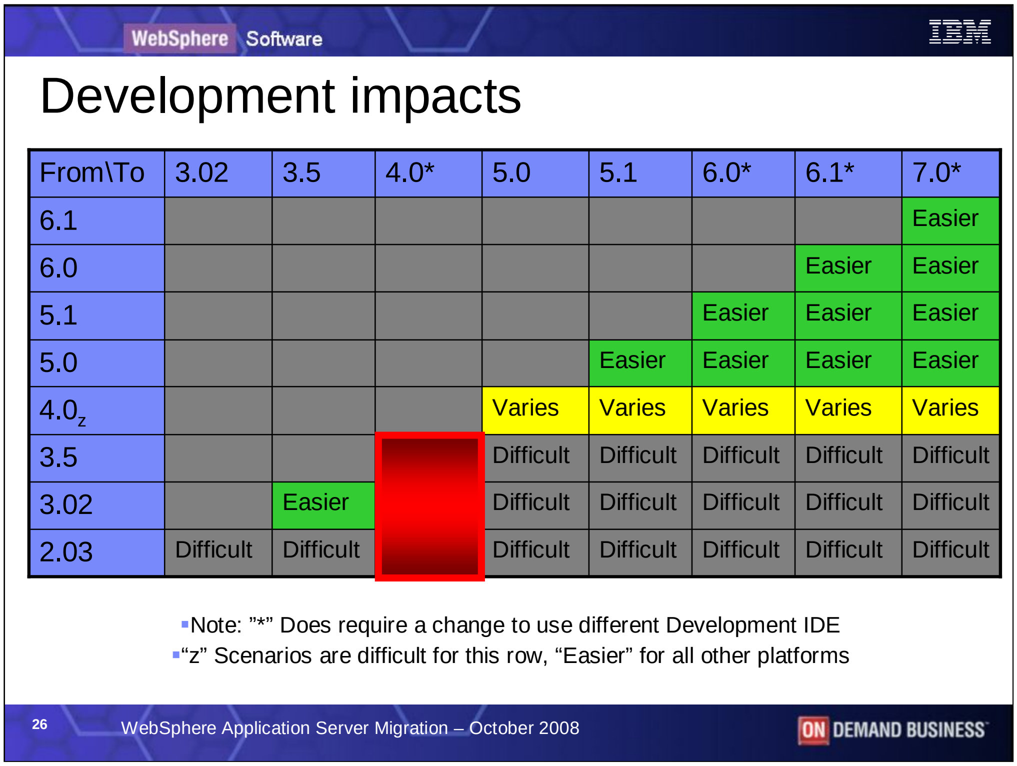
V7.0 Deprecated Stabilized and removed features:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rmig_deprecationlist.html

v6.1 Deprecated and removed features:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rmig_deprecationlist.html

v6.0 Deprecated and removed features:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rmig_deprecationlist.html

v5.1 Deprecated and removed features:
http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rmig_deprecationlist.html



WebSphere Software

Development impacts

From\To	3.02	3.5	4.0*	5.0	5.1	6.0*	6.1*	7.0*
6.1								Easier
6.0							Easier	Easier
5.1						Easier	Easier	Easier
5.0					Easier	Easier	Easier	Easier
4.0_z				Varies	Varies	Varies	Varies	Varies
3.5				Difficult	Difficult	Difficult	Difficult	Difficult
3.02		Easier		Difficult	Difficult	Difficult	Difficult	Difficult
2.03	Difficult	Difficult		Difficult	Difficult	Difficult	Difficult	Difficult

- Note: "*" Does require a change to use different Development IDE
- "z" Scenarios are difficult for this row, "Easier" for all other platforms

26 WebSphere Application Server Migration – October 2008

This table shows some examples of Development impacts for some of the common scenarios involved in Migration scenarios. This table is organized with the "from" scenarios arranged in rows and "to" versions arranged in columns. The various cells at the intersection of the rows and columns gives you a high level perspective of the effort involved in these various migration scenarios. Some of the more common scenarios are highlighted:

From version 3.02 to version 3.5

This was a simple scenario. The programming model was unchanged and the programming specifications either did not change or were very compatible.

From version 3.5 or earlier to version 4.0 or later

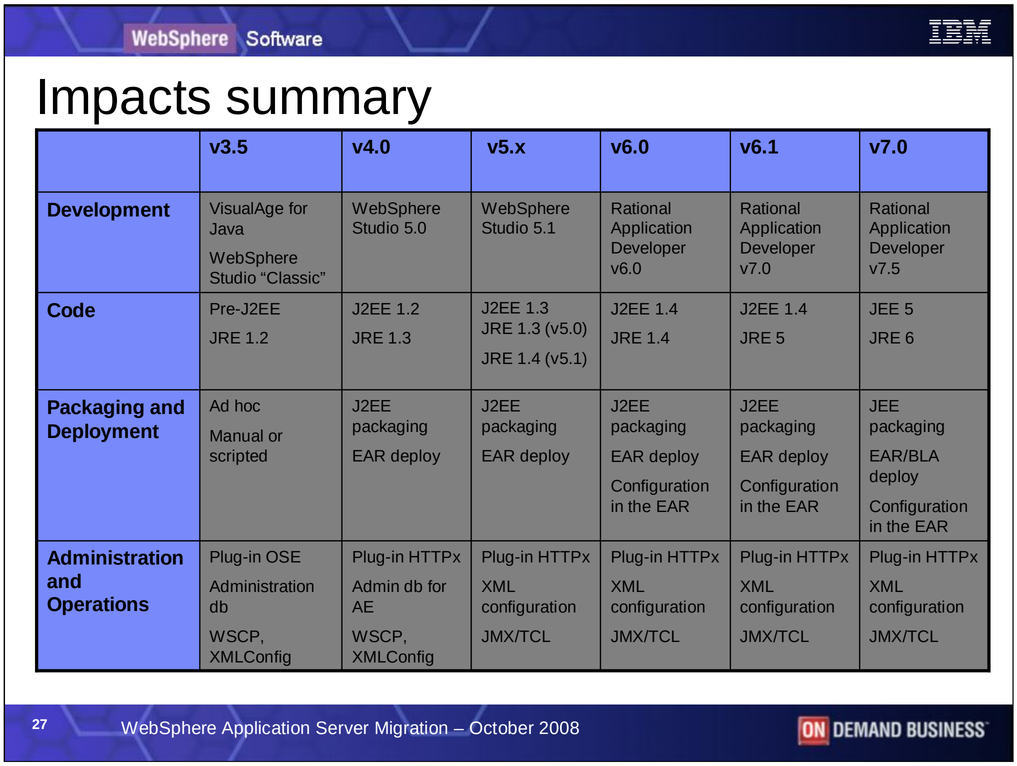
This represents the chasm noted earlier, any scenario crossing the v4.0 boundary had to deal with a change in packaging. This was the first WebSphere Application Server version were J2EE was supported. This required a change in packaging away from the proprietary deployment model. A change in development tooling was also required to use WSAD to support this environment.

From version 4.0 to version 5.0 or later

This is in general a simple scenario. For non-zOS platforms most of that applications require little or no source code changes. However, special care must be taken when deploying and testing the final migrations. Due to the change in WebSphere Application Server administration model there are some changes that are reflected in the form of behavioral changes of applications. Some of these behaviors are subtle and are sometimes difficult to detect until the worst possible time – under load in production. Pay special care to the attached appendices on moving from v4.0. The "z" note is because the zOS migration scenario is more significant since that version of WebSphere Application Server on zOS is not the same as the other platforms supported by WebSphere Application Server and is not as compatible.

From version 5.0 and later

In general these scenarios offer a high degree of compatibility for applications. For example, crossing the J2EE 1.4 boundary of version 6.0 and version 6.1 introduced some incompatibilities due to specification clarifications. Another aspect to be aware of is the change in development tooling from WSAD to RAD in version 6.0 and later. However, there are always challenges in any migration scenario and the appropriate appendix should be carefully checked for further details.


WebSphere Software

Impacts summary

	v3.5	v4.0	v5.x	v6.0	v6.1	v7.0
Development	VisualAge for Java WebSphere Studio "Classic"	WebSphere Studio 5.0	WebSphere Studio 5.1	Rational Application Developer v6.0	Rational Application Developer v7.0	Rational Application Developer v7.5
Code	Pre-J2EE JRE 1.2	J2EE 1.2 JRE 1.3	J2EE 1.3 JRE 1.3 (v5.0) JRE 1.4 (v5.1)	J2EE 1.4 JRE 1.4	J2EE 1.4 JRE 5	JEE 5 JRE 6
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	JEE packaging EAR/BLA 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	Plug-in HTTPx XML configuration JMX/TCL

27 WebSphere Application Server Migration – October 2008

This chart summarizes some of the key aspects of the different WebSphere Application Server versions. The change in development tooling is noted. In these scenarios there is migration support between the different levels of the same tool. The J2EE and JDK levels should be noted since these can cause some incompatibilities.

Another question that is often asked is the "Configuration in EAR" option that was introduced in WebSphere Application Server version 6.0. This support is primarily intended to enhance testing environments in that configuration objects such as resources can be stored directly in the application EAR.. There may be some policies in place in your company about controlling what resources are deployed in the production environment. If so then the "Configuration in EAR" option should not be used for production environments. This can be done using both the wsadmin scripts and the administration console.

option on wsadmin application install is: noprocessEmbeddedConfig

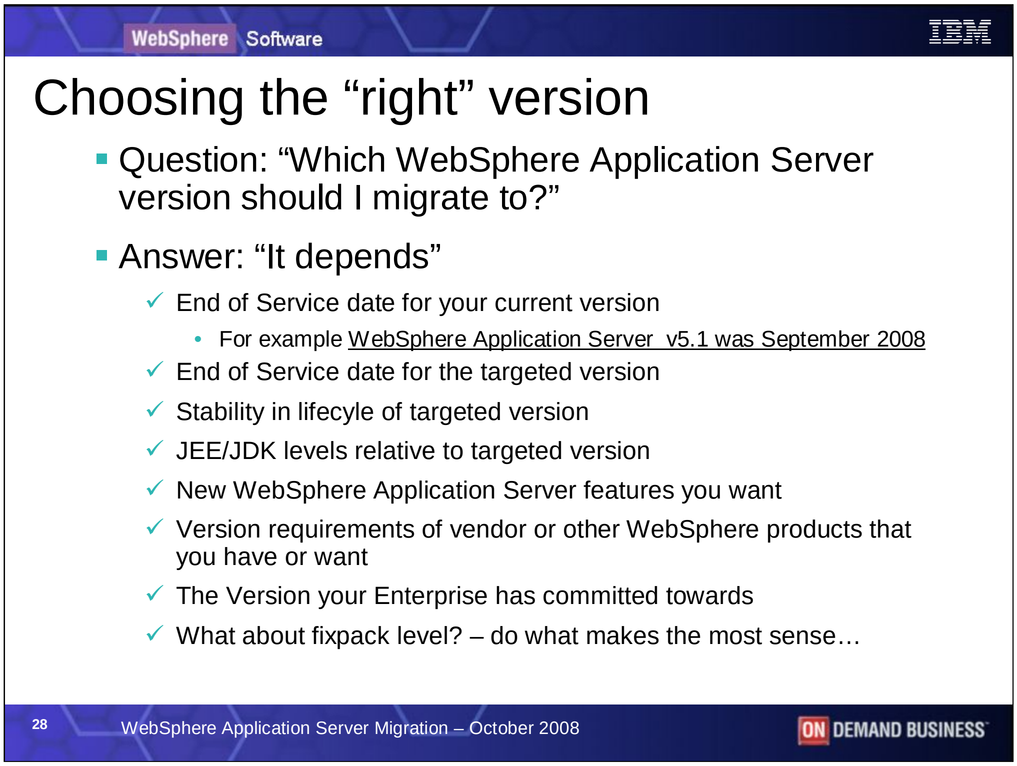
admin console: "Process embedded configuration ", which is defaulted to false

source:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/rxml_taskoptions.html

The other aspect of putting configuration in the enterprise application is if configuration changes between versions. The enterprise applications are updated to handle this if the migration runtime tools are used, however they are not processed if the migration runtime tools are not used and must be done manually. Typically there are few changes that would have to be made but it is a factor to consider.

This chart also re-enforces the earlier description of the runtime tooling differences starting in WebSphere Application Server version 5.0.

For WebSphere Application Server version 7.0 there are a few changes to consider. The first is the Development tools are not RAD 7.5. There is a lot of commonality with prior RAD versions. J2EE is now JEE and has been upgraded to 5, the latest and greatest. The Java runtime is version 6, also an upgrade from WebSphere Application Server v6.1. New in WebSphere Application Server v7.0 is the concept of BusinessLevelApplications (BLAs). These are a composition model that extends the notion of "Application". It manages JEE and non-JEE artifacts like SCA packages, libraries, proxy filters etc. However, standard EAR files are still supported.



WebSphere Software

Choosing the “right” version

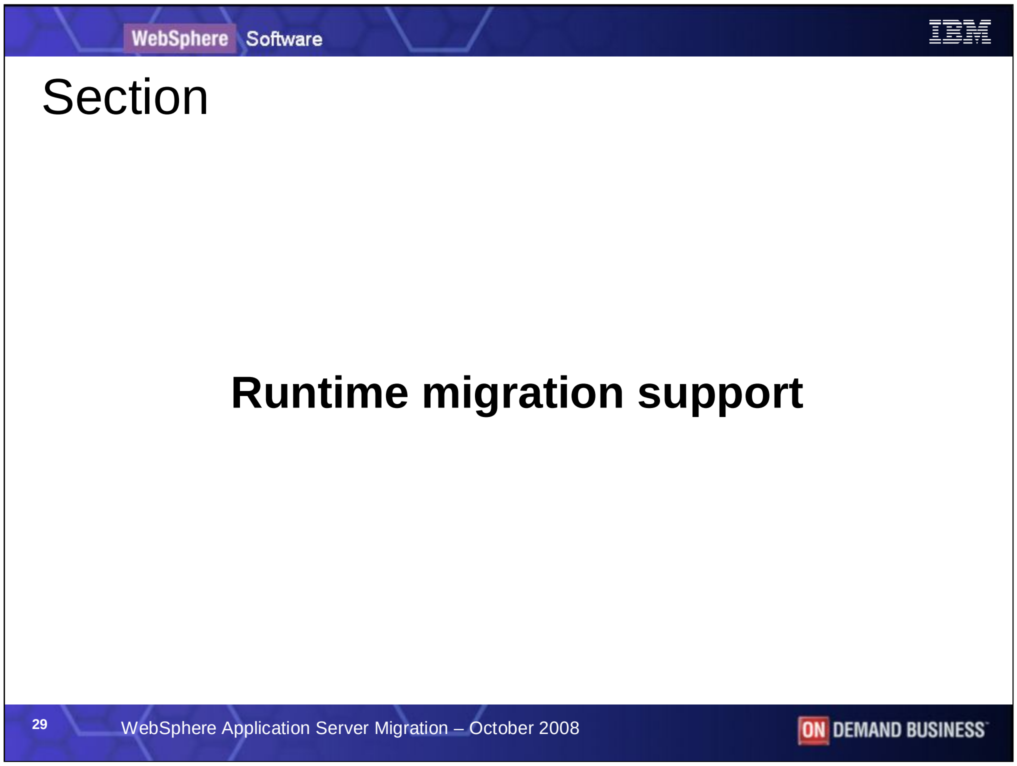
- Question: “Which WebSphere Application Server version should I migrate to?”
- Answer: “It depends”
  - ✓ End of Service date for your current version
    - For example <u>WebSphere Application Server v5.1 was September 2008</u>
  - ✓ End of Service date for the targeted version
  - ✓ Stability in lifecyle of targeted version
  - ✓ JEE/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…

28 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

This is a typical question. Sometimes it would seem that the answer is obvious based on the lifecycles of the various WebSphere Application Server versions at a particular point in time. In many cases the answer is obvious but as you can see from all the factors in the list there are more things to be considered than what is the most recent WebSphere Application Server version. Typically you will want to consider having the longest possible life cycle of the WebSphere Application Server version that you will migrate into. However, you will also have to consider when you believe you will have sufficient stability of any WebSphere Application Server version. We would like to think that a WebSphere Application Server is immediately stable when it is first released but that has not always been the case in the past. Sometimes specification levels are an important consideration. For example, WebSphere Application Server versions 5.1 and 6.1 are attractive to a number of customers just due to the JDK level. One careful consideration that was previously discussed is the required WebSphere Application Server that some WebSphere stack products or 3rd party vendors may require in order to support their product. This may require supporting more than one WebSphere Application Server version at the same time.

The question of fixpack level to target often comes up. The fixpack level that is installed on the existing WebSphere Application Server version typically is not important. The one caveat is for Interoperability support that is described in a later section. If you will be using the runtime migration tooling the fixpack on the existing WebSphere Application Server is not relevant because all the runtime migration tooling runs on the new WebSphere Application Server version. The fixpack level of the new WebSphere Application Server version also is worth considering. It is a good general idea to support the “latest reasonable fixpack level” of the new WebSphere Application Server version. This is so you will ge the latest fixes and not have to retest with the latest fixpack level soon after migration. As always though there are practical considerations. If you have done significant testing on a particular fixpack and a new one is available – should you use it instead and redo all your testing? Probably not, in those cases it is probably best to proceed as planned and pick up the next fixpack using your existing fixpack installation strategy. You should take a look at the included APAR lsi to check for fixes to problem areas that you may have experienced os will us eheavily.



WebSphere Software

IBM

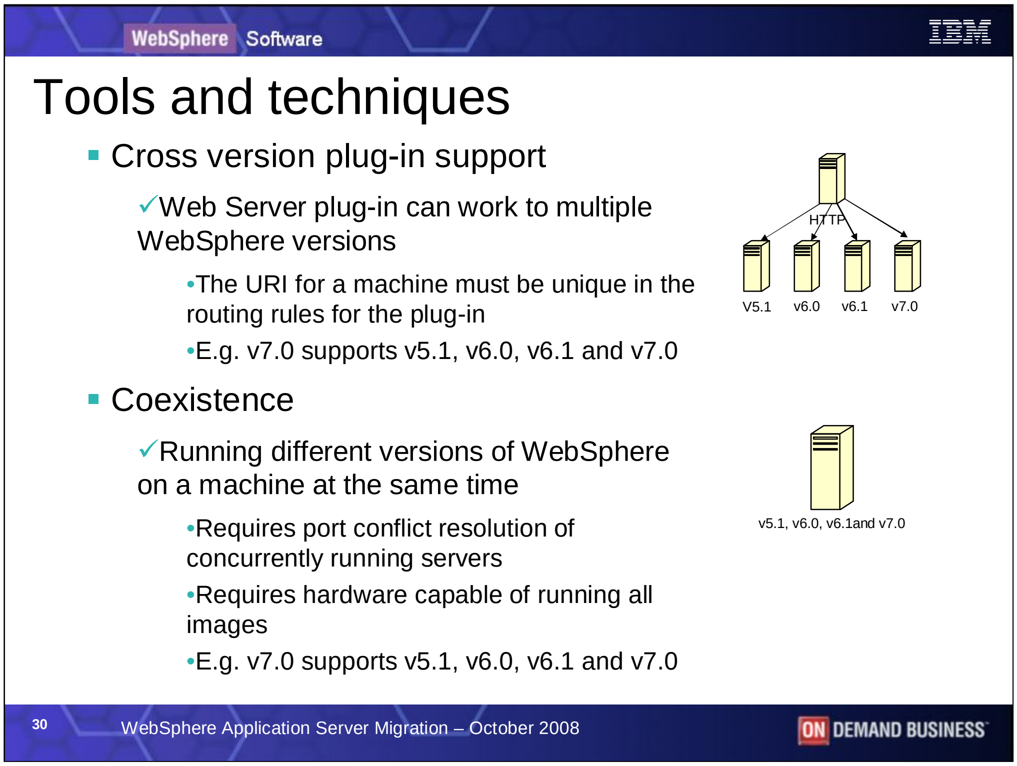
Section

Runtime migration support

29 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

The “Runtime Migration support” section describes the tools and techniques that can be used to migrate the WebSphere Application Server Migration between versions



WebSphere Software

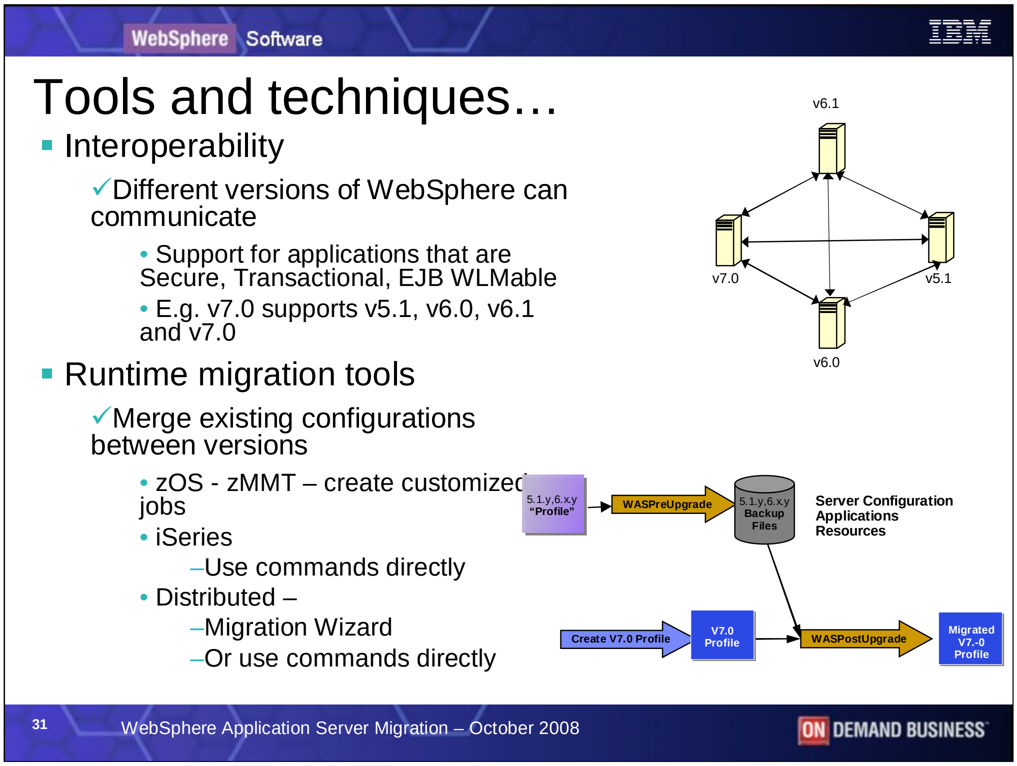
Tools and techniques

- 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. v7.0 supports v5.1, v6.0, v6.1 and v7.0
- 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. v7.0 supports v5.1, v6.0, v6.1 and v7.0

30 WebSphere Application Server Migration – October 2008

Both Cross version plug-in support and Coexistence are supported mechanisms that can be used to do incremental migration. Cross version plug-in support allows you to first upgrade the HTTP Server version to the latest one required for the WebSphere Application Server version that is the target of the migration. This allows you to route traffic to all supported WebSphere Application Server versions while upgrading your environment.

Coexistence allows for more than one version of WebSphere Application Server to be installed on the same physical hardware at the same time. This allows you to not require duplicate additional physical hardware to be available for migration scenarios. You can incrementally support these multiple versions concurrently. If you want to have multiple instances of WebSphere Application Server running at the same you must ensure the common requirements of not having conflicting port definitions and ensure there is sufficient resources (processor capacity, disk space and memory) available for concurrent execution. Alternatively you can keep the old WebSphere Application Server version installed during the migration process to ensure you have a quick fallback plan for disaster recovery.



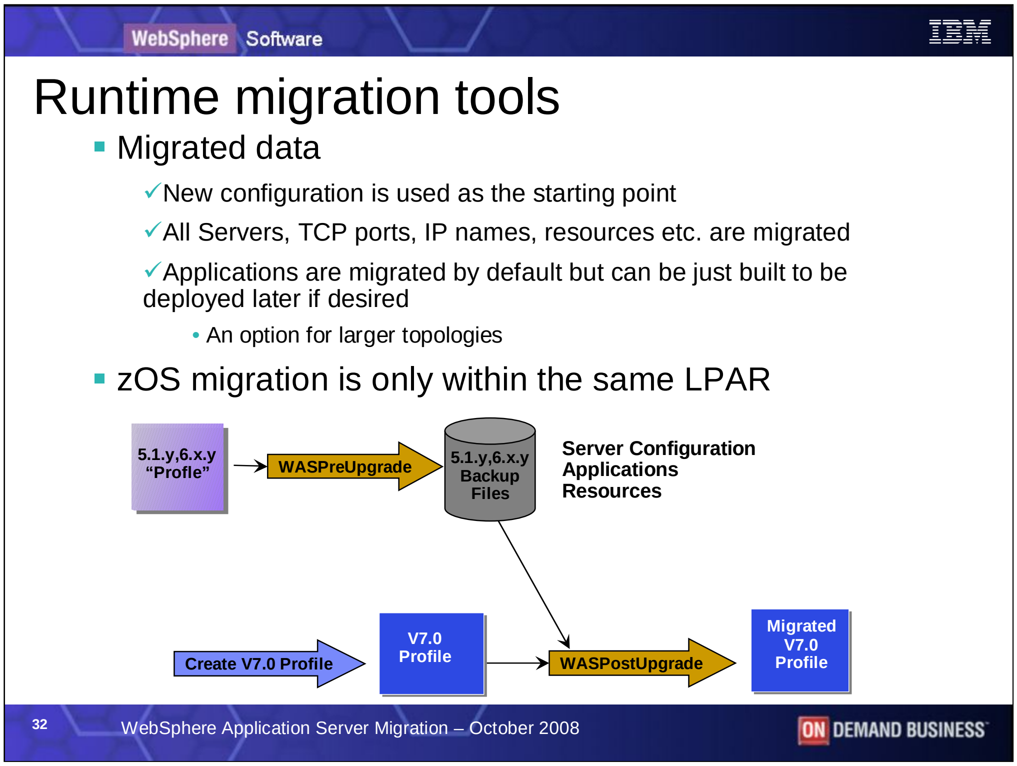
Tools and techniques…

- Interoperability
  - ✓Different versions of WebSphere can communicate
    - Support for applications that are Secure, Transactional, EJB WLMable
    - E.g. v7.0 supports v5.1, v6.0, v6.1 and v7.0
- Runtime migration tools
  - ✓Merge existing configurations between versions
    - zOS - zMMT – create customized jobs
    - iSeries
      - –Use commands directly
    - Distributed –
      - –Migration Wizard
      - –Or use commands directly

Interoperability is simply the ability of different WebSphere Application Server versions to communicate with each other. This enables the ability to do incremental migration of nodes that need to interoperate and avoid the “big bang requirement of migrating all nodes at the same time. WebSphere Application Server Version 6.1 is generally interoperable with WebSphere Application Server Version 5.x and Version 6.0.x. However, there are specific requirements to address for each version. In general, you should apply the latest fix level to support interoperability. See the following website for more information including any specific fixpack levels that are either required or recommended.

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.base.doc/info/aes/ae/tmig_interop.html

The runtime migration tooling gives you the capability of copying the existing configuration from the WebSphere Application Server that you are currently using to the configuration of the new WebSphere Application Server that you will now be using. This is not the only approach that can be used when migrating but it is an option. There are different interfaces on different platforms which all use the same underlying tooling support. The zOS platform uses zMMT panels to create jobs that will call the runtime migration tooling. Distributed (all platforms except zOS and i5OS) have a Migration wizard that makes the job easier. All platforms except zOS can call the underlying migration tooling scripts directly. More details on what this tooling does is discussed in more detail.



WebSphere Software IBM

Runtime migration tools

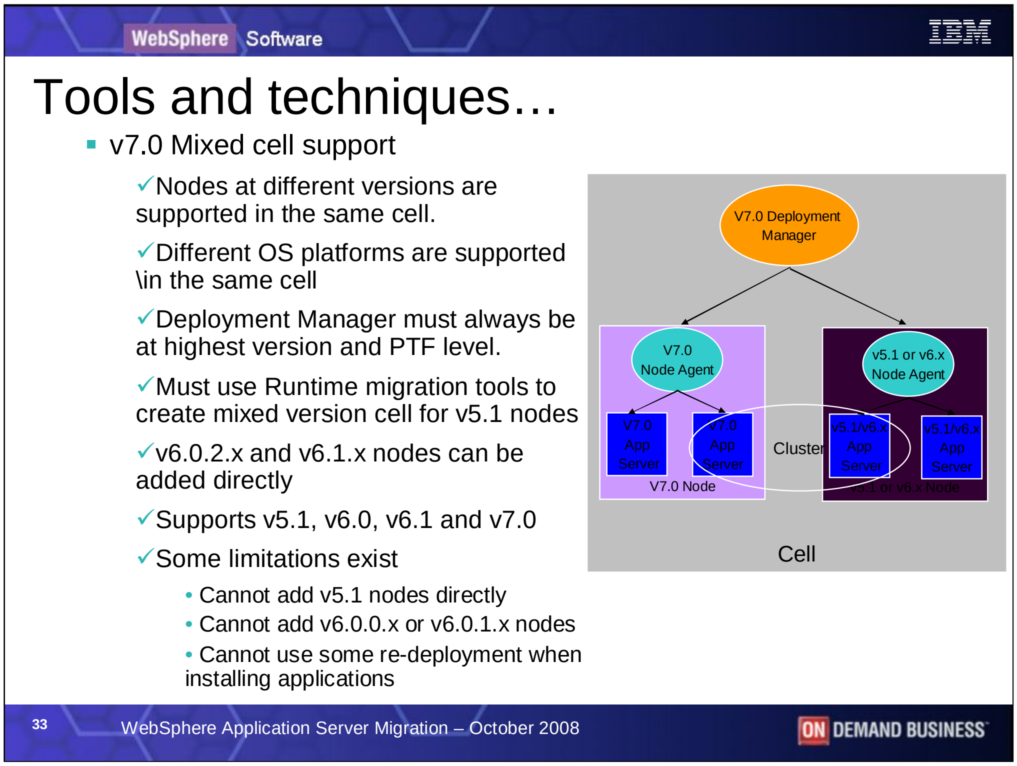
- Migrated data
  - ✓New configuration is used as the starting point
  - ✓All Servers, TCP ports, IP names, resources etc. are migrated
  - ✓Applications are migrated by default but can be just built to be deployed later if desired
    - An option for larger topologies
- zOS migration is only within the same LPAR

5.1.y,6.x.y "Profle"
WASPreUpgrade
5.1.y,6.x.y Backup Files
Server Configuration
Applications
Resources
Create V7.0 Profile
V7.0 Profile
WASPostUpgrade
Migrated V7.0 Profile

32 WebSphere Application Server Migration – October 2008 ON DEMAND BUSINESS

The runtime migration tools do not destroy the configuration of the existing WebSphere Application Server that you are currently using. This is important for a potential fallback plan as discussed earlier. The first step after the new WebSphere Application Server is installed is to create a new profile in the new WebSphere Application Server image. For some scenarios, such as deployment manager scenarios, certain values must match with the existing configuration. For deployment manager nodes this is the cell name. For Distributed platforms the Migration Wizard can be used to make this job easier. The runtime migration tools are two scripts. One copies the configuration from the existing WebSphere Application Server into a backup directory. The other scripts takes the content of that backup directory and merges it into the new profile that was created in the new WebSphere Application Server image. By default all the configuration objects, including servers, resources, virtual hosts, etc, are merged into the new profile. All applications are repackaged from the existing WebSphere Application Server configuration and are installed into the new profile of the new WebSphere Application Server image. There are a variety of settings on these commands that will be covered in detail in one of the appendices of this material.

One option that is typically not understood is the features of the migration tools that can be used for various scenarios like large topologies or more specifically cells with several hundred or more applications. The migration tools create installable enterprise applications from deployed applications in the old cell and installs them in the new cell. This can be very time consuming. One option is to just have the migration tools create installable enterprise applications but not install them in the new cell. These applications can be added later iteratively. This can address scenarios where migration has to be done in a certain maintenance window but does not have to be completed during that time.



WebSphere Software

Tools and techniques…

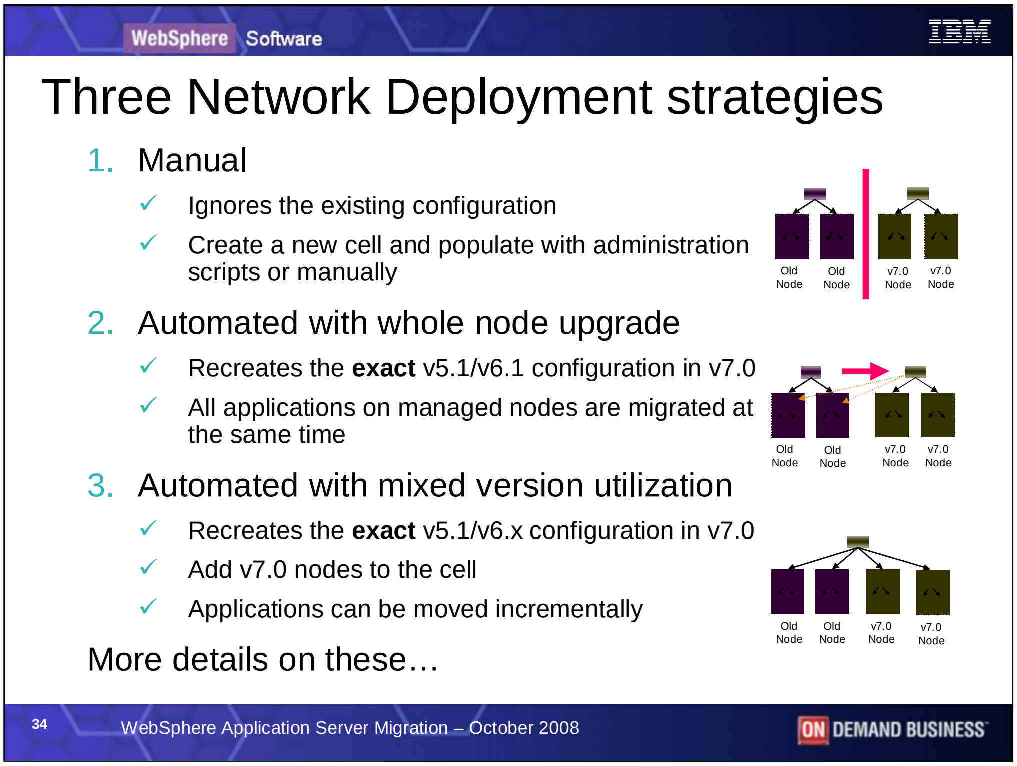
- v7.0 Mixed cell support
  - ✓Nodes at different versions are supported in the same cell.
  - ✓Different OS platforms are supported \in the same cell
  - ✓Deployment Manager must always be at highest version and PTF level.
  - ✓Must use Runtime migration tools to create mixed version cell for v5.1 nodes
  - ✓v6.0.2.x and v6.1.x nodes can be added directly
  - ✓Supports v5.1, v6.0, v6.1 and v7.0
  - ✓Some limitations exist
    - Cannot add v5.1 nodes directly
    - Cannot add v6.0.0.x or v6.0.1.x nodes
    - Cannot use some re-deployment when installing applications

33 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

For deployment manager configurations mixed version support is also provided. This is another facet of incremental upgrade that allows managed nodes to be migrated independently. This support was first available in WebSphere Application Server version 6.0 and with it you can manage nodes at version 5.0, 5.1 and 6.0. For WebSphere Application Server version 7.0 all nodes from 5.1 through 7.0 (except v6.0.0.x and v6.0.1.x) are supported. Different operating systems are also supported transparently. There are several details to be aware of when planning to use this support. The deployment manager must always be at the highest version and fixpack level. The runtime Migration tooling is the only way to use this support in order to setup the initial environment. More details on the capabilities and limitations of this support can be found in one of the appendices of this material.

The primary intent of this support is to provide a good migration path from previous WebSphere Application Server versions. Although there is no time limit on how long a cell can run in mixed version it is usually considered transitionary and is not recommended for long periods of use.



WebSphere Software

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** v5.1/v6.1 configuration in v7.0
   - ✓ All applications on managed nodes are migrated at the same time
3. Automated with mixed version utilization
   - ✓ Recreates the **exact** v5.1/v6.x configuration in v7.0
   - ✓ Add v7.0 nodes to the cell
   - ✓ Applications can be moved incrementally

More details on these…

34 WebSphere Application Server Migration – October 2008

There are three different common alternatives that can be chosen from when migrating deployment manager configurations. There are different trade-offs and starting points for each alternative. There is no one prescribed approach that is applicable to all environments and situations. The best approach is to carefully consider the options and their trade-offs and make the right choice for you. This chart will briefly introduce the three options. The details of each will be covered further in upcoming charts.

Manual

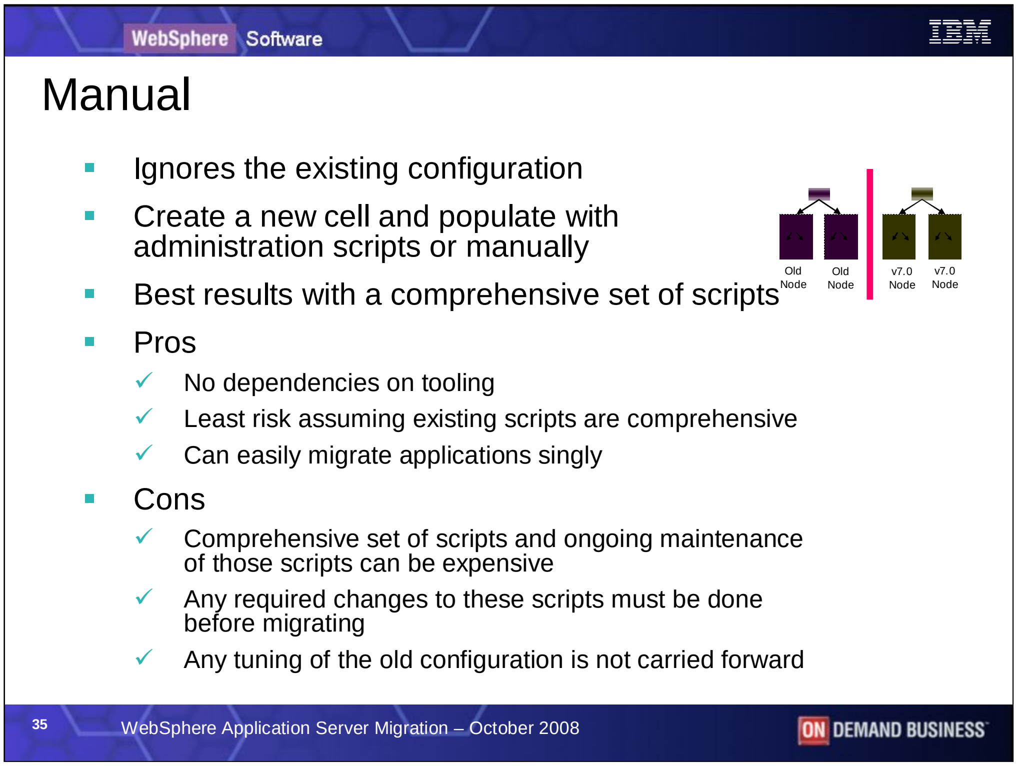
This approach ignores the existing WebSphere Application Server configuration and sets up a new configuration in parallel to the existing WebSphere Application Server configuration. This choice works very well if a rigorous and complete set of wsadmin scripts are already available and strictly used in standard administration that can be used to setup the environment. This is less reliable if there are not existing scripts that meet this criteria. This approach has the advantage of it enables the ability to more individually migrate applications. It is also simpler and has fewer dependencies assuming it does meet the described criteria. We will see why this is true when we explore more details of the other options

Automated with whole node upgrade

This approach uses the runtime migration tooling that was previously described to move the existing WebSphere Application Server configuration to the new WebSphere Application Server configuration profile. This has the advantage of replicating the exact configuration from the existing configuration, no data is lost. All configuration tuning and tweaking will be carried forward into the new configuration. This approach does require migrating all data on each managed node at the same time. This can cause complications for some large topologies. When the whole node is migrated all the applications that are deployed on that node must be tested and available on the newest existing WebSphere Application Server version at the same time. This can cause additional scheduling impacts. More on this will be discussed later.

Automated with mixed node utilization

This approach builds off of the previous one in that nodes of the new existing WebSphere Application Server configuration version can be added to the migrated deployment manager configuration. These nodes can then be used to more individually deploy applications. Note that these new nodes are not replicated from a node that already exists in the previous existing WebSphere Application Server configuration. It is added as a brand new node. The migrated cell level configuration is available to each of the new nodes and is shared but any node-specific information is not copied. More on this will also be discussed later.


WebSphere Software
IBM
Old Node
Old Node
v7.0 Node
v7.0 Node
ON DEMAND BUSINESS

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

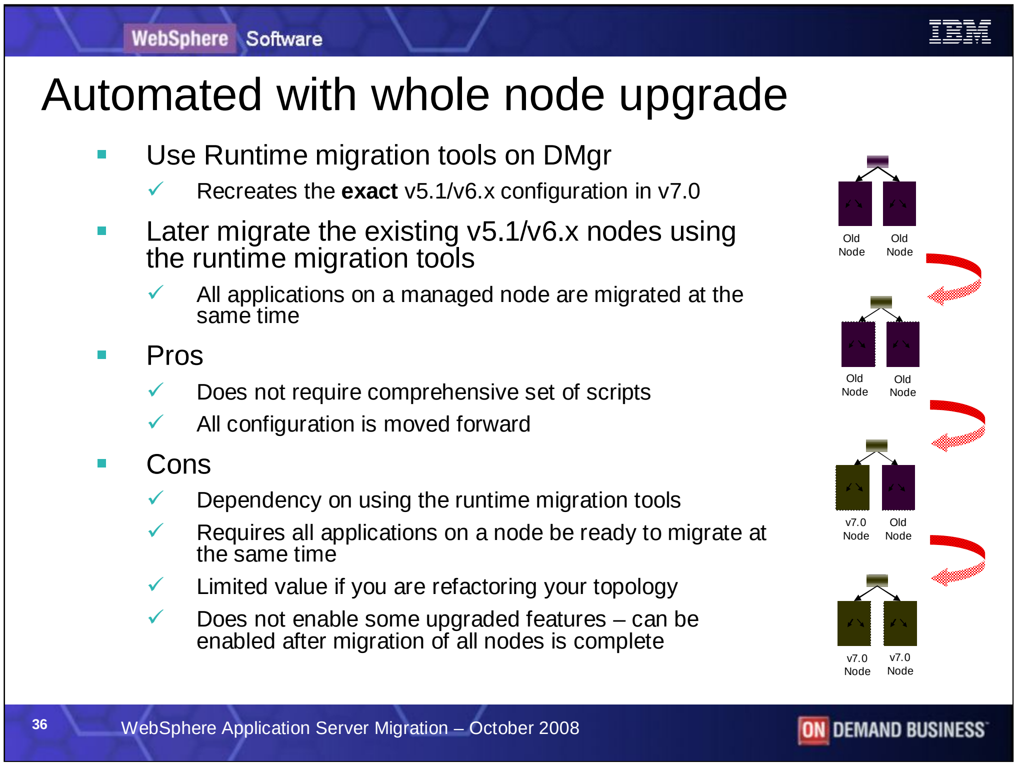
35 WebSphere Application Server Migration – October 2008

As previously introduced, this approach ignores the existing WebSphere Application Server configuration and sets up a new configuration in parallel to the existing WebSphere Application Server configuration. This choice works very well if a rigorous and complete set of wsadmin scripts are already available and strictly used in standard administration that can be used to setup the environment. The reason “rigorous and complete set of wsadmin scripts are already available and strictly used in standard administration ” is key and is not obvious. Many customers may believe that they meet this criteria. Careful examination of the this criteria is important. If you do not have these types of scripts then you may lose valuable tuning and tweaking configuration data that was learned and earned in the existing WebSphere Application Server configuration. However, if your production environment does make heavy use of scripts or you can faithfully recreate your WebSphere Application Server configuration then this is an excellent option.

Note that if you want to reorganize your topology then this approach may be the only alternative. More on this will be covered later.

One other consideration when using this approach is that any required changes to your wsadmin scripts must be completed before migrating. For the other approaches the runtime migration tooling can be used to mitigate some of these changes but they are not used for this approach. You can also see when comparing the steps of using this approach that it is simpler than the other approaches. See the appendices in the back of this material for more information on these wsadmin script changes.

This approach has the advantage of it enables the ability to more individually migrate applications. It is also simpler and has fewer dependencies assuming it does meet the described criteria. We will see why this is true when we explore more details of the other options



Automated with whole node upgrade

- Use Runtime migration tools on DMgr
  - ✓ Recreates the **exact** v5.1/v6.x configuration in v7.0
- Later migrate the existing v5.1/v6.x 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 – can be enabled after migration of all nodes is complete

This approach uses the runtime migration tooling that was previously described to move the existing WebSphere Application Server configuration to the new WebSphere Application Server configuration profile. This has the advantage of replicating the exact configuration from the existing configuration, no data is lost. Note that the advantage of copying the exact configuration can also be a disadvantage is some cases. For example, if you want to rearrange your node topology or node names then this may not be the best approach because all the node configuration and node names will be carried forward exactly as they are in the existing WebSphere Application Server configuration. Another common example to consider is if you want to rearrange your server or application topologies. This is less of a change and you could do this after the initial migration fairly easily.

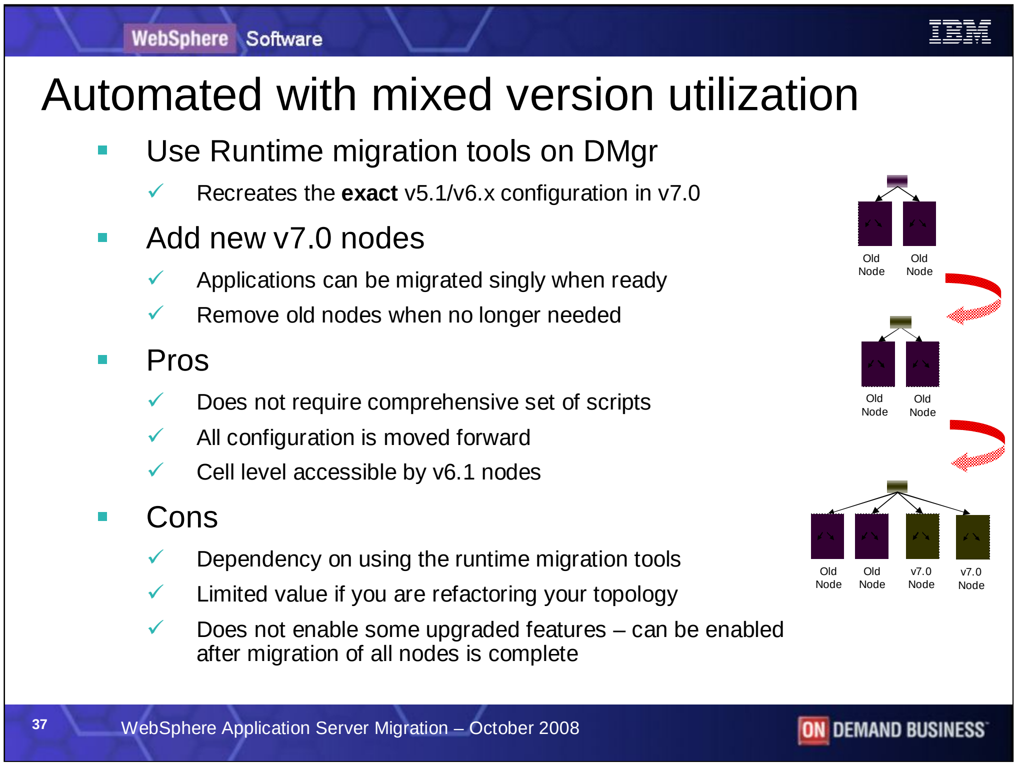
This approach does require migration of all data on each managed node at the same time. This can cause complications for some large topologies. When the whole node is migrated all the applications that are deployed on that node must be tested and available on the newest existing WebSphere Application Server version at the same time. This can cause additional scheduling impacts because all those application teams must be focused on delivering this in the same general timeframe. This may be difficult to manage because most of the tine the development teams are focused on adding new features or function to their existing applications and not on migrating to a new WebSphere Application Server version.

This approach also enables mixed version support for those cases where you would like to use this capability to manager multiple versions for the migration transition period. The tooling also provides script compatibility support where your existing wsadmin scripts do not have to all be changed immediately.

One other factor to consider is the usage of new features in the new WebSphere Application Server version and how the runtime migration tooling is involved. In order to provide a smooth migration path and ensure mixed version support works the runtime migration tooling carries the exact configuration forward. If there is a newer version of equivalent support in the new WebSphere Application Server version it is not used. In most cases this is not a problem because typically new features must be coded to and tested in an independent cycle anyway. For some case, e.g. the new Security model of v6.1, this is not as optimal because the new Security model could be used directly without source code changes to applications. In general these new features must be enabled post-migration.

The basic steps of this approach are:

1. Migrate the deployment manager to the newest WebSphere Application Server version
2. Migrate each managed node until all the nodes have been migrated.


WebSphere Software
IBM
Old Node
Old Node
Old Node
Old Node
Old Node
Old Node
v7.0 Node
v7.0 Node
ON DEMAND BUSINESS

Automated with mixed version utilization

- Use Runtime migration tools on DMgr
  - ✓ Recreates the **exact** v5.1/v6.x configuration in v7.0
- Add new v7.0 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 – can be enabled after migration of all nodes is complete

This approach builds off of the previous one in that nodes of the new existing WebSphere Application Server configuration version can be added to the migrated deployment manager configuration. These nodes can then be used to more individually deploy applications. Note that these new nodes are not replicated from a node that already exists in the previous existing WebSphere Application Server configuration. It is added as a brand new node. The migrated cell level configuration and cluster configuration is available to each of the new nodes and is shared but any node-specific information is not copied. More on this will also be discussed later.

This approach has the same characteristics as the “**Automated with whole node upgrade approach”.** It just provides another option that can be used to incrementally approach migration.

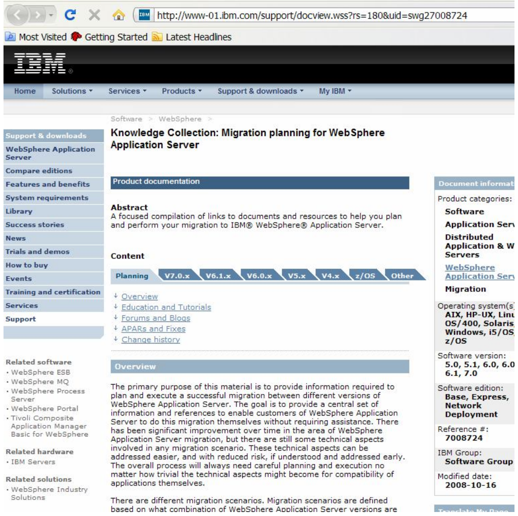
Section

Version specific Highlights

The “Migration Strategies” shows the typical strategies that can be chosen, based on each individual situation

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
  - ✓ http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27008724



Knowledge Collection: Migration planning for WebSphere Application Server

Product documentation

Abstract
A focused compilation of links to documents and resources to help you plan and perform your migration to IBM® WebSphere® Application Server.

Content

Overview

The primary purpose of this material is to provide information required to plan and execute a successful migration between different versions of WebSphere Application Server. The goal is to provide a central set of information and references to enable customers of WebSphere Application Server to do this migration themselves without requiring assistance. There has been significant improvement over time in the area of WebSphere Application Server migration, but there are still some technical aspects involved in any migration scenario. These technical aspects can be addressed easier, and with reduced risk, if understood and addressed early. The overall process will always need careful planning and execution no matter how trivial the technical aspects might become for compatibility of applications themselves.

There are different migration scenarios. Migration scenarios are defined based on what combination of WebSphere Application Server versions are

Document informat

Product categories:
Software
Application Serv
Distributed Application & W Servers
WebSphere Application Serv
Migration

Operating system(s
AIX, HP-UX, Linu OS/400, Solaris Windows, i5/OS z/OS

Software version:
5.0, 5.1, 6.0, 6.0 6.1, 7.0

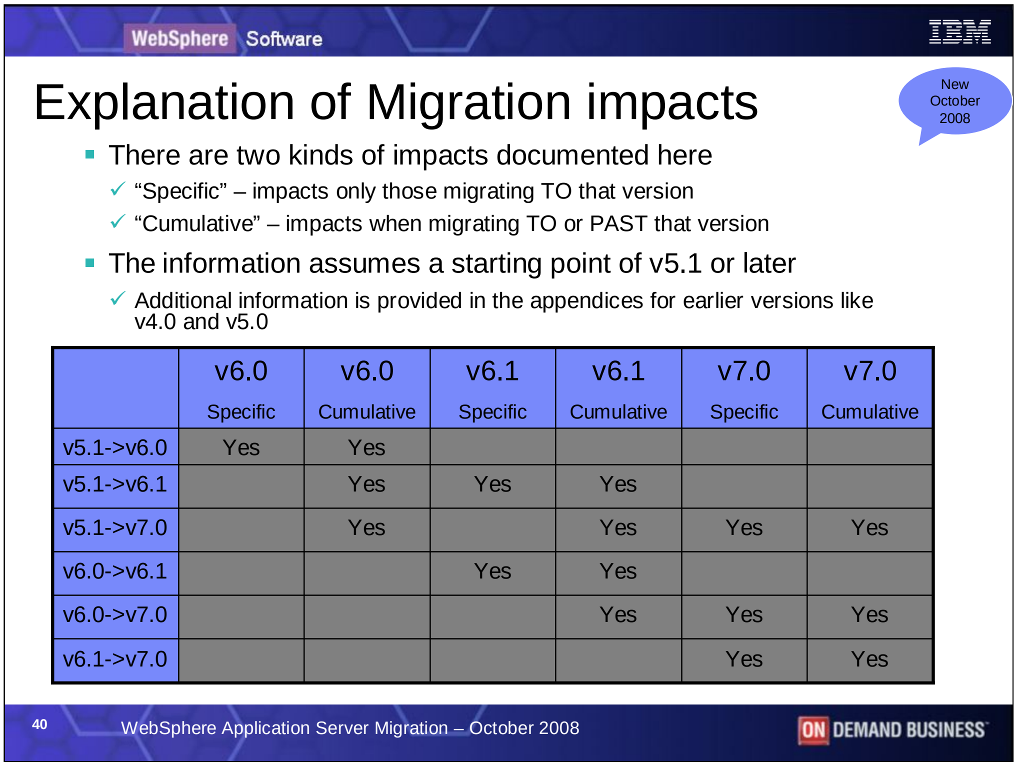
Software edition:
Base, Express, Network Deployment

Reference #:
7008724

IBM Group:
Software Group

Modified date:
2008-10-16

This information has been detailed with speaker notes and is available to anyone at http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27008724. It will be updated as more information is learned. This information is available to everyone.



Explanation of Migration impacts

New October 2008

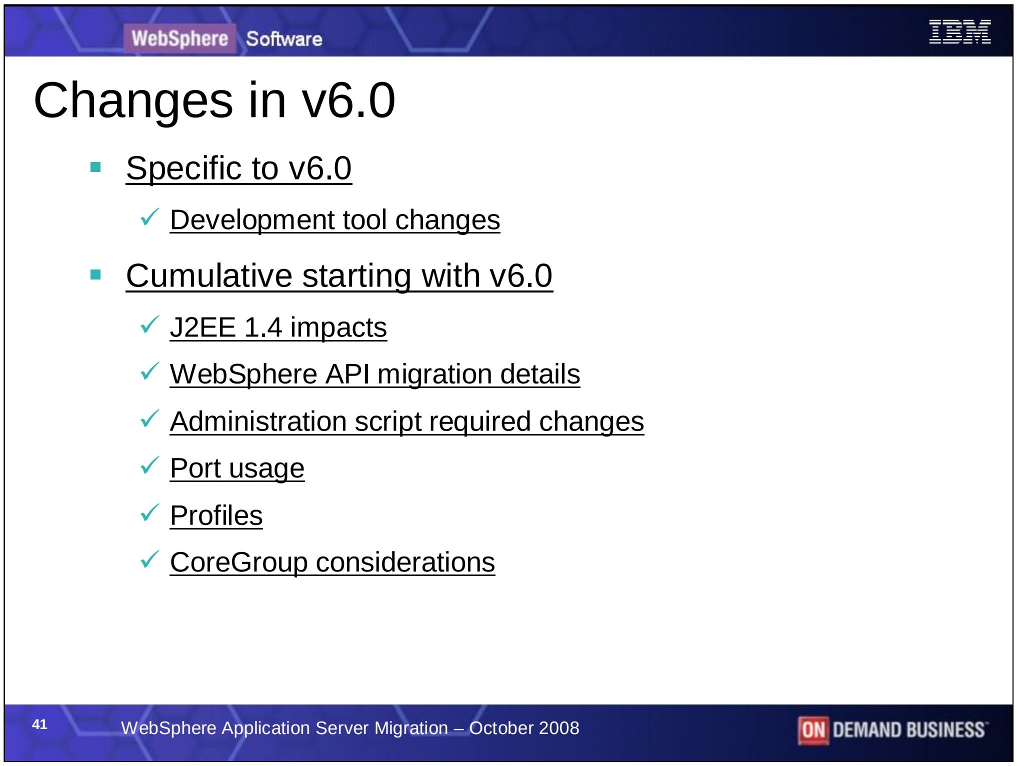
- There are two kinds of impacts documented here
  - “Specific” – impacts only those migrating TO that version
  - “Cumulative” – impacts when migrating TO or PAST that version
- The information assumes a starting point of v5.1 or later
  - Additional information is provided in the appendices for earlier versions like v4.0 and v5.0

	v6.0	v6.0	v6.1	v6.1	v7.0	v7.0
	Specific	Cumulative	Specific	Cumulative	Specific	Cumulative
v5.1->v6.0	Yes	Yes				
v5.1->v6.1		Yes	Yes	Yes		
v5.1->v7.0		Yes		Yes	Yes	Yes
v6.0->v6.1			Yes	Yes		
v6.0->v7.0				Yes	Yes	Yes
v6.1->v7.0					Yes	Yes

This chart lists the considerations when moving between versions. There are two kinds of impacts. The impacts that are specific to a version are just that – they only impact those that move to that version. The cumulative impacts are those that are additive in natural if they are in the middle of the version upgrade.

For example, if we were migrating from v6.0 to v7.0 we would have to be concerned with the cumulative impacts of v6.1 plus all the migration impacts of moving to v7.0

The table list all combinations for migrating from v5.1 onward.

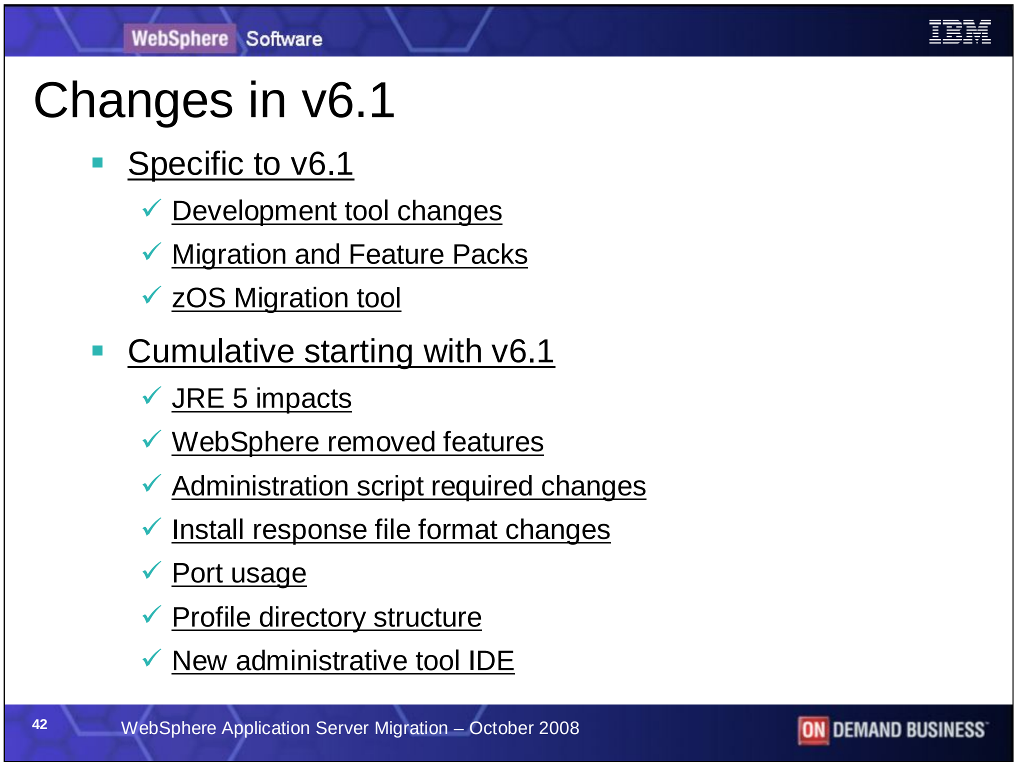


WebSphere Software

Changes in v6.0

- Specific to v6.0
  - ✓ Development tool changes
- Cumulative starting with v6.0
  - ✓ J2EE 1.4 impacts
  - ✓ WebSphere API migration details
  - ✓ Administration script required changes
  - ✓ Port usage
  - ✓ Profiles
  - ✓ CoreGroup considerations

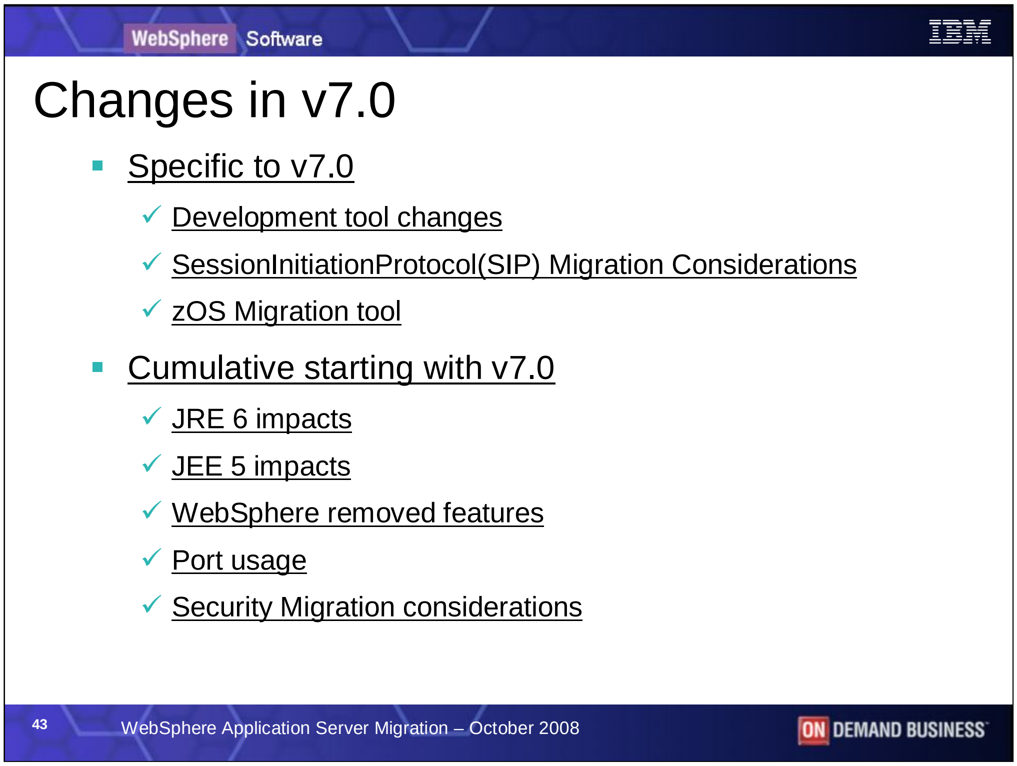
This chart lists the considerations when moving to or past version 6.0. Note that there are both impacts that are specific to a version and the cumulative impacts as described earlier in the deck.



Changes in v6.1

- Specific to v6.1
  - ✓ Development tool changes
  - ✓ Migration and Feature Packs
  - ✓ zOS Migration tool
- Cumulative starting with v6.1
  - ✓ JRE 5 impacts
  - ✓ WebSphere removed features
  - ✓ Administration script required changes
  - ✓ Install response file format changes
  - ✓ Port usage
  - ✓ Profile directory structure
  - ✓ New administrative tool IDE

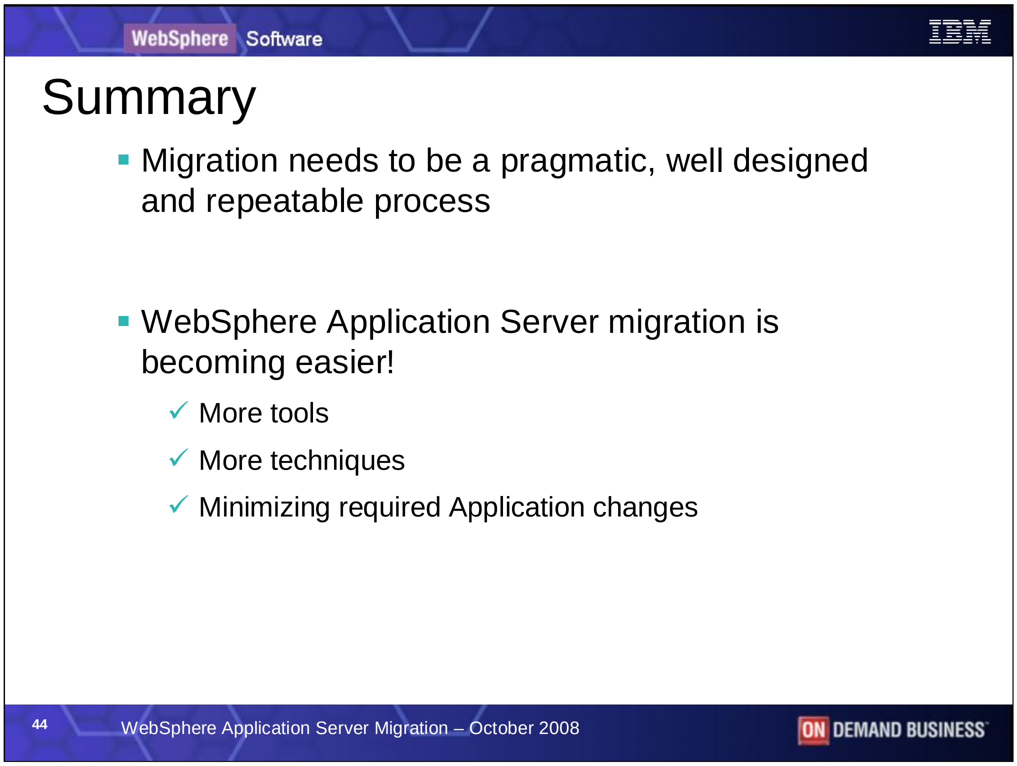
This chart lists the considerations when moving to version 6.1. Note that there are both impacts that are specific to a version and the cumulative impacts as described earlier in the deck.



Changes in v7.0

- Specific to v7.0
  - ✓ Development tool changes
  - ✓ SessionInitiationProtocol(SIP) Migration Considerations
  - ✓ zOS Migration tool
- Cumulative starting with v7.0
  - ✓ JRE 6 impacts
  - ✓ JEE 5 impacts
  - ✓ WebSphere removed features
  - ✓ Port usage
  - ✓ Security Migration considerations

This chart lists the considerations when moving to version 7.0. Note that there are both impacts that are specific to a version and the cumulative impacts as described earlier in the deck.



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

In summary, the job of migration of WebSphere Application Server is becoming easier. However, the task cannot be trivialized and careful planning and execution is highly recommended.

WebSphere Software

Questions?

October 2008

ON DEMAND BUSINESS

WebSphere Software

Thank You

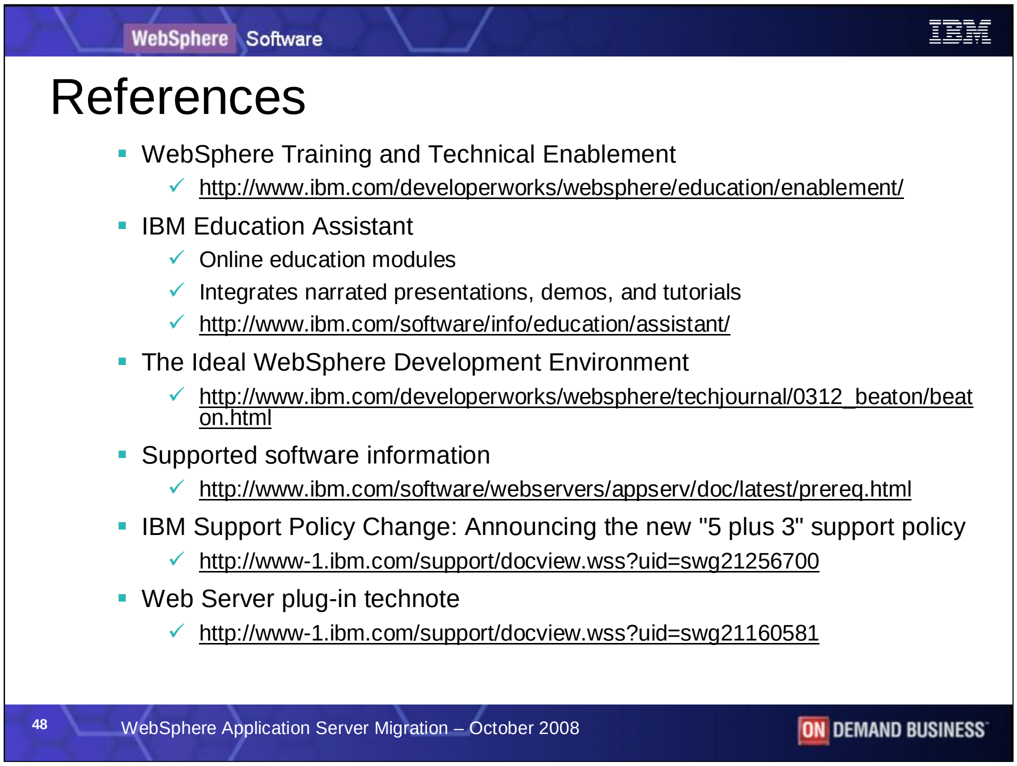
October 2008

ON DEMAND BUSINESS

WebSphere Software

References

October 2008

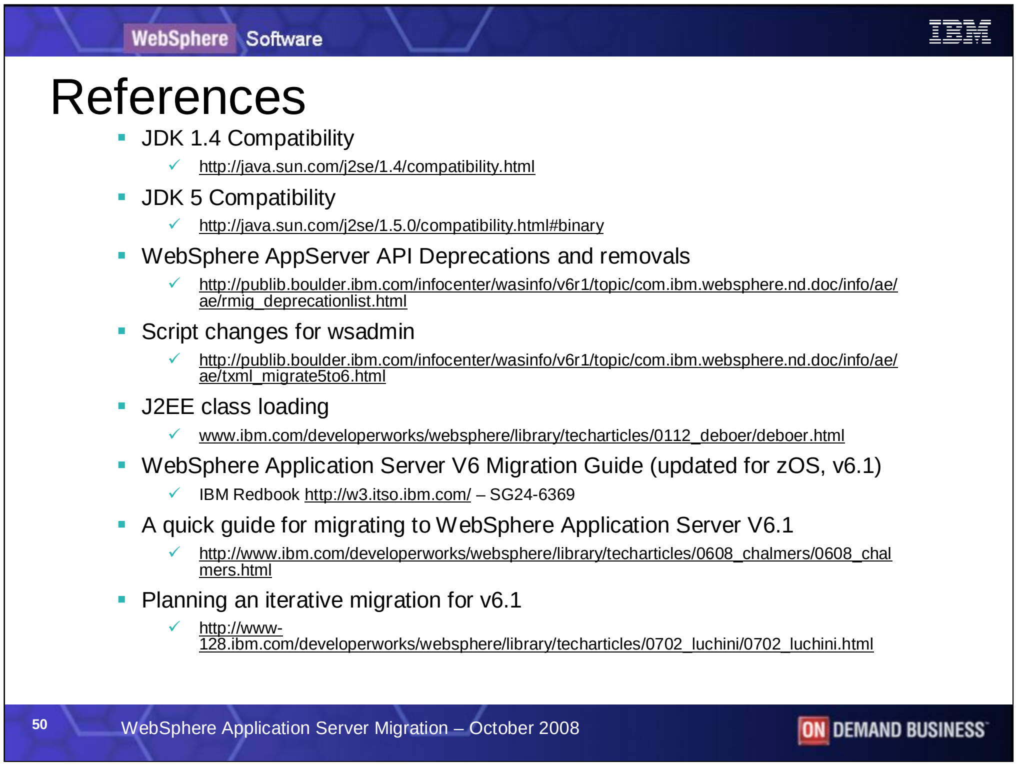


References

- WebSphere Training and Technical Enablement
  - ✓ http://www.ibm.com/developerworks/websphere/education/enablement/
- IBM Education Assistant
  - ✓ Online education modules
  - ✓ Integrates narrated presentations, demos, and tutorials
  - ✓ http://www.ibm.com/software/info/education/assistant/
- The Ideal WebSphere Development Environment
  - ✓ http://www.ibm.com/developerworks/websphere/techjournal/0312_beaton/beaton.html
- Supported software information
  - ✓ http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html
- IBM Support Policy Change: Announcing the new "5 plus 3" support policy
  - ✓ http://www-1.ibm.com/support/docview.wss?uid=swg21256700
- Web Server plug-in technote
  - ✓ http://www-1.ibm.com/support/docview.wss?uid=swg21160581

References

- Systems Management for Version 6
  - ✓ Overview of enhancements: http://www.ibm.com/developerworks/websphere/techjournal/0501_williamson/0501_williamson.html
  - ✓ Incremental cell upgrade: http://www.ibm.com/developerworks/websphere/techjournal/0502_cheng/0502_cheng.html
  - ✓ Profiles explained: http://www.ibm.com/developerworks/websphere/techjournal/0505_wang/0505_wang.html
  - ✓ New AdminTask support: http://www.ibm.com/developerworks/websphere/techjournal/0509_wang/0509_wang.html
  - ✓ Enhanced Application Deployment: http://www.ibm.com/developerworks/websphere/techjournal/0510_apte/0510_apte.html
- zOS references
  - ✓ Migrating Applications from WebSphere for zOS V4 and V3.5 to V5
    - IBM Redbook http://w3.itso.ibm.com/ – SG24-7044
  - ✓ Migrating an ND configuration from V5.x to V6.0
    - http://www.ibm.com/support/techdocs – Search White Papers for WP100559
  - ✓ Migrating to Version 6.1
    - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100771
  - ✓ Heterogeneous cell support for zOS
    - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100644
  - ✓ JDBC migration utility for Legacy to Universal DB2 drivers
    - http://www-1.ibm.com/support/docview.wss?uid=swg27007826

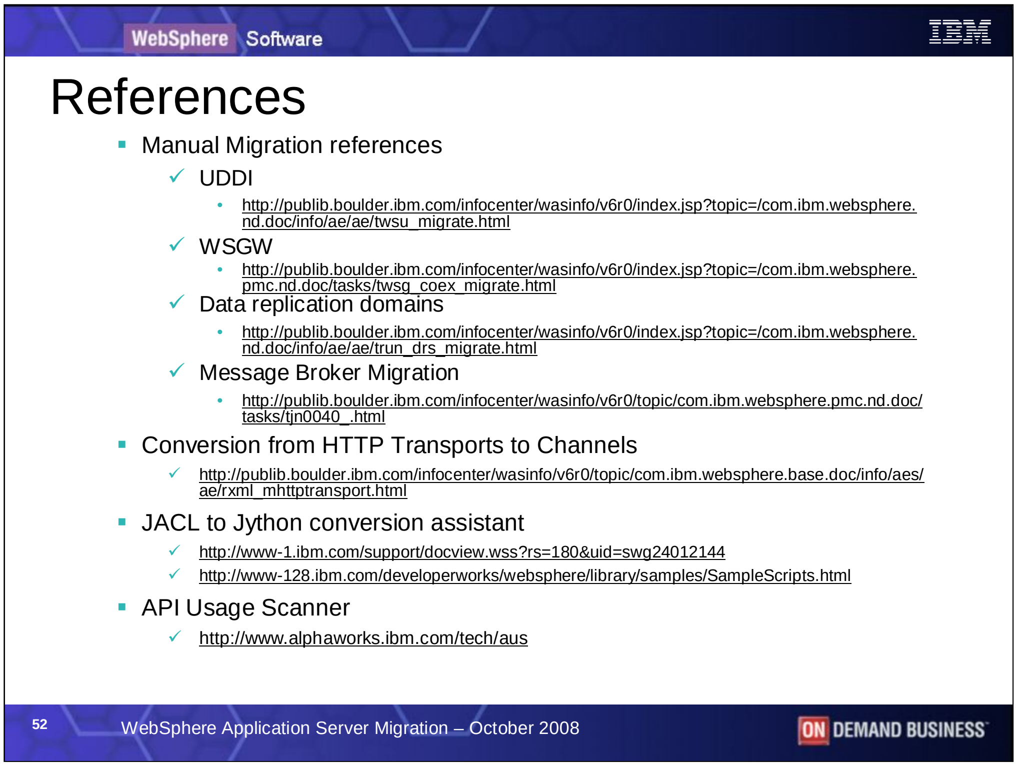


References

- JDK 1.4 Compatibility
  - ✓ http://java.sun.com/j2se/1.4/compatibility.html
- JDK 5 Compatibility
  - ✓ http://java.sun.com/j2se/1.5.0/compatibility.html#binary
- WebSphere AppServer API Deprecations and removals
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_deprecationlist.html
- Script changes for wsadmin
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate5to6.html
- J2EE class loading
  - ✓ www.ibm.com/developerworks/websphere/library/techarticles/0112_deboer/deboer.html
- WebSphere Application Server V6 Migration Guide (updated for zOS, v6.1)
  - ✓ IBM Redbook http://w3.itso.ibm.com/ – SG24-6369
- A quick guide for migrating to WebSphere Application Server V6.1
  - ✓ http://www.ibm.com/developerworks/websphere/library/techarticles/0608_chalmers/0608_chalmers.html
- Planning an iterative migration for v6.1
  - ✓ http://www-128.ibm.com/developerworks/websphere/library/techarticles/0702_luchini/0702_luchini.html

References

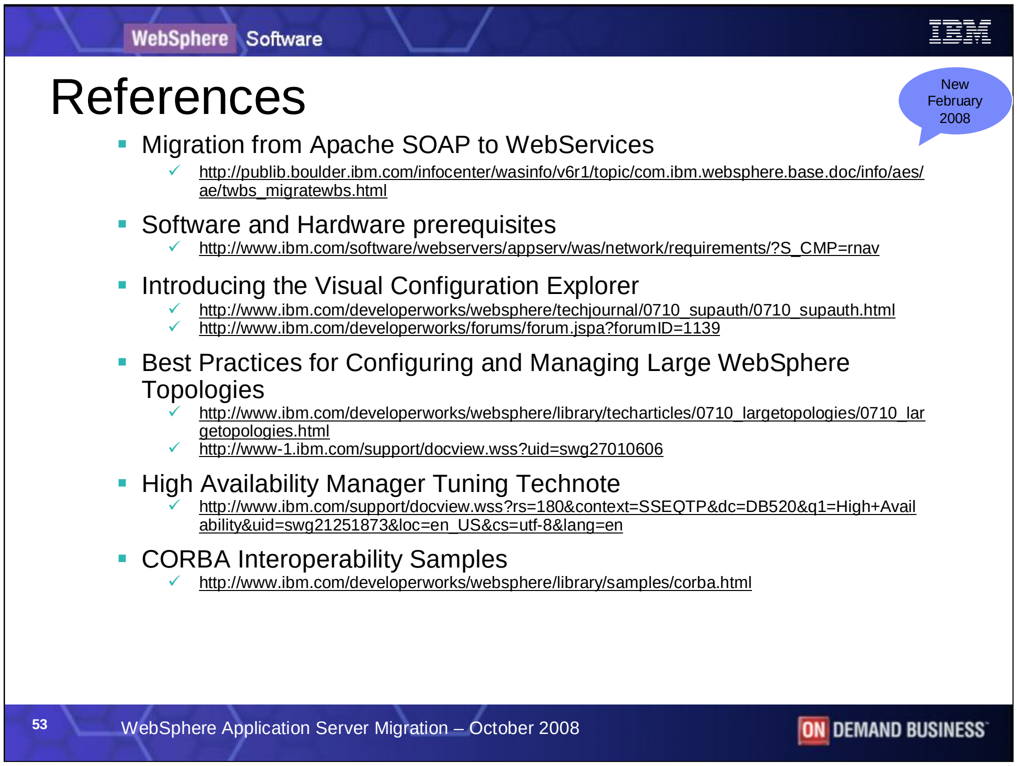
- CCF-to-J2C Architecture Migration
  - ✓ IBM Redbook http://w3.itso.ibm.com/– SG24-6362
  - ✓ developerWorks 5 part series http://www.ibm.com/developerworks/websphere/library/techarticles/0403_searle/0403_searle1.html
- JMS Listener to Message Driven Bean migration
  - ✓ http://publib.boulder.ibm.com/infocenter/ws51help/topic/com.ibm.websphere.base.doc/info/aes/ae/tmb_migmb.html.
- Rational Application Developer Performance Tips
  - ✓ http://www.ibm.com/developerworks/rational/library/05/517_radtip/index.html
  - ✓ http://www.redbooks.ibm.com/abstracts/sg246449.html
- Sample scripts for WebSphere Application Server
  - ✓ http://www.ibm.com/developerworks/websphere/library/samples/SampleScripts.html
  - ✓ Includes updates for v6.1 Security
- Security changes for v6.1
  - ✓ http://www-128.ibm.com/developerworks/websphere/techjournal/0612_birk/0612_birk.html
- XML reference
  - ✓ http://www.w3.org/TR/REC-xml/#sec-guessing



References

- Manual Migration references
  - ✓ UDDI
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/twsu_migrate.html
  - ✓ WSGW
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.pmc.nd.doc/tasks/twsg_coex_migrate.html
  - ✓ Data replication domains
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_drs_migrate.html
  - ✓ Message Broker Migration
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.pmc.nd.doc/tasks/tjn0040_.html
- Conversion from HTTP Transports to Channels
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/rxml_mhttptransport.html
- JACL to Jython conversion assistant
  - ✓ http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012144
  - ✓ http://www-128.ibm.com/developerworks/websphere/library/samples/SampleScripts.html
- API Usage Scanner
  - ✓ http://www.alphaworks.ibm.com/tech/aus

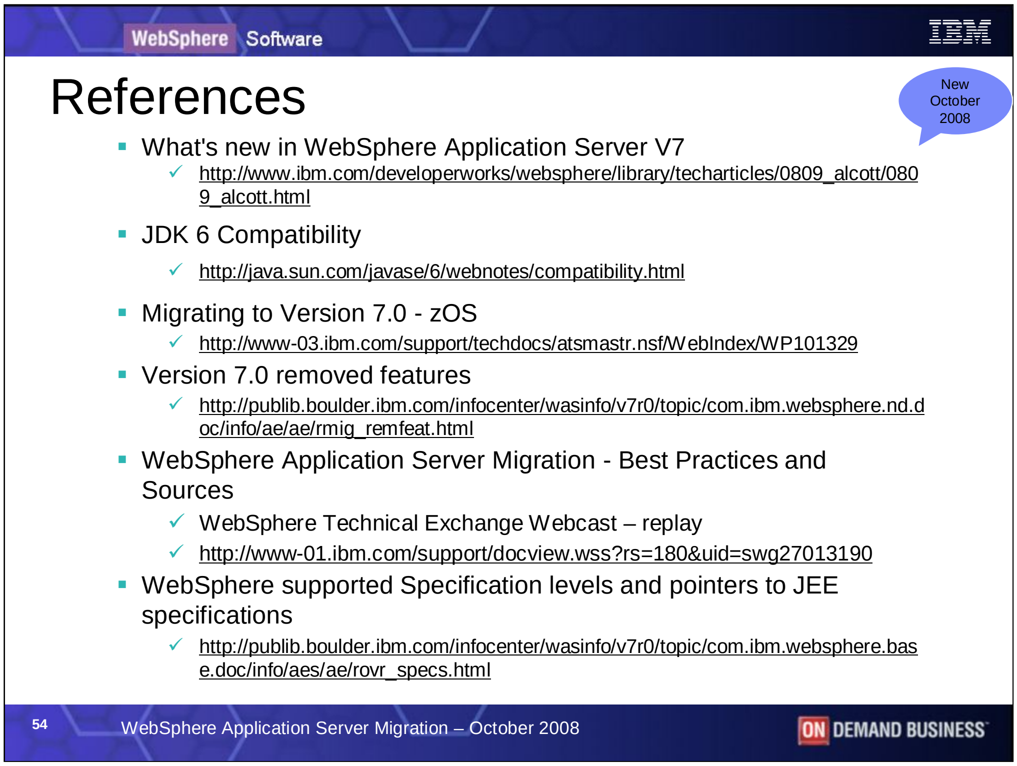
API Usage Scanner takes as input a set of compiled Java code, which could consist of individual class files or class files contained within nested archives such as JAR, WAR, and EAR files. These class files, together with scanning rules, are used by AUS to create a set of reports showing where references to the APIs are found and how those APIs are used.



References

New
February
2008

- Migration from Apache SOAP to WebServices
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.base.doc/info/aes/ae/twbs_migratewbs.html
- Software and Hardware prerequisites
  - ✓ http://www.ibm.com/software/webservers/appserv/was/network/requirements/?S_CMP=rnav
- Introducing the Visual Configuration Explorer
  - ✓ http://www.ibm.com/developerworks/websphere/techjournal/0710_supauth/0710_supauth.html
  - ✓ http://www.ibm.com/developerworks/forums/forum.jspa?forumID=1139
- Best Practices for Configuring and Managing Large WebSphere Topologies
  - ✓ http://www.ibm.com/developerworks/websphere/library/techarticles/0710_largetopologies/0710_largetopologies.html
  - ✓ http://www-1.ibm.com/support/docview.wss?uid=swg27010606
- High Availability Manager Tuning Technote
  - ✓ http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=DB520&q1=High+Availability&uid=swg21251873&loc=en_US&cs=utf-8&lang=en
- CORBA Interoperability Samples
  - ✓ http://www.ibm.com/developerworks/websphere/library/samples/corba.html



References

New October 2008

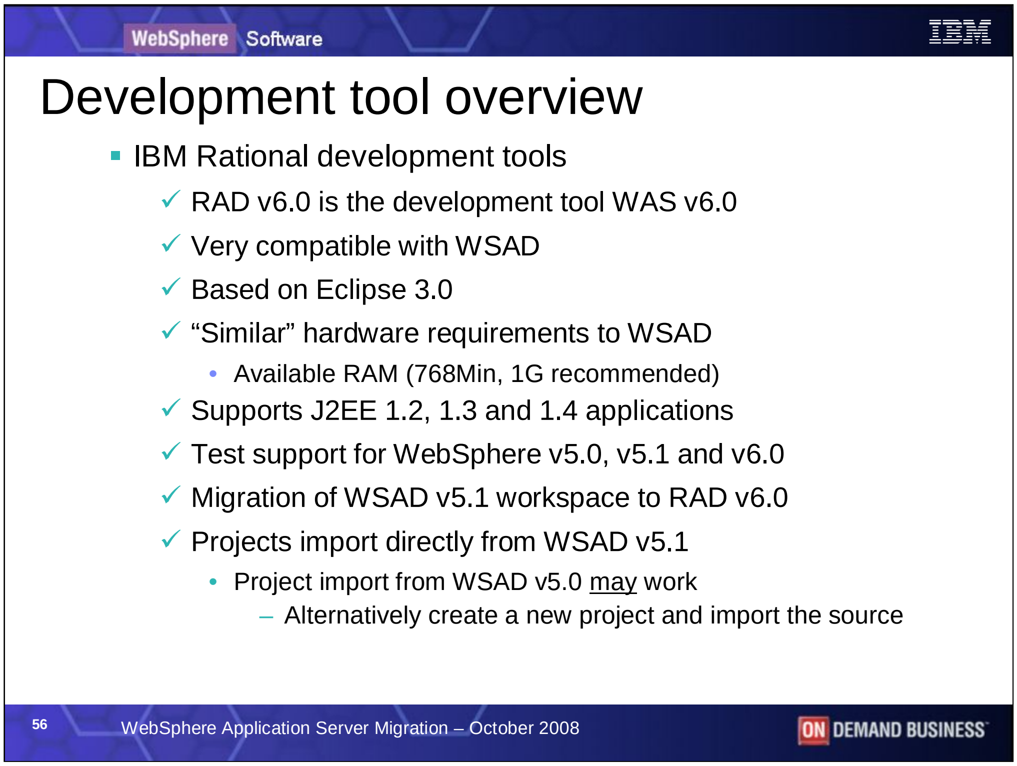
- What's new in WebSphere Application Server V7
  - ✓ http://www.ibm.com/developerworks/websphere/library/techarticles/0809_alcott/0809_alcott.html
- JDK 6 Compatibility
  - ✓ http://java.sun.com/javase/6/webnotes/compatibility.html
- Migrating to Version 7.0 - zOS
  - ✓ http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101329
- Version 7.0 removed features
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_remfeat.html
- WebSphere Application Server Migration - Best Practices and Sources
  - ✓ WebSphere Technical Exchange Webcast – replay
  - ✓ http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27013190
- WebSphere supported Specification levels and pointers to JEE specifications
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/aes/ae/rovr_specs.html

These references are for v7.0, more will be added as they are published

WebSphere Software

v6.0 specific changes

October 2008



WebSphere Software

Development tool overview

- IBM Rational development tools
  - ✓ RAD v6.0 is the development tool 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

56 WebSphere Application Server Migration – October 2008

RAD v6.0 is the development tool that is designed to work with WebSphere Application Server v6.0. It has full test support for WebSphere Application Server v5.0, v5.1 and v6.0 which means that you can use the built in servers or hook into an existing supported version of the server.

Other slides go into more detail with respect to the compatibility of WSAD v5.1 workspaces and RAD v6.0, but to sum those slides up, there is good support.

One thing you may notice just going down the specification for RAD v6 is the increased memory, processor and storage needs over WSAD v5.1 Nothing can be done about those increased needs, but there is a good article that can help you tune RAD v6 to get the most out of your experience.

Article location: http://www-128.ibm.com/developerworks/rational/library/05/517_radtip/index.html

WebSphere Software

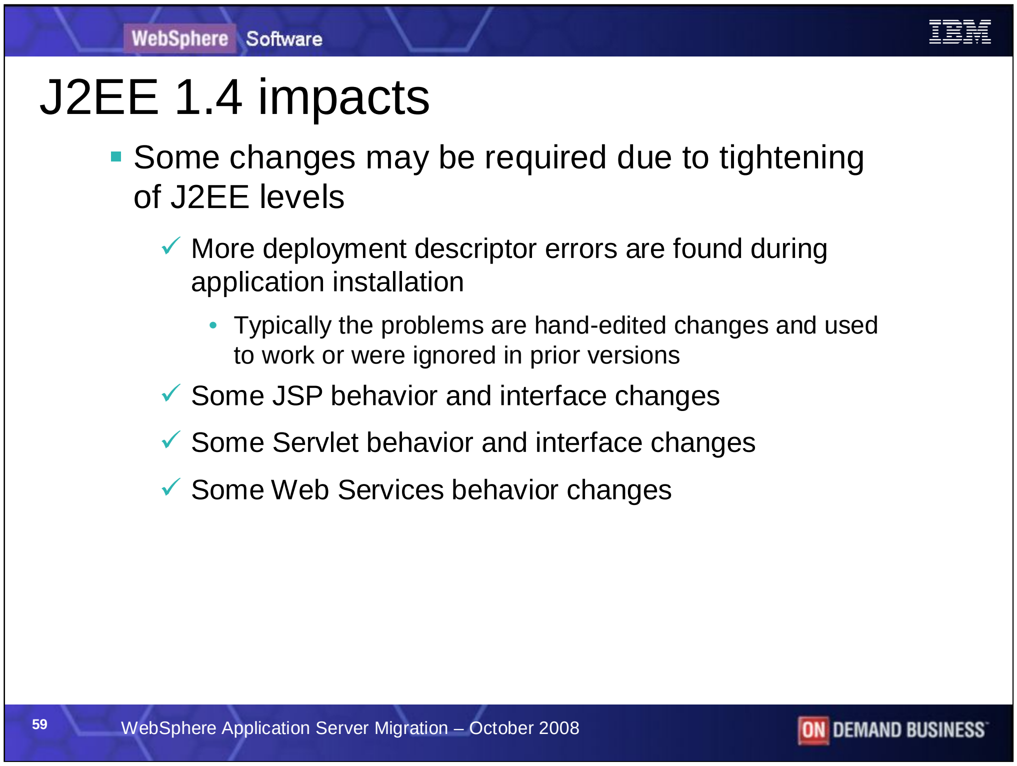
v6.0 cumulative changes

October 2008

ON DEMAND BUSINESS

v6.0 cumulative

J2EE 1.4 migration details


WebSphere Software
IBM

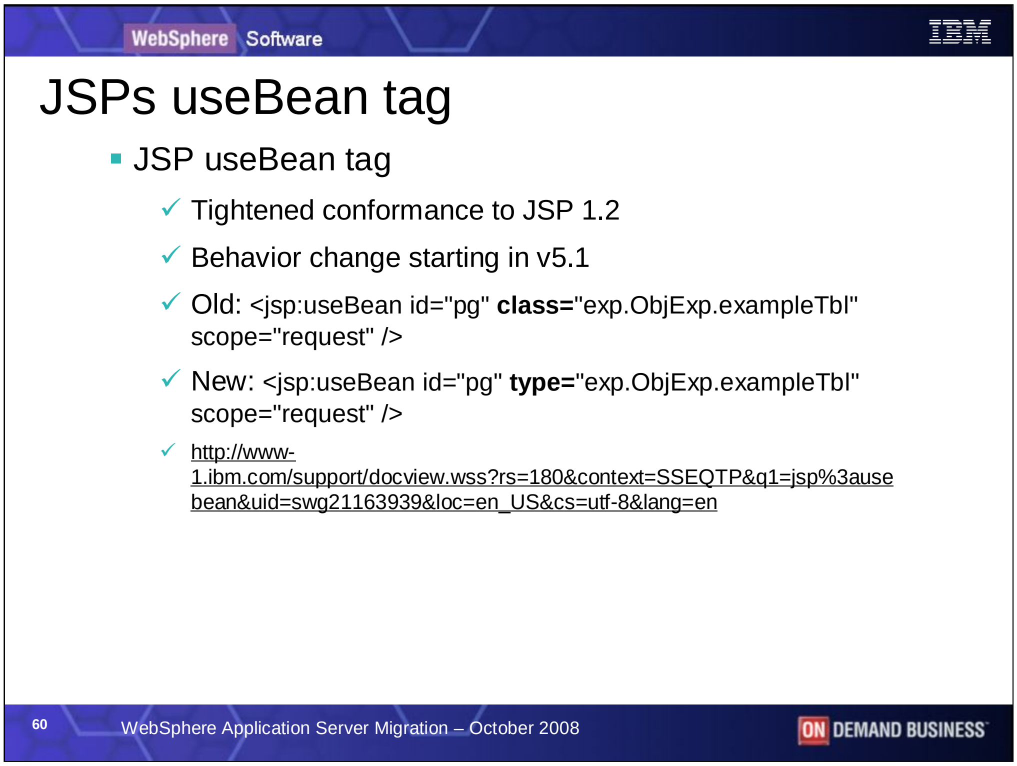
J2EE 1.4 impacts

- 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

Each version of the J2EE specification that is released contains many additions and refinements to the existing specification and the release of J2EE 1.4 is no different.

J2EE is forward compatible, so if your application worked in v1.2 or v1.3 it should work in v1.4 with a few exceptions. The authors of the J2EE specification have made every effort to be clear on all points, but due to the size and complexity of the specification things slip through and thus we are left with ambiguities. Each Application Server will interpret these ambiguities differently. J2EE v1.4 resolved many of these ambiguities and thus your applications should be checked for consistent behavior between your current version of WebSphere Application Server and your target version of WebSphere Application Server.

The best source of information for all of the known changes or refinements to the current version of the specification is in the “compatibility” section of the individual specifications.

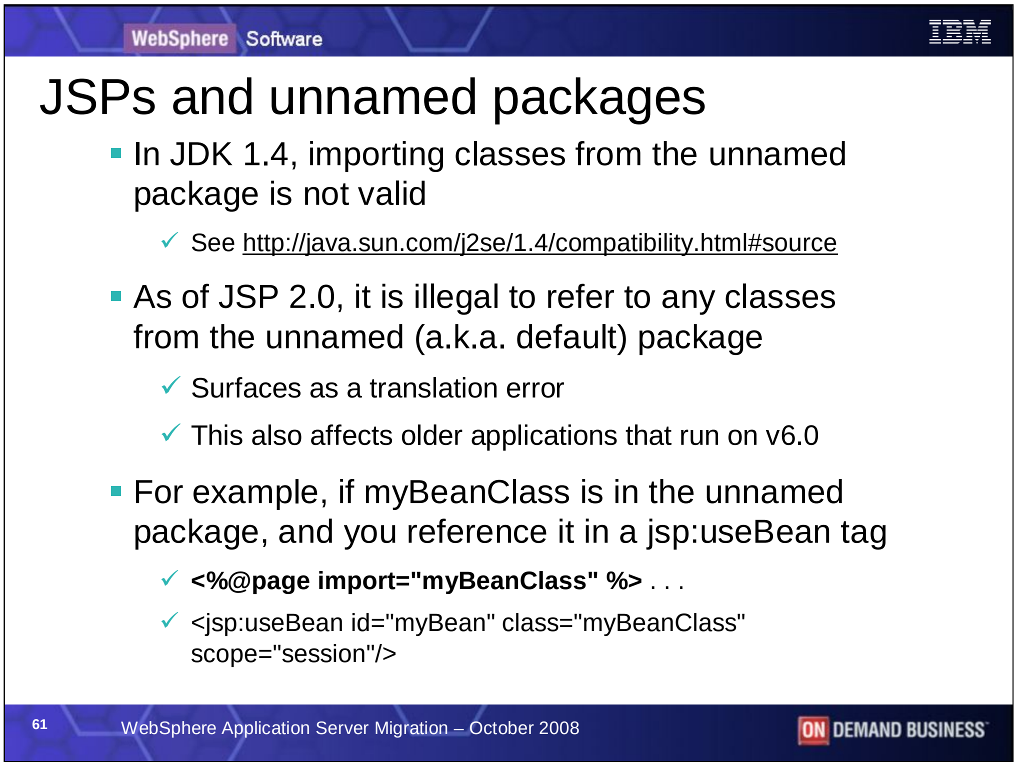


JSPs useBean tag

- 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" />
  - ✓ http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q1=jsp%3ausebean&uid=swg21163939&loc=en_US&cs=utf-8&lang=en

WebSphere Application Server has tightened its conformance with the JSP 2.0 specification with regards to the usage of <jsp:useBean>. Prior to v5.1 the “class” attribute could have been used in the place of the “type” attribute, but starting in v5.1 this usage scenario has been tightened and will now throw a “javax.servlet.ServletException: Unable to compile class for JSP” if used incorrectly. See the included link for a complete description of the change.

The main difference between the “class” attribute and the “type” attribute is that the “class” attribute is a concrete class with a no-argument(or default) constructor that will be instantiated, as opposed to the “type” attribute which specifies an interface or class(can be super class) that will be used to reference an Object that has already been created elsewhere in your JSP.



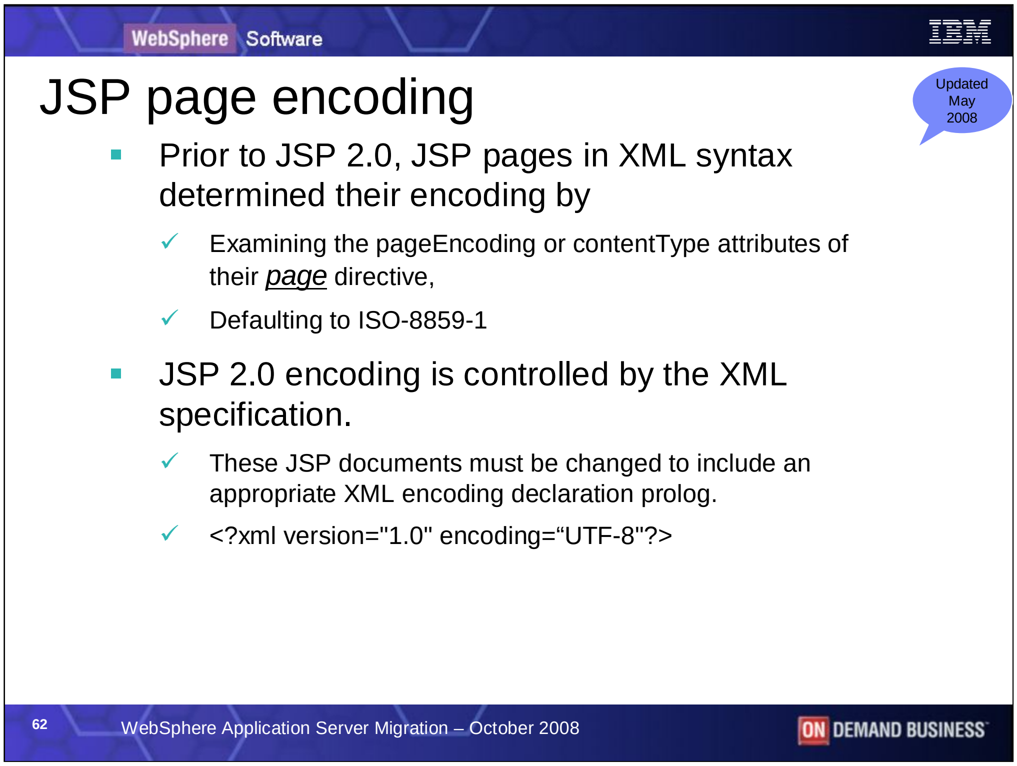
JSPs and unnamed packages

- In JDK 1.4, importing classes from the unnamed package is not valid
  - ✓ See http://java.sun.com/j2se/1.4/compatibility.html#source
- 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"/>

JDK v1.4 tightened its enforcement of the Java Language Specification. In JDK versions prior to v1.4, code that imported classes from an unnamed package would compile even though it did not comply with the Java Language Specification, but in JDK v1.4 this same code will fail to compile.

JSP v2.0 has the same inherent change in behavior as described in the preceding section due to it’s requirement of JDK v1.4. If your JSP does not comply with this new tightening of the specification you will receive errors during the translation step(This is the step where the JSP gets converted into Servlet code).

The link supplied in this slide contains details relative to this issue and other JDK v1.4 compatibility issues.



JSP page encoding

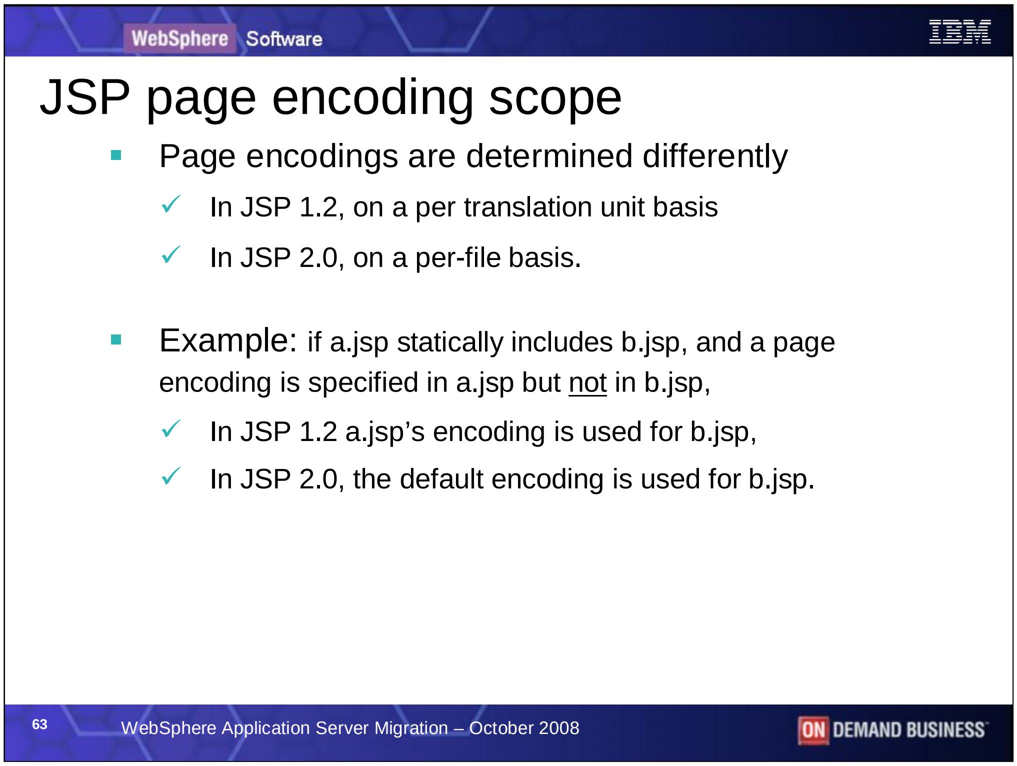
Updated May 2008

- 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 v1.2 contained ambiguities in the JSP encoding portion of the specification that have been fleshed out in JSP v2.0, thus possibly resulting in undesirable behavior.

Prior to JSP v2.0 page encoding was determined by inspecting the “pageEncoding” and “contentType” attributes and if neither was present it would default to ISO-8859-1. Starting in JSP v2.0 page encoding is determined by section 4.3.3 and appendix F.1 of the XML specification and the “pageEncoding” attribute is only checked for consistency, thus if your JSPs relied on the “pageEncoding” attribute they will no longer get encoded properly.

See http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/rweb_jsp_migration.html for more information

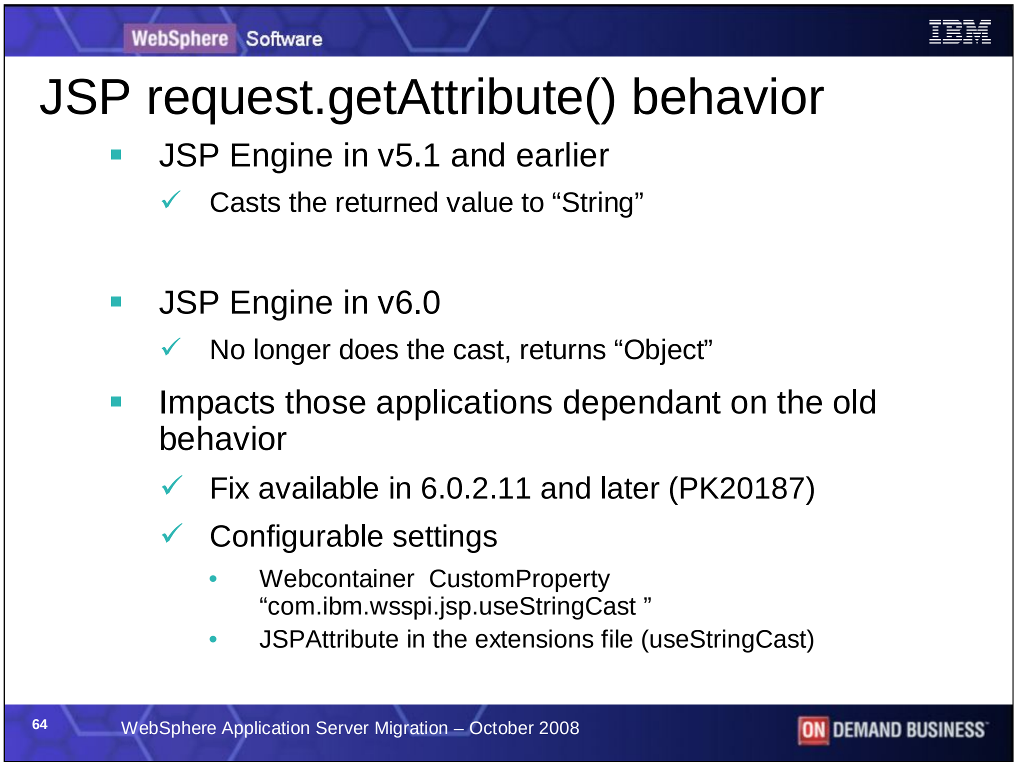


JSP page encoding scope

- 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 <u>not</u> 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 v1.2 contained ambiguities in the JSP encoding portion of the specification that have been fleshed out in JSP v2.0, thus possibly resulting in undesirable behavior.

In JSP v1.2, page encodings are determined on translation unit basis whereas in JSP v2.0, page encodings are determined on the basis of each file. Therefore, if the a.jsp file statically includes the b.jsp file, and a page encoding is specified in the a.jsp file but not in the b.jsp file, in JSP v1.2 the encoding for the a.jsp file is used for the b.jsp file, but in JSP v2.0, the default encoding is used for the b.jsp file.

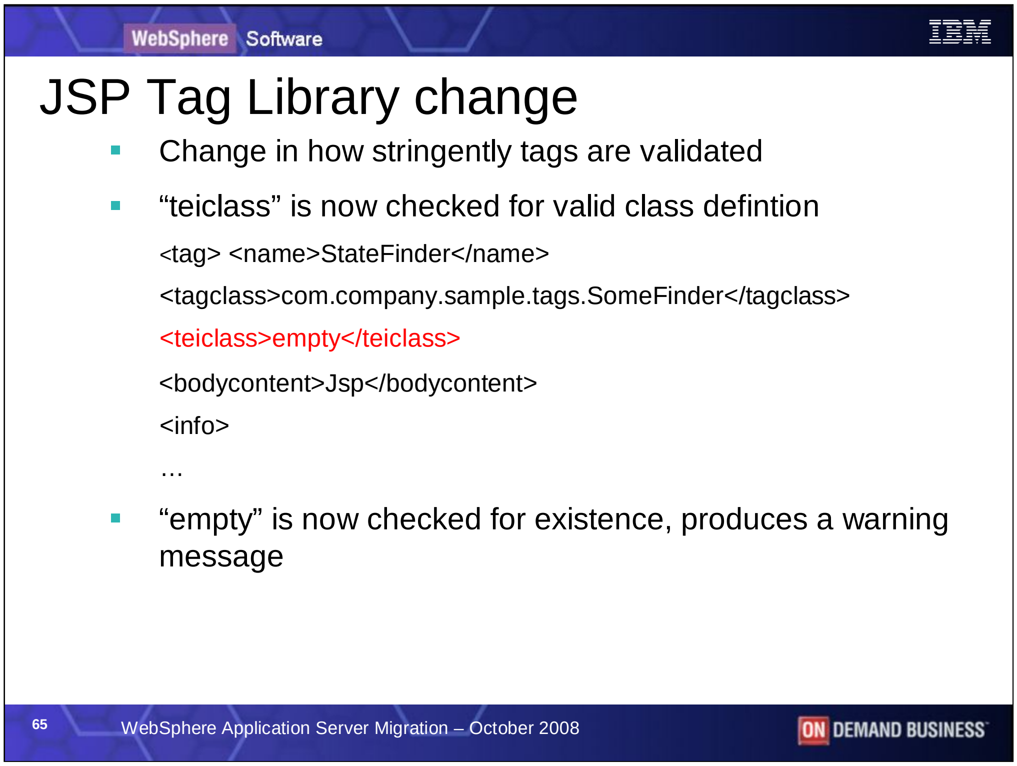


JSP request.getAttribute() behavior

- 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)

The JSP engine included with v5.1 of WebSphere Application Server implicitly casts the return value of “request.getAttribute()” during the translation step to a String. In v6.0 this behavior has been changed to return a generic Object, which may impact applications that are dependant on the old behavior. Needless to say this is an undesirable behavioral change and thus APAR PK20187 was released and is available in fixpack 6.0.2.11. This fix introduces a custom property that can be set in the WebContainer which will effect all applications or it can be set on a per application basis. This custom property when set will force the container to abide by the old behavior and add the cast operation back in during the translation step, thus avoiding the need to modify your applications.

See the following link for detailed instructions: http://www-1.ibm.com/support/docview.wss?uid=swg1PK20187



JSP Tag Library change

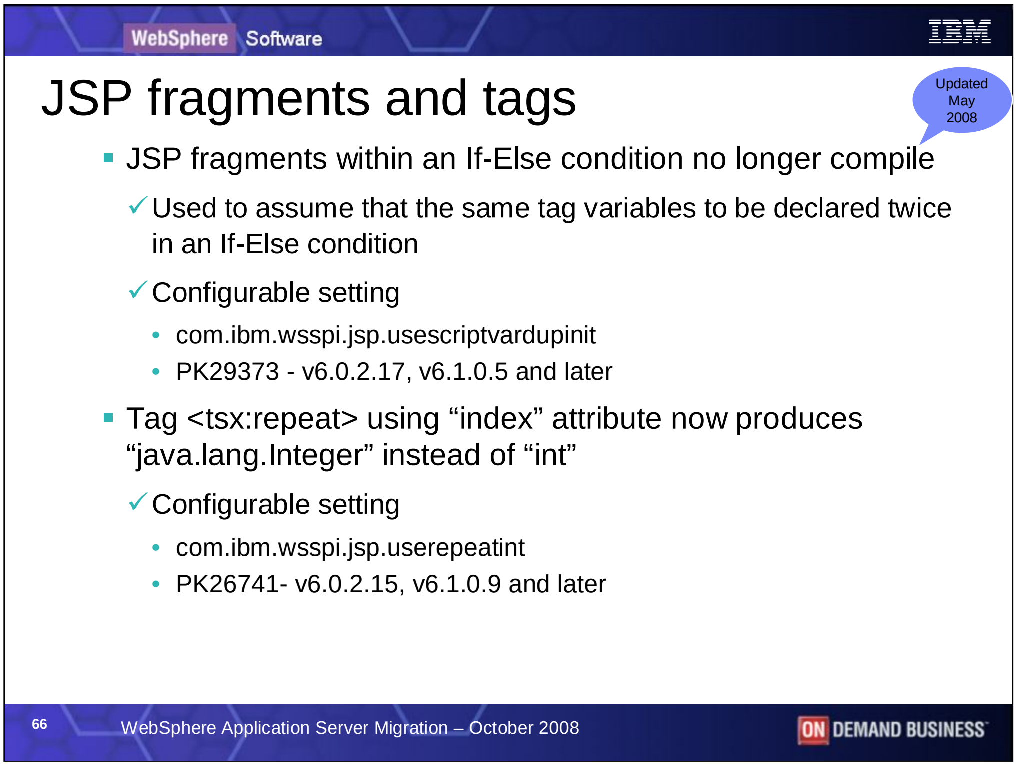
- Change in how stringently tags are validated
- “teiclass” is now checked for valid class defintion

```
<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

Starting in WebSphere Application Server v6 the <teiclass> tag if empty(or contained a reference to a class that could not be loaded) would cause an exception to be thrown to the logs. This is a change in behavior from versions prior to v6 and thus has caused confusion among customers. To resolve this confusion APAR PK27099 was released into fixpack 6.0.2.15. This APAR changed the behavior back to original behavior prior to v6.

See the following link for detailed instructions: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg1PK27099 for more information. Prior to In v6.0.2.15 and v6.1.0.3 the error would also cause an error message.



JSP fragments and tags

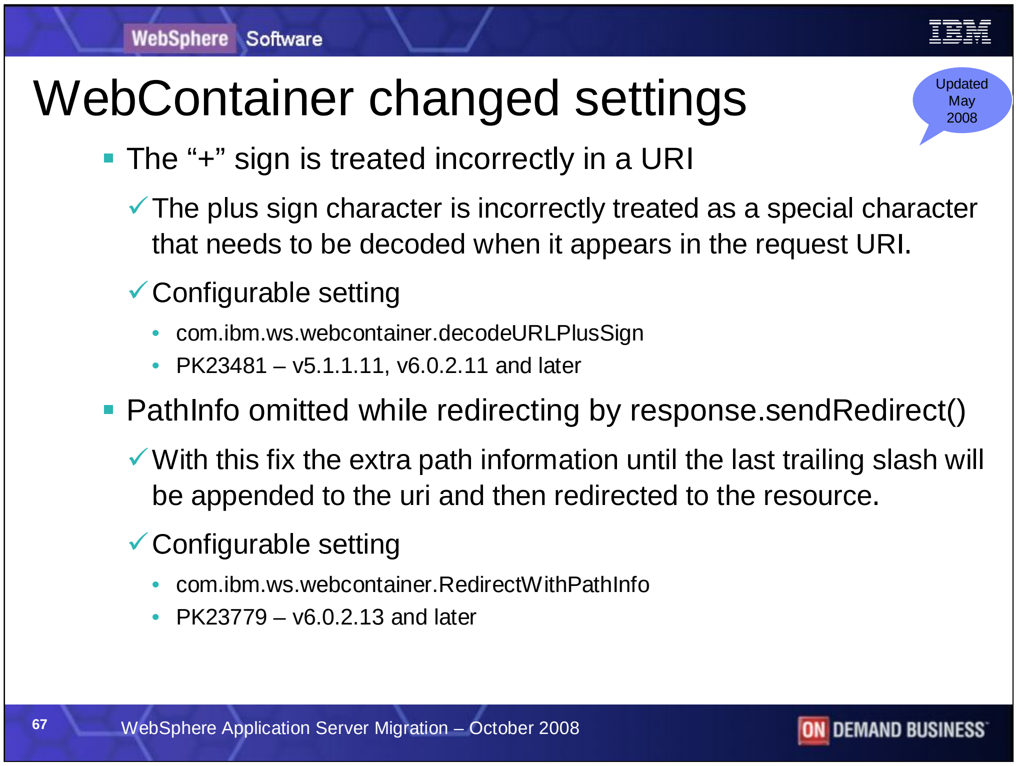
Updated May 2008

- 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

In WebSphere Application Server v5, the code generated for a jsp assumed the same tag variables to be declared twice in an If-Else condition even if the variable had a 'page' scope. This behavior has been changed in v6. To avoid forcing application changes, APAR PK29373 was introduced in fixpack v6.0.2.17/v6.1.0.5. This APAR introduced a property “usescriptvardupinit” which could be set at the application scope or WebContainer scope resulting in the behavior being reverted back to original behavior prior to v6.

Starting in WebSphere Application Server v6 the “index” attribute of the <tsx:repeat> tag takes a java.lang.Integer Object as an argument. This is a change in behavior from versions prior to v6 in which it took a primitive integer as an argument. To resolve this change in behavior and to avoid forcing application changes, APAR PK26741 was introduced in fixpack v6.0.2.15. This APAR introduced a property “useRepeatInt” which could be set at the application scope or WebContainer scope resulting in the behavior being reverted back to original behavior prior to v6.

See the following link for detailed instructions: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg1PK26741



WebContainer changed settings

Updated May 2008

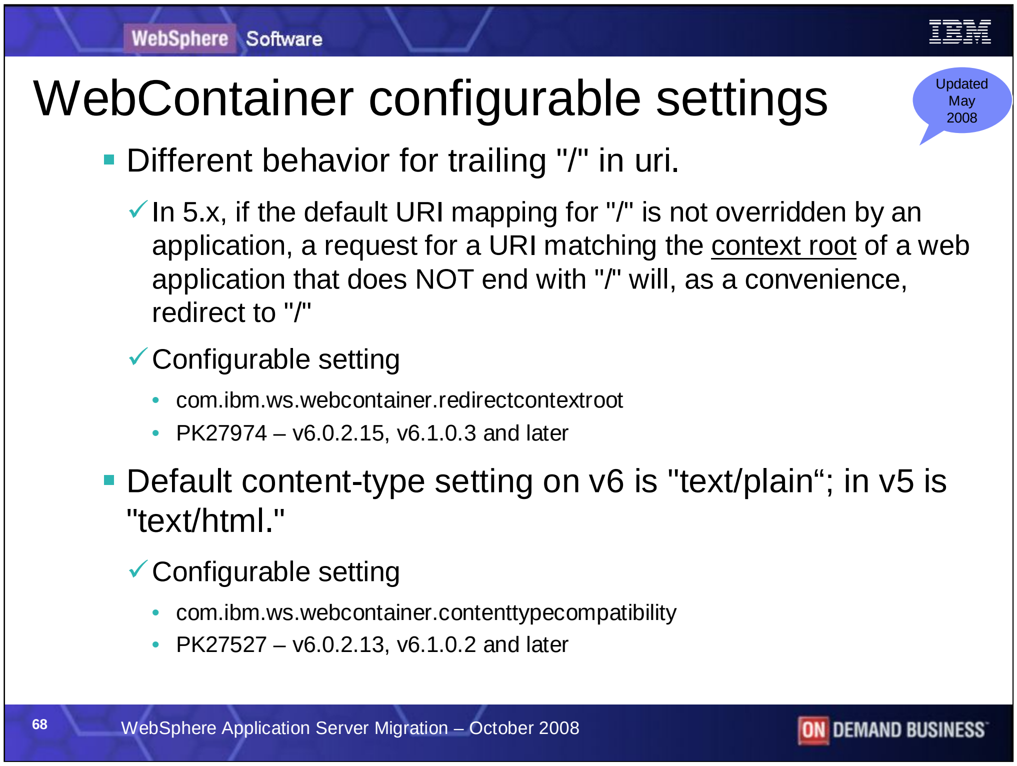
- The “+” sign is treated incorrectly in a URI
  - ✓ The plus sign character is incorrectly treated as a special character that needs to be decoded when it appears in the request URI.
  - ✓ Configurable setting
    - com.ibm.ws.webcontainer.decodeURLPlusSign
    - PK23481 – v5.1.1.11, v6.0.2.11 and later
- 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

“The “+” sign is treated incorrectly in a URI”

Application Server is treating the plus sign character as a special character that needs to be decoded when it appears in the request URI. This is not adhering to the URI specification which states that the plus sign is only a special character when it is part of the query string portion of the URL. By decoding the plus sign, the webcontainer is unable to service requests that contain the plus sign as part of the request URI (context path, servlet path or pathInfo). To resolve this problem, the webcontainer has been modified to no longer decode the plus sign when the webcontainer property com.ibm.ws.webcontainer.decodeURLPlusSign is set to false. The default behavior is true to prevent possible regression of function for customers currently relying on the current incorrect behavior of decoding the plus sign. See the following link for detailed instructions: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg1PK23481

response.sendRedirect(…) incorrectly removes the pathInfo(Example: path=“this/is/my/path/myFile.jsp” → “sendRedirect(…) does not pass along “this/is/my/path”) when it redirects. This problem has been found in v6.0 and APAR PK23779 has been released in fixpack v6.0.2.13 to resolve this issue. This fix introduces a new custom property that can be set, so that the response.sendRedirect(…) operates properly. The “incorrect” behavior remains the default behavior so as to not break existing applications dependent on this “incorrect” behavior.

See the following link for detailed instructions: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg1PK23779



WebSphere Software

WebContainer configurable settings

Updated May 2008

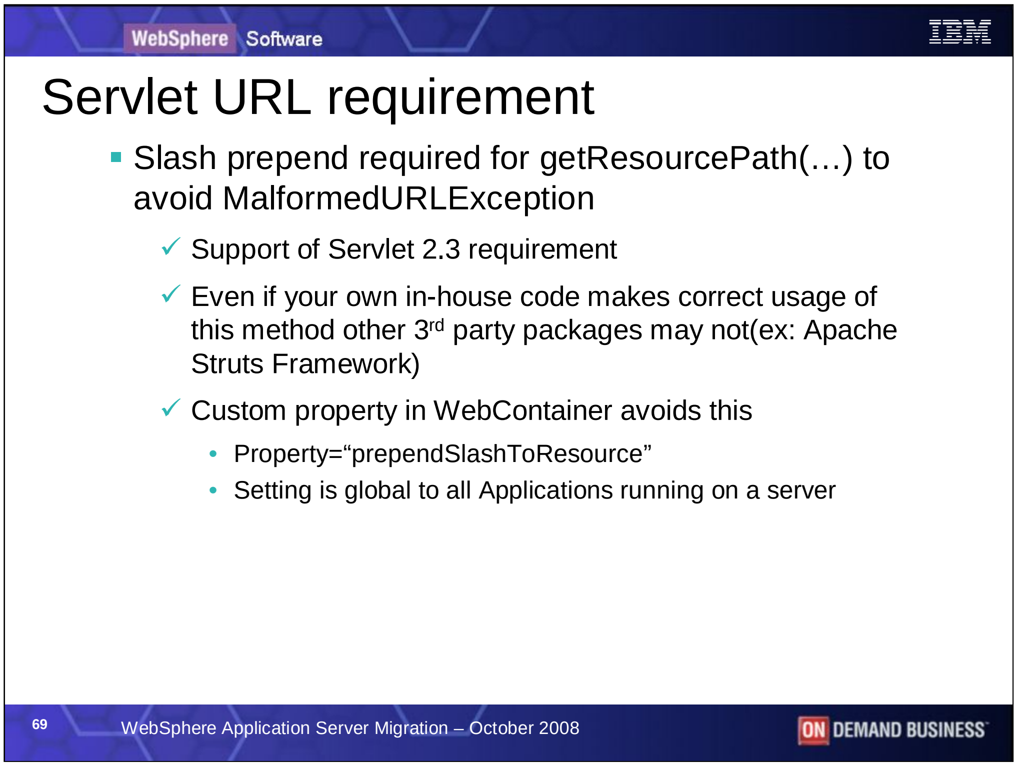
- Different behavior for trailing "/" in uri.
  - ✓ In 5.x, if the default URI mapping for "/" is not overridden by an application, a request for a URI matching the context root of a web application that does NOT end with "/" will, as a convenience, redirect to "/"
  - ✓ Configurable setting
    - com.ibm.ws.webcontainer.redirectcontextroot
    - 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

Starting in WebSphere Application Server v6.0 the automatic redirection of a missing trailing “/” in the context root does not occur. This is a behavior change from versions prior to v6.0 in which a trailing “/” was appended to the context root and then redirected(Example: URI: http://localhost:9080/webapp -> redirectsTo -> http://localhost:9080/webapp/). To resolve this change in behavior and to avoid forcing application changes, APAR PK27974 was introduced in fixpack v6.0.2.15. This APAR introduced a property which could be set at the WebContainer scope resulting in the behavior being reverted back to original behavior prior to v6.0.

See the following link for detailed instructions: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg1PK27974

Servlet v2.4 specification requires the “content-type” defaults to “text/plain” in WebSphere Application Server v6.0. This is a change in behavior from v5 in which “content-type” defaults to “text/html”. To resolve this change in behavior and to avoid forcing application changes, APAR PK27527 was introduced in fixpack v6.0.2.13. This APAR introduced a property which could be set at the WebContainer scope resulting in the behavior being reverted back to original behavior prior to v6.0

See the following link for detailed instructions: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg1PK27527



Servlet URL requirement

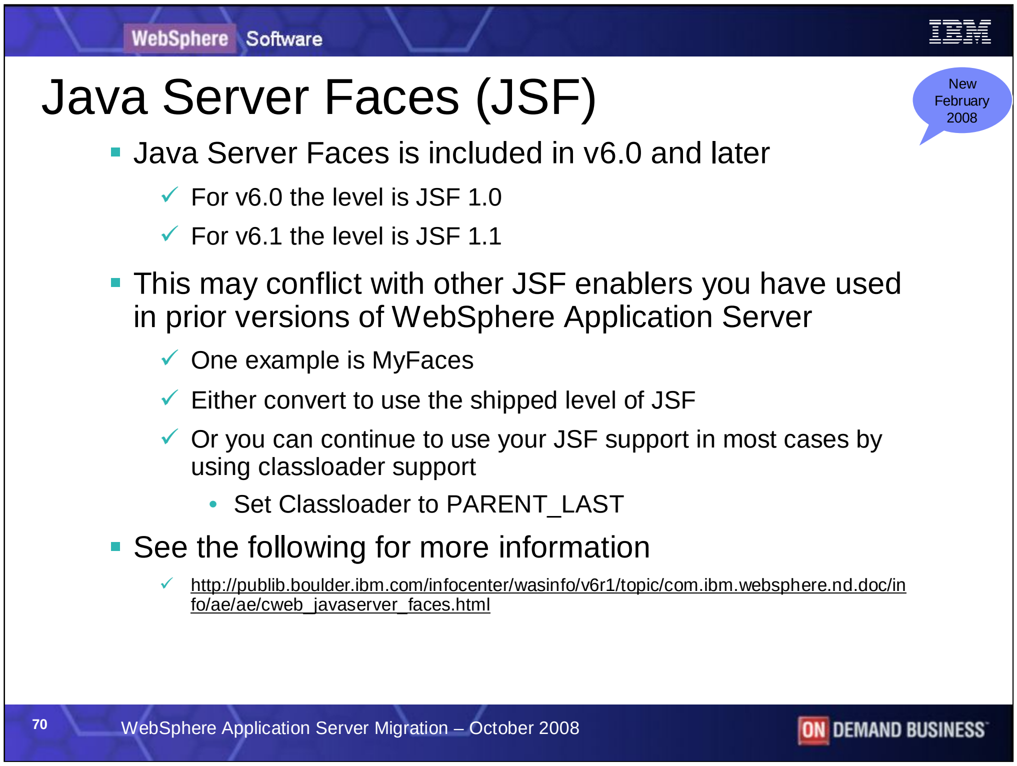
- Slash prepend required for getResourcePath(…) to avoid MalformedURLException
  - ✓ Support of Servlet 2.3 requirement
  - ✓ Even if your own in-house code makes correct usage of this method other 3rd party packages may not(ex: Apache Struts Framework)
  - ✓ Custom property in WebContainer avoids this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

getResourcePath(…) and getResourceAsStream(…) throw MalformedULRException when called from within a ServletContext with no leading “/”

Due to a change in the v2.4 Servlet specification WebSphere Application Server v6.0 requires that you have a leading “/” for methods “ServletContext.getResourcePath(…)” and “ServletContext.getResourceAsStream(…)”. If you do not include a leading “/” a MalformedURLException will result. A property was introduced at the WebContainer scope that will revert the behavior back to versions prior to v6.0. This property MUST be set in order to get the old behavior, it does not revert to the old behavior by default like quite a few other behavior changes discussed thus far.

Even if you do not fall prey to this problem in your own in-house code you may be using 3rd party packages that may. One known and popular package that has this issue is the Apache Struts Framework.

See the following link for additional information: http://www-1.ibm.com/support/docview.wss?uid=swg21190234

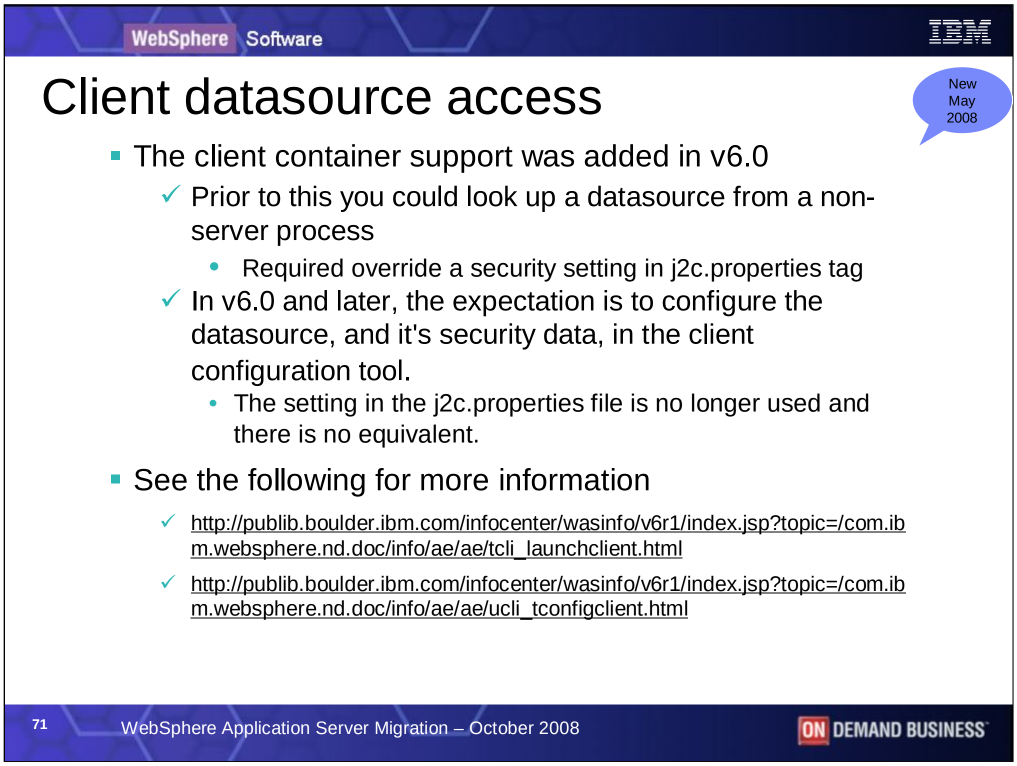


Java Server Faces (JSF)

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
- See the following for more information
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/cweb_javaserver_faces.html

JSF support could be used independently of WebSphere Application Server interfaces prior to v6.0. Starting with WebSphere Application Server v6.0 a level of JSF is included as part of the product. It may not be at the same level currently used by existing customers. You can choose to either use the new support provided by WebSphere Application Server or you can use the same support that you are currently using by utilizing the classloader support as shown.



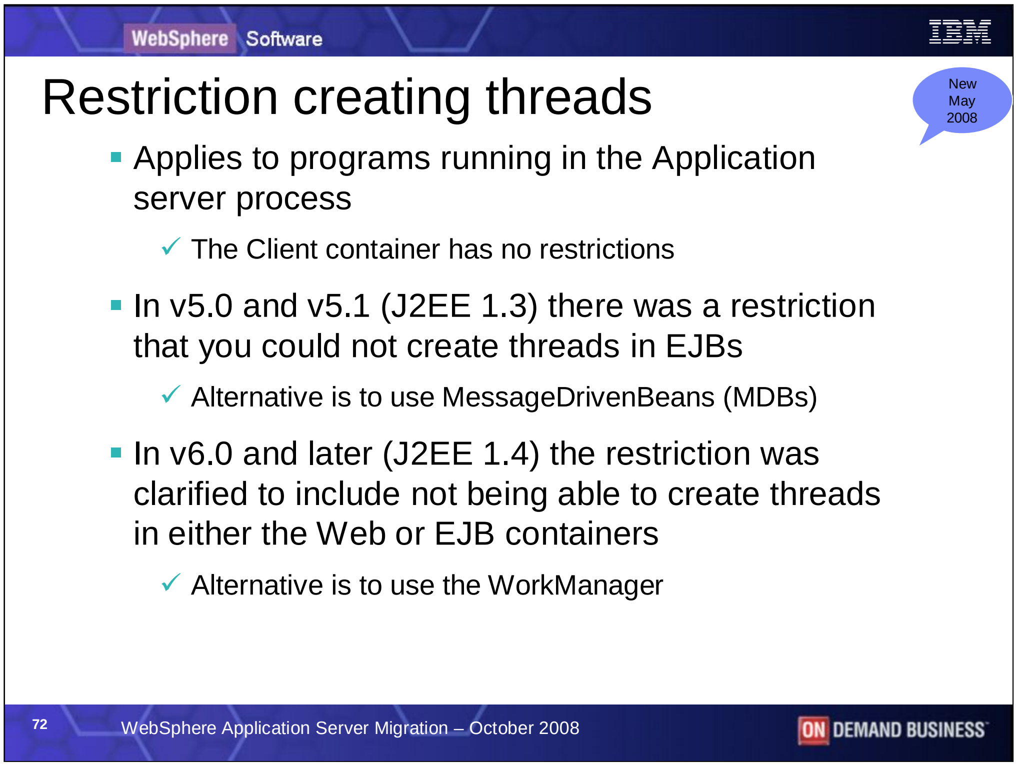
Client datasource access

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.
- See the following for more information
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tcli_launchclient.html
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/ucli_tconfigclient.html

As noted the j2c.properties file could be used to override the security requirement prior to v6.0. The setting needed to look like this:

```
 <security-properties connectionFactoryJNDIName="myDataSource">
    <secureMode>false</secureMode>
  </security-properties>
```

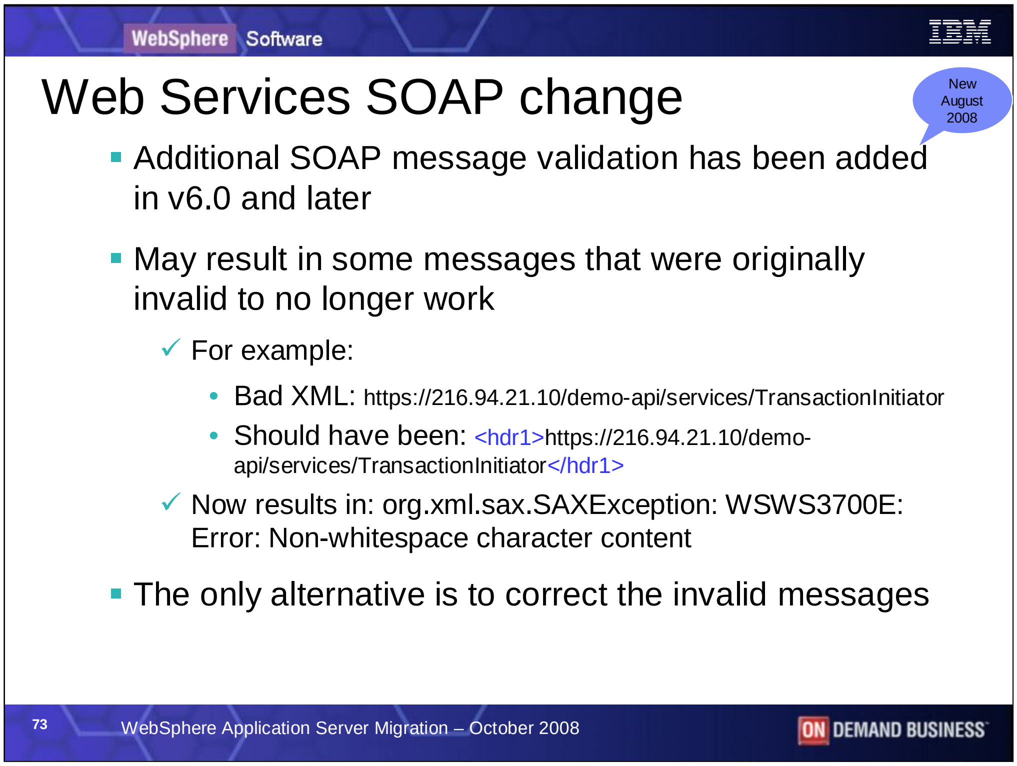


Restriction creating threads

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

The WorkManager function that is available in WebSphere Application Server v6.0 is the basis for JSR 237 joint submission by BEA and IBM. It is commonly referenced as CommonJ WorkManager.



Web Services SOAP change

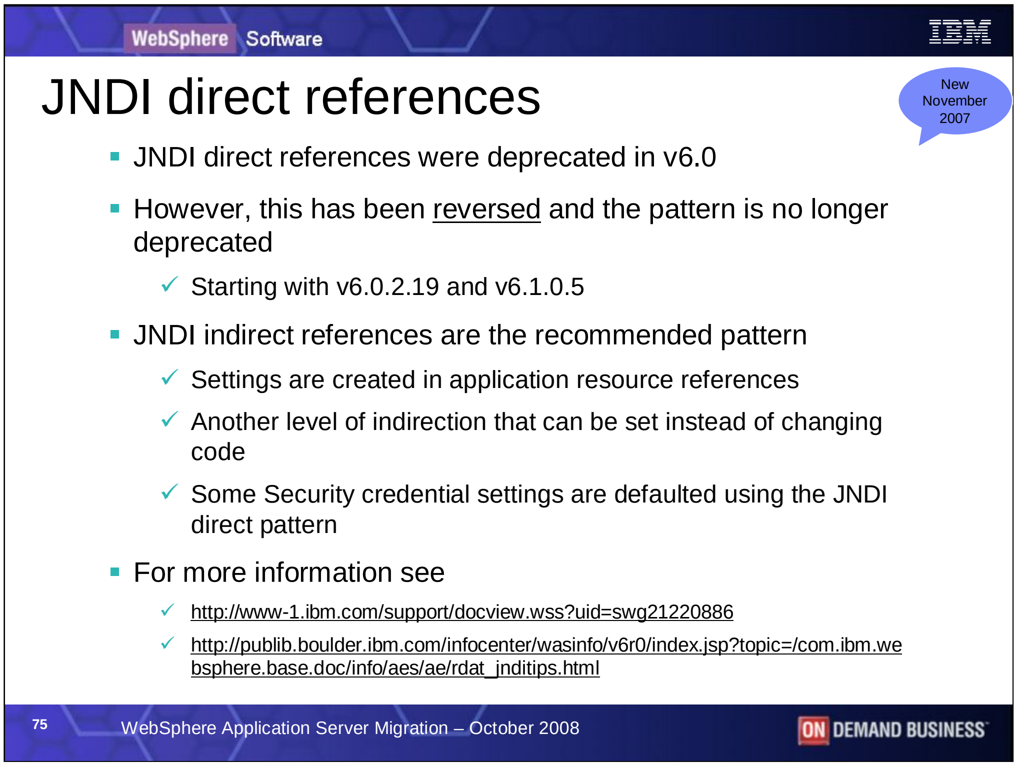
New
August
2008

- 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

There are not many reports of this issue but making sure it is here for completeness

v6.0 cumulative

WebSphere API migration details



WebSphere Software

JNDI direct references

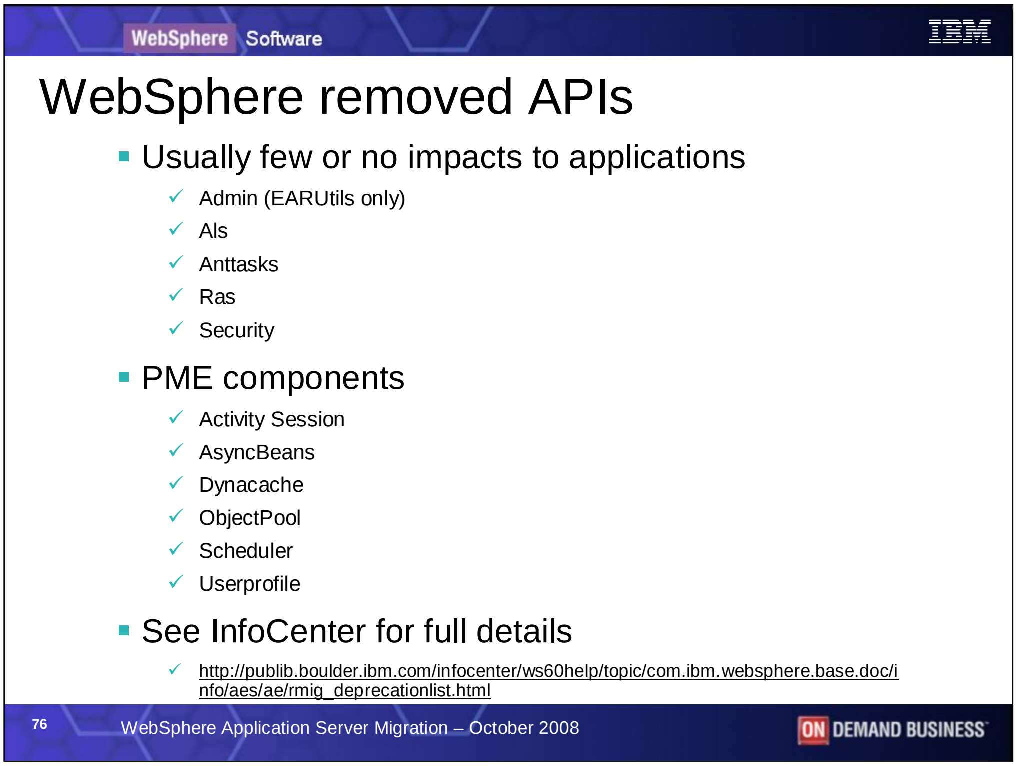
New November 2007

- 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
  - ✓ http://www-1.ibm.com/support/docview.wss?uid=swg21220886
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rdat_jnditips.html

APAR PK32169 is removing the J2CA0294W warning messaging (unless you have debug trace turned on) in Fix Pack 5. This is the warning that you see when performing a direct JNDI lookup of a connection factory i.e. you go straight to the global namespace rather than using a resource-reference and the application's local namespace (*java:comp/env*). This message has always been rather contentious as it asserted that direct JNDI lookups were being deprecated yet various parts of the product still continued to use them. There was even a technote describing how to suppress the warnings as, in some cases, they were causing significant performance degradation.

While it's nice to see clarification that this functionality is not to be deprecated, it is worth pointing out what you are missing by not using resource-references. First, and most obviously, you are missing out on a layer of indirection, hardcoding in your application the name of an administratively defined resource. This leads to the possibility of naming collisions between different applications running in the same cell. However, as the warning message was always at pains to point out, without a resource-reference you are also getting the default settings for various properties.

One of these properties is the resource-authentication or res-auth which defaults to *Application*. This means that, in a secure environment, you have to either specify the connection credentials programmatically on creation or via a component managed authentication alias on the connection factory. Neither of these options is particularly good practice. The first means that the program has to hold the credentials somewhere and the second means that any server-side application with access to the JNDI namespace can create connections. Much better to use a resource-reference, specify a res-auth of *Container*, and then place the alias in the application bindings so it's only available to the application that you intended.



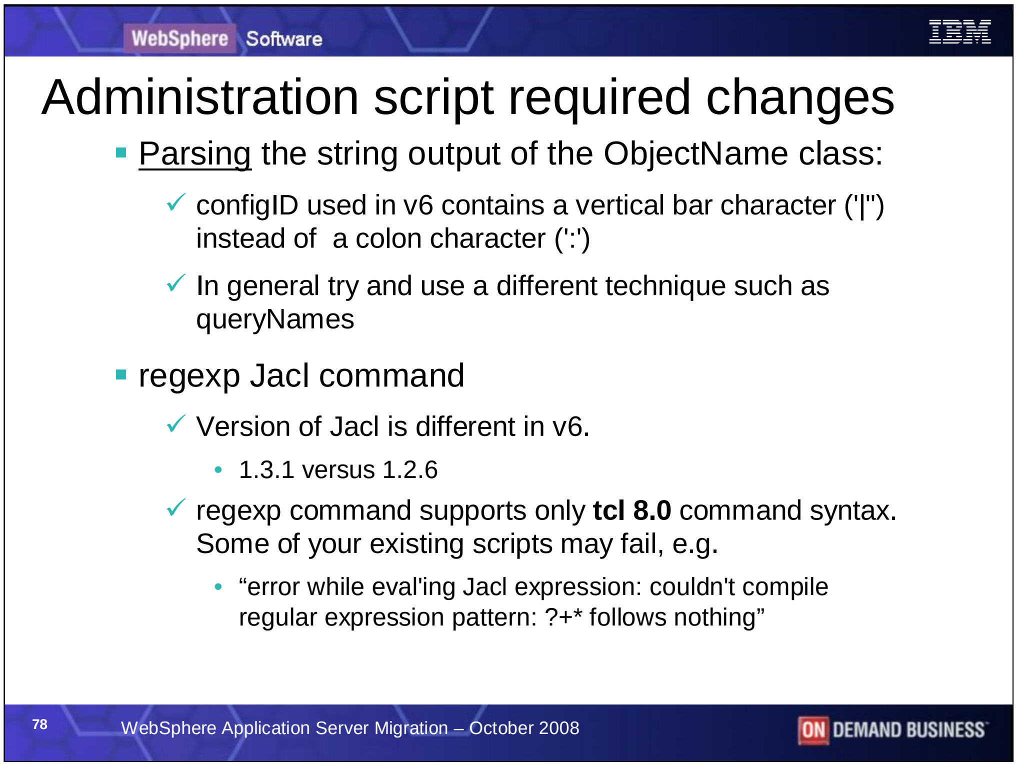
WebSphere removed APIs

- Usually few or no impacts to applications
  - ✓ Admin (EARUtils only)
  - ✓ Als
  - ✓ Anttasks
  - ✓ Ras
  - ✓ Security
- PME components
  - ✓ Activity Session
  - ✓ AsyncBeans
  - ✓ Dynacache
  - ✓ ObjectPool
  - ✓ Scheduler
  - ✓ Userprofile
- See InfoCenter for full details
  - ✓ http://publib.boulder.ibm.com/infocenter/ws60help/topic/com.ibm.websphere.base.doc/info/aes/ae/rmig_deprecationlist.html

Due to general house keeping and/or keeping up with required specification changes results in the removal and deprecation of APIs. Listed in the slide are the removed APIs in WebSphere Application Server v6.0. Not listed on the slide, but listed in the link provided on the slide are the deprecations. As a good rule of thumb, deprecated APIs should be resolved in your code once they have been identified as deprecated. This preemptive strategy allows for better time and cost management involved in making these changes and avoids the “frantic rush” to make these changes once they fall into the category of “removed”.

v6.0 cumulative

Administration Scripting changes details



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"

Due to changes in the JMX specification for allowed format of the ObjectName class, the configID used in Version 6 contains a vertical bar character ("|") instead of a colon character (":") as a delimiter. This change in configID format should not present a problem for most wsadmin scripts because configIDs are not guaranteed to be consistent between wsadmin executions and should not be stored persistently for reuse. ConfigIDs should always be retrieved from a query of the configuration executed right before their usage in a script.

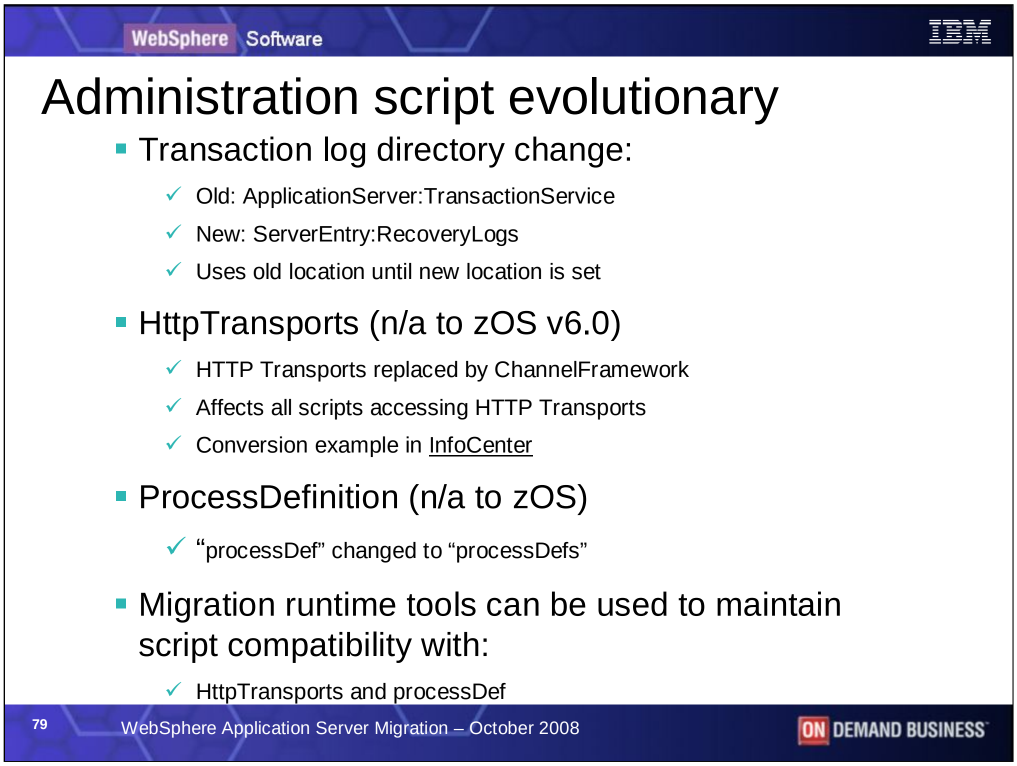
A new version of Jacl(1.3.1) was shipped with WebSphere Application Server v6. With this new Jacl version, the regexp command supports only tcl 8.0 regexp command syntax. If your existing v5.x Jacl scripts use regexp command syntax that is supported in Jacl 1.2.6 but not in Jacl 1.3.1, you may not get a match, or you may get a compile error for your regexp command similar to the following:

```
com.ibm.bsf.BSFException: error while eval'ing Jacl expression:
couldn't compile regular expression pattern: ?+* follows nothing
```

There is no workaround for this problem and development has indicated that this is by design and that there is no easy fix, thus you will need to make modifications to your existing scripts to make them compliant.

To get the entire listing of the error see section 9.5 of the v6 Migration Guide(http://www.redbooks.ibm.com/abstracts/sg246369.html") or InfoCenter document:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate5to6.html



WebSphere Software

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
  - ✓ Conversion example in InfoCenter
- ProcessDefinition (n/a to zOS)
  - ✓ "processDef" changed to "processDefs"
- Migration runtime tools can be used to maintain script compatibility with:
  - ✓ HttpTransports and processDef

79 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

The location of Transaction recovery log attribute has change from ApplicationServer:TransactionService to ServerEntry:RecoverLogs. The ServerEntry:RecoverLogs attribute will not be activated(meaning runtime will not use) until this attribute has been set by you. Your existing scripts will still work and you can make modifications to the ApplicationServer:TransactionService recovery log attribute and the runtime will still honor those changes. Once you make any modifications to the ServerEntry:RecoverLogs attribute the runtime will start ignoring ApplicationServer:TransactionServer recovery log attribute and your scripts will need changing at this point.

Full description of the issue and some example script changes can be located in section 9.2.1 of the v6 Migration Guide(http://www.redbooks.ibm.com/abstracts/sg246369.html") or InfoCenter document:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate5to6.html

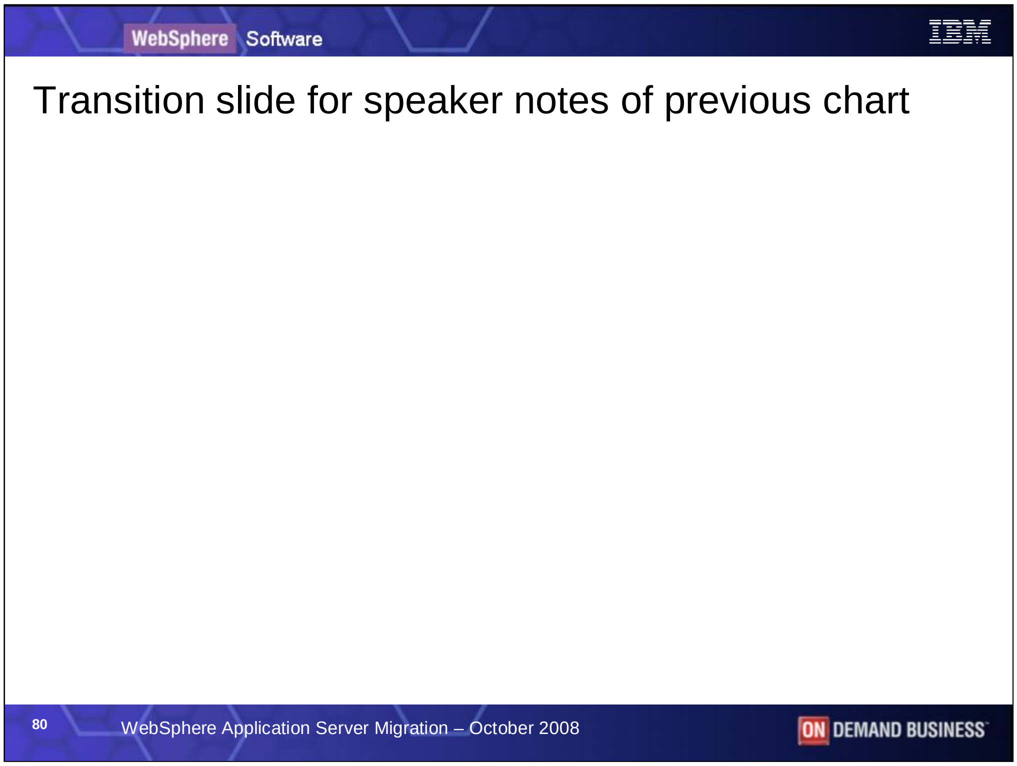
Introduced in v6 is new architecture call the ChannelFramework. This new architecture is communications architecture which replaces the Http Transports. Your existing Http Transports will still work and thus your scripts will not need to be modified to work in v6. The ChannelFramework runtime will convert your Http Transports to Channels/Chains(a component of the ChannelFramework) at runtime.

Note#1: You cannot configure your Http Transports in v6 via the adminconsole, you must use scripts via wsadmin to make any modifications.

Note#2: For the HTTP Transports you can make your scripts portable by first checking for an empty HTTP Transport list. If it is empty this is a full v6.0 configuration and you can use configuration for Channels instead

For instructions on how to convert your Http Transports into ChannelFramework counterparts refer to InfoCenter documents:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate5to6.html

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_chain_transport.html



Transition slide for speaker notes of previous chart

IBM
ON DEMAND BUSINESS

ProcessDef Object has been replaced by ProcessDefs on all platforms except zOS (where it was already ProcessDefs). Again scripts do not need to be changed, v6 still supports the old style. For an example in the script changes refer to InfoCenter document:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate5to6.html

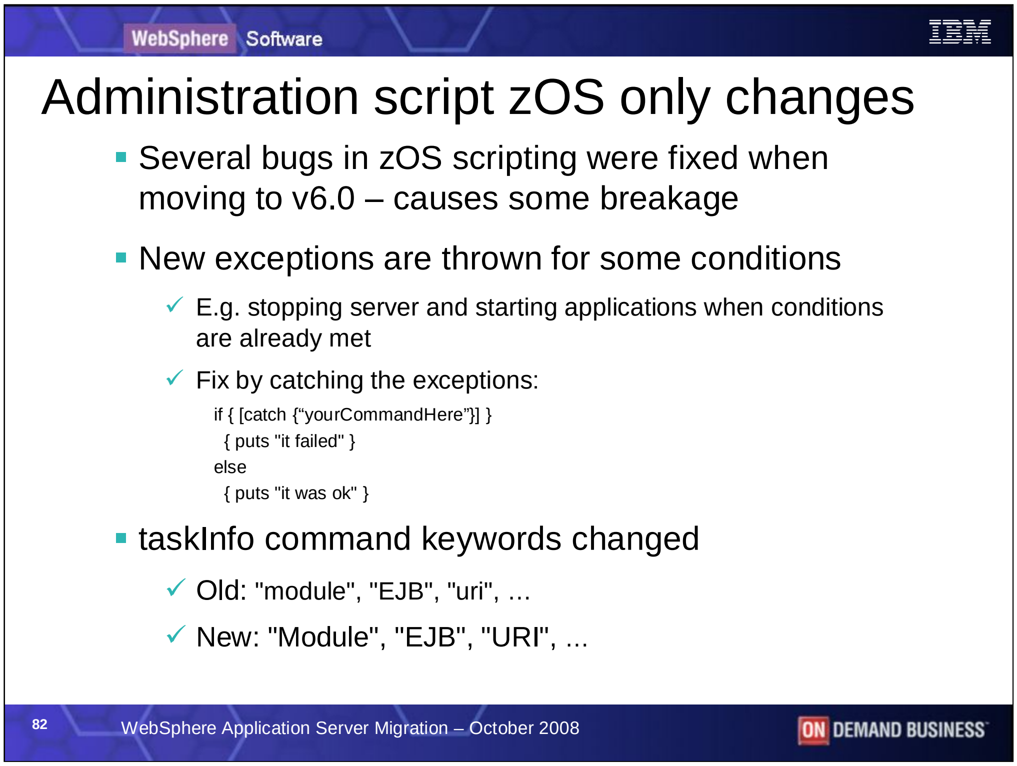
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rxml_mprocessdef.html

The Migration tooling(WASPostUpgrade.sh –scriptCompatibility true(default)) offers support to maintain script compatibility with WebSphere Application Server v5. This can be accomplished by running WASPostUpgrade with the –scriptCompatibility flag set to true or taking the default which happens to be true as well. Once you have completed your updates to your scripts to take advantage of the new features found in v6 you can then run the Migration tool named convertScriptCompatibility.sh which will change your configuration(Http Transports->ChannelFramework and processDef->processDefs) to take advantage of the new v6 support.

If scripts are not an issue or you are willing to make the investment now then running WASPostUpgrade.sh with –scriptCompatibility false is the recommended path. This has the advantage of narrowing the migration window.

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.”



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", ...

The Systems management code was unified across the WebSphere platforms in version 6.0. When this happened some of the behaviors changed on the zOS platform. One of these changes was to throw exceptions for more cases than previously were detected. This can cause some scripts to change behavior, One example of this is throwing an exception on a stopServer command when that server has already been stopped.

The solution is to include more try/catch blocks around wsadmin commands using syntax as follows:

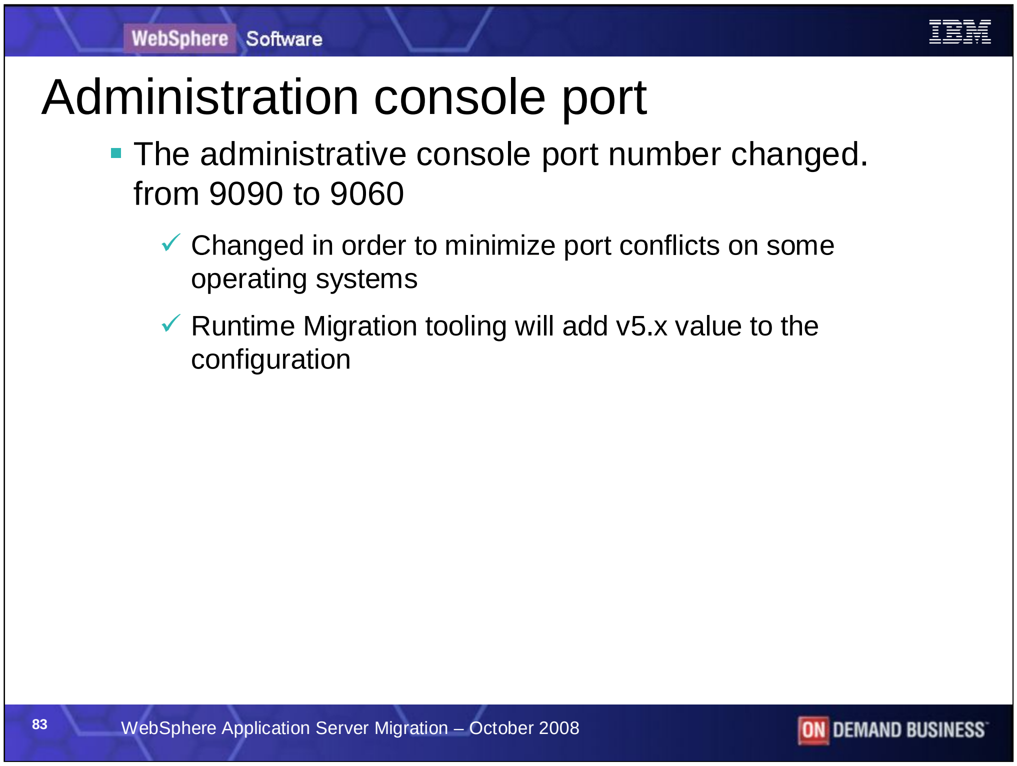
```
if { [catch {“yourCommandHere”}] }
  { puts "it failed" }
else
  { puts "it was ok" }
```

The Systems management code was unified across the WebSphere platforms in version 6.0. When this happened some of the behaviors changed on the zOS platform. One of these changes as in the casing of AdminApp taskInfo command keywords. Many task names have changed between V5.x and V6.x for similar or the exact same operation. You may need to update existing scripts if you are migrating from V5.x to V6.x

For example, some of the old task commands were: "module", "EJB", "uri", while the new task commands are: "Module", "EJB", "URI",

Scripting incompatibility differences

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate5to6.html

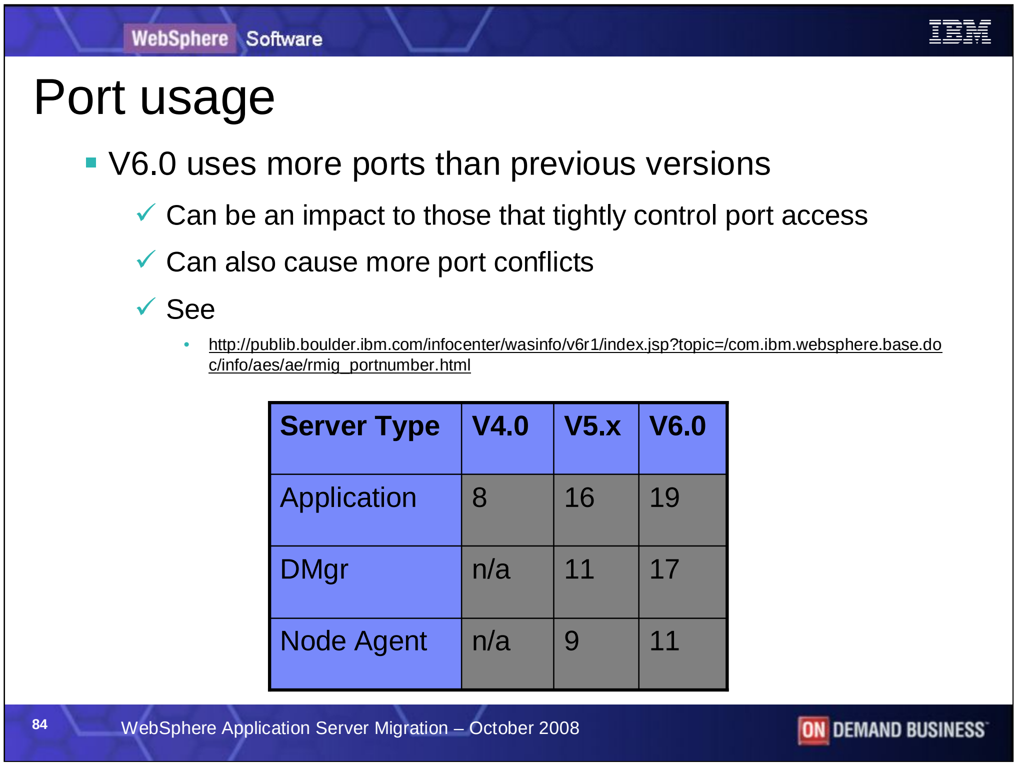


Administration console port

- 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

WebSphere Application Server v6 has changed the administrative console port from 9090 to 9060. This was a result of the port 9090 being used by a commonly used tool(WebSM) on AIX machines to manage the AIX environment. If you are migrating an AIX machine this will not be an issue, because either you already fixed the port issue or you do not have WebSM running on port 9090. The migration tooling will bring the old port value(9090) forward during the migration, so if you are performing migration on a non AIX machine this port change should not be an issue.

Further information on this change can be found in InfoCenter document:
http://publib.boulder.ibm.com/infocenter/wpdoc/v510/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/os_aix.html



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
  - ✓ See
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rmig_portnumber.html

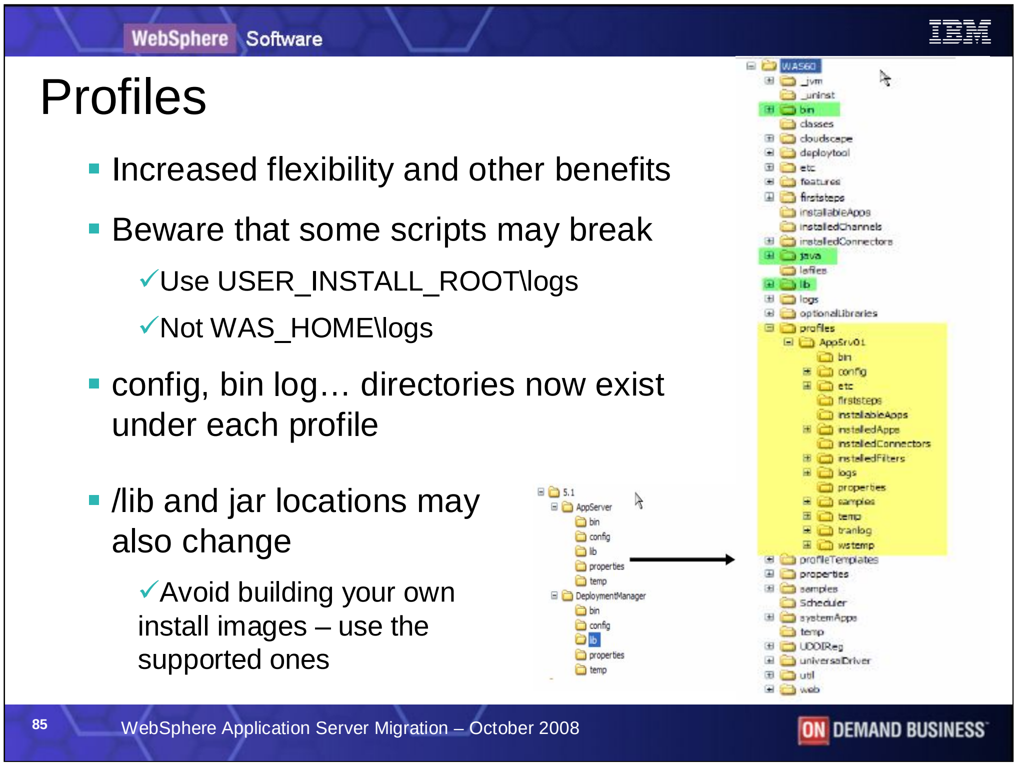
Server Type	V4.0	V5.x	V6.0
Application	8	16	19
DMgr	n/a	11	17
Node Agent	n/a	9	11

The number of ports reserved out of the box for each successive version of WebSphere Application Server has been steadily increasing due to new feature requirements or changes in existing features.

This increase in the number of ports being reserved can have an impact on your environment in the following ways:

- Your Firewall will need to be updated to reflect these additional ports.
- More ports means there is more potential for port conflicts within WebSphere Application Server and with external services residing on the same ports.

The link provided in this slide gives you a summation of the reserved ports for the different versions of WebSphere Application Server. This summation can be used to give you a rough idea of the port ranges that will be in use out of the box, but note these ports may vary slightly when dealing with a migrated environment.



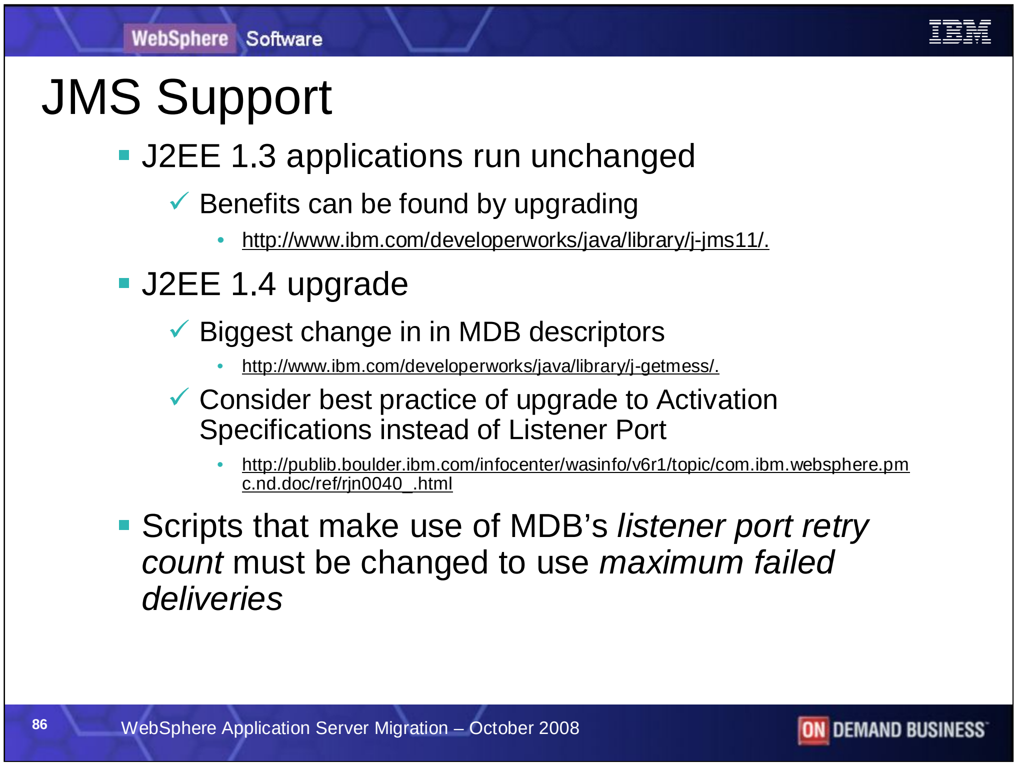
Profiles

- 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

The introduction of Profiles in v6 may cause some migration issues. Profiles in a nutshell is the separation of the product binaries from user data. In pre v6 environments the product binaries(WAS_INSTALL_ROOT, WAS_HOME…) and the user data(USER_INSTALL_ROOT, USER_HOME…) were set to the same value, because there was no clean separation. WebSphere Application Server v6 was changed in such a way that these variables no longer point to the same location. If your scripts were incorrectly using USER_INSTALL_ROOT for example to point to the product binaries then your scripts will no longer work and will need to be updated.

A more detailed explanation of Profiles can be found at: http://www-128.ibm.com/developerworks/websphere/techjournal/0501_williamson/0501_williamson.html

Also worth noting is just about anything else can change as well. If you are dependant on the executable structures of any of the WebSphere Application Server directory structures then you will have a risk that these can change on version boundaries. This is especially true for cases such as building your own client images. For these cases it is much safer to used the installation images that are shipped with and supported by WebSphere Application Server.



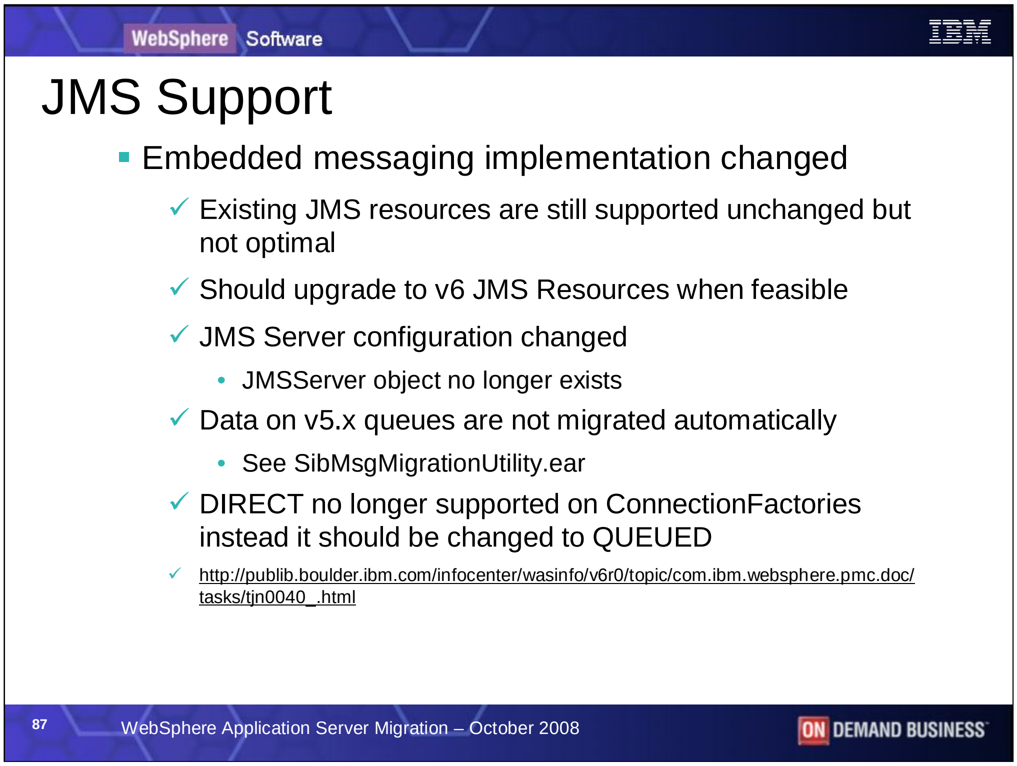
JMS Support

- J2EE 1.3 applications run unchanged
  - ✓ Benefits can be found by upgrading
    - http://www.ibm.com/developerworks/java/library/j-jms11/.
- J2EE 1.4 upgrade
  - ✓ Biggest change in in MDB descriptors
    - http://www.ibm.com/developerworks/java/library/j-getmess/.
  - ✓ Consider best practice of upgrade to Activation Specifications instead of Listener Port
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.pmc.nd.doc/ref/rjn0040_.html
- Scripts that make use of MDB's *listener port retry count* must be changed to use *maximum failed deliveries*

JMS has went through some changes and enhancements and details can be found in the links provided in the slide.

Scripts will need to be modified if you are referencing MDB's listener port retry count. In v6 the Java Connection Architecture has no concept of a listener port retry count, so it is not supported by the v6 default messaging provider. This should not be an issue, the v6 default messaging provides destinations with a maximum failed deliveries setting which can be used as a direct replacement.

More information on this can be found in InfoCenter document:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.pmc.express.doc/ref/rjn0040_.html



WebSphere Software

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
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.pmc.doc/tasks/tjn0040_.html

87 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

Embedded messaging has been phased out in WebSphere Application Server v6 and was replace with the default messaging provider. Your existing JMS resources will still operate correctly unchanged, but this not optimal and should be migrated when resources and time permit.

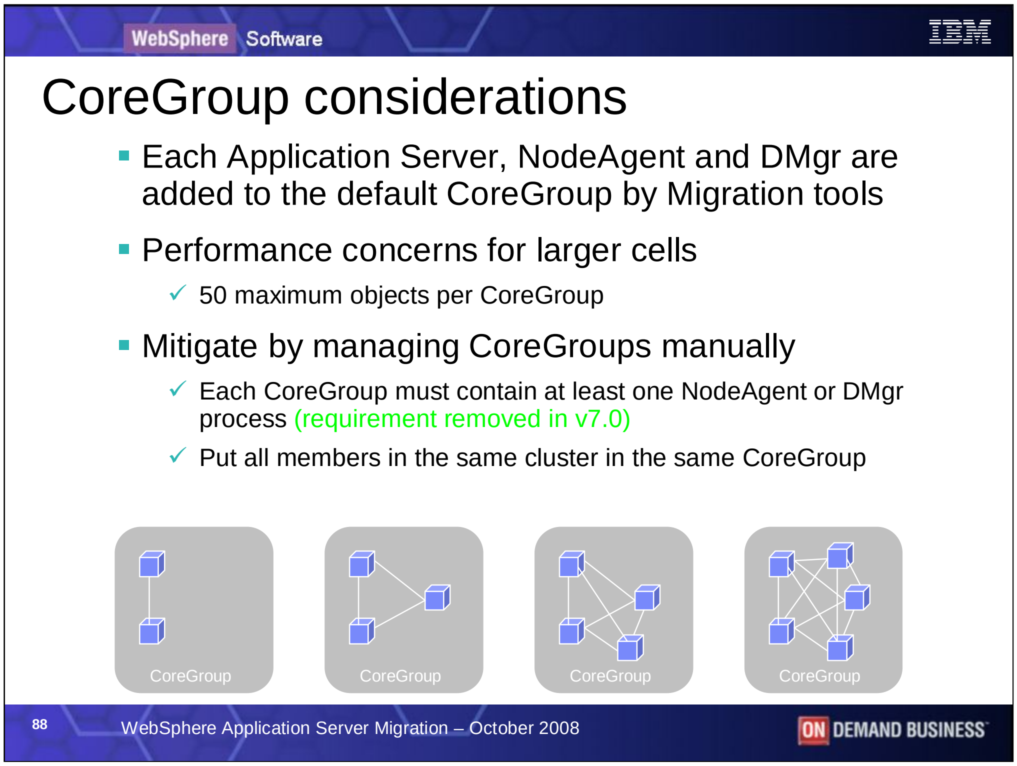
JMS Server configuration has been removed. It is unlikely that your scripts are directly referencing the JMSServer Object(more than likely you will be interacting with JMS resources), but if you are then your scripts will need to be modified to be compliant with v6.

Data on the v5 queues is not migrate as part of the Migration tooling(WASPreUpgrade.sh and WASPostUpgrade.sh). This will need to be done manually in one of two ways. Either you can drain the queues, by not allowing new messages to be published and allowing messages to be read and once queue is empty shut down applications completely from using v5 JMSServer and start using v6 equivalent. Other option is to used a utility that is provided for download called SibMsgMigrationUtility.ear which will migrate your queue information forward into v6.

When migrating a WebSphere Application Server Version 5 node to Version 6, you do not need to make any changes to JMS resource definitions. JMS applications can continue to use their same configurations of Version 5 JMS resources, with the following exception.

The exception to this is for JMS applications that use the Version 5 embedded messaging provider's DIRECT port for publish/subscribe messaging, as set on the WebSphere topic connection factory. If any Version 5 WebSphere topic connection factory has the Port property set to DIRECT, change it to QUEUED before use with the Version 6 default messaging provider.

More information on Messaging migration can be found in the link provided in the slide.



CoreGroup considerations

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

88 WebSphere Application Server Migration – October 2008

WebSphere Application Server includes a high availability manager component. The services that the high availability manager provides are only available to WebSphere Application Server components.

A high availability manager instance runs on every application server, proxy server, node agent and deployment manager in a cell. A cell can be divided into multiple high availability domains known as core groups. Each high availability manager instance establishes network connectivity with all other high availability manager instances in the same core group, using a specialized, dedicated, and configurable transport channel. The transport channel provides mechanisms which allow the high availability manager instance to detect when other members of the core group start, stop, or fail.

Within a core group, high availability manager instances are elected to coordinate high availability activities. An instance that is elected is known as a core group coordinator. The coordinator is highly available, such that if a process that is serving as a coordinator stops or fails, another instance is elected to assume the coordinator role, without loss of continuity.

During migration the Migration tooling will add all Application Servers, NodeAgents and DMgr to the default CoreGroup. WebServers and Generic Servers are the only server types that are not added to CoreGroups.

Web Servers and Generic Servers are the only server types not added to CoreGroup.

Due to way Coregroups members communicate amongst themselves there are potential performance issues with Coregroups with whose membership count exceeds 50 objects (This is not a cut in stone number, it’s just a guide for migration planning purposes). If your environment contains more than 50 processes(Application Servers, NodeAgents, DMgr) then Coregroup migration planning is a must. Migration tooling will need manual intervention to accomplish this Coregroup migration. Simplified view of steps are as follows(just enough info to give you an idea as to what is involved):

1) Migrate DMgr

2) Migrate each node.

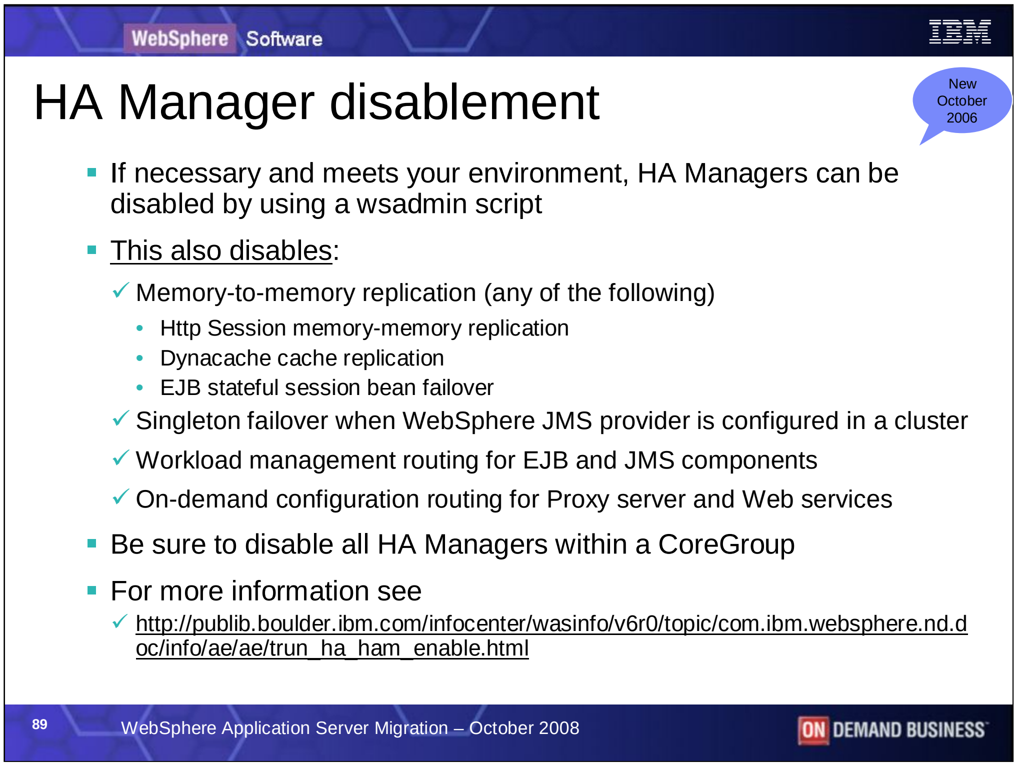
3) Migration tooling will add all of your node processes to the DefaultCoreGroup, so keep an eye on it, and once its membership grows to roughly 50 you will place these into a unique coregroup.

4) DefaultCoreGroup should contain few or no members after step 3.

5) Continue on with the migration of the nodes.

6) Repeat steps 3-5 as many times as needed.

A much more detailed description of the planning and steps involved to accomplish the above steps can be found in InfoCenter



HA Manager disablement

New October 2006

- If necessary and meets your environment, HA Managers can be disabled by using a wsadmin script
- This also disables:
  - ✓ Memory-to-memory replication (any of the following)
    - Http Session memory-memory replication
    - Dynacache cache replication
    - EJB stateful session bean failover
  - ✓ Singleton failover when WebSphere JMS provider is configured in a cluster
  - ✓ Workload management routing for EJB and JMS components
  - ✓ On-demand configuration routing for Proxy server and Web services
- Be sure to disable all HA Managers within a CoreGroup
- For more information see
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/trun_ha_ham_enable.html

HA Management capabilities utilize additional resources(processor, memory…) and it is understood that not all environments require the capabilities that the HA Manager provides, so a mechanism was introduced to disable these capabilities. You can disable HA Management support via wsadmin scripting.

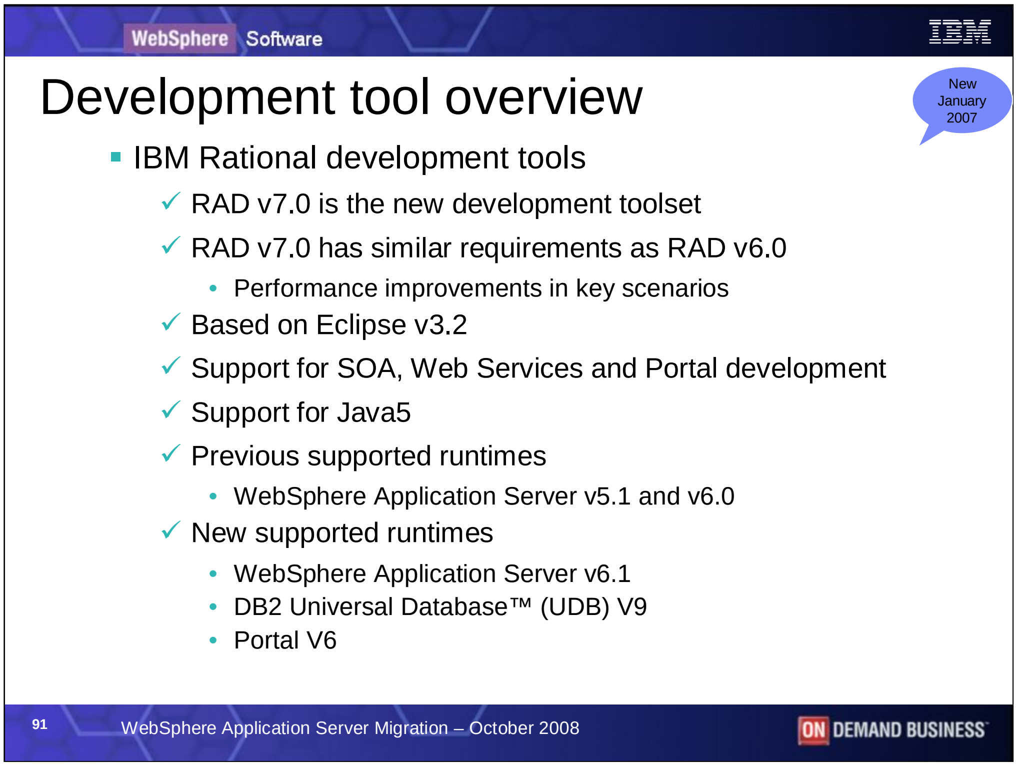
Remember any changes made require a recycle of the affected CoreGroup. Additionally, all members of the CoreGroup either must have HA Management enabled or disabled, there is no support for a mixed bag.

Link provided in slide provides more detailed description and scripting examples to accomplished what was just discussed.

WebSphere Software

v6.1 specific changes

October 2008



Development tool overview

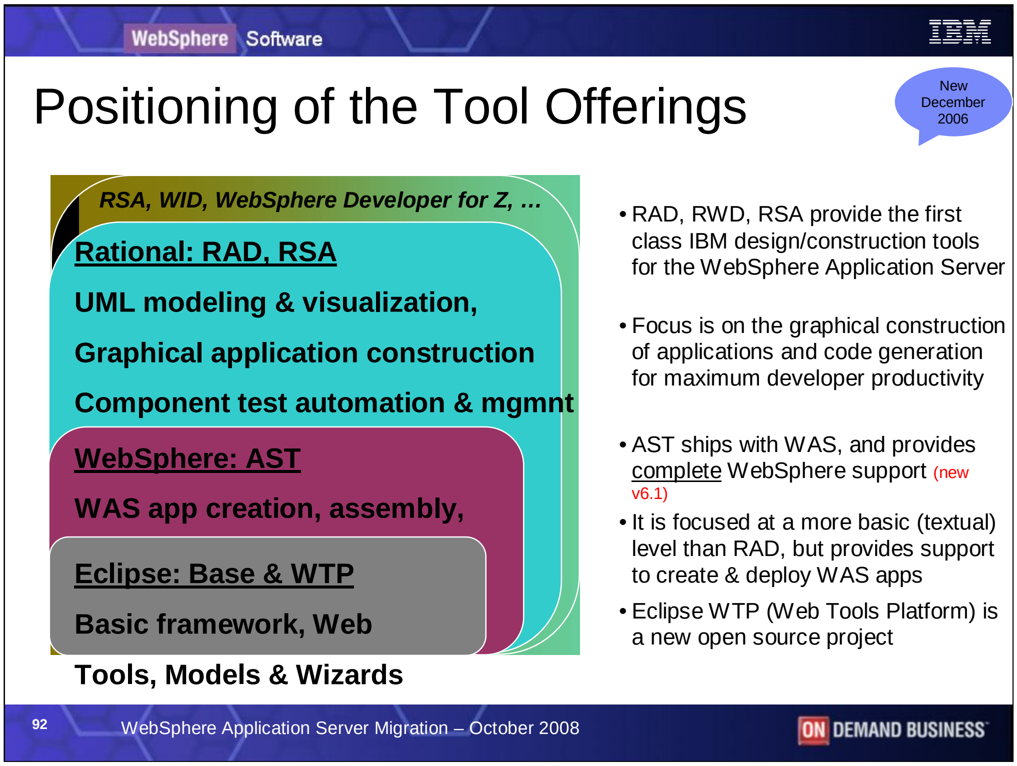
New January 2007

- 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

RAD v7.0 is the newest development toolset designed to work with the latest WebSphere Application Server v6.1. A summary of performance tuning and RAD v7.0 requirements can be found at the following sites:

http://www-306.ibm.com/software/awdtools/developer/application/sysreq/index.html?S_CMP=rnav

http://www-128.ibm.com/developerworks/rational/library/06/1205_lloyd_rad/



Positioning of the Tool Offerings

New December 2006

RSA, WID, WebSphere Developer for Z, …

Rational: RAD, RSA

UML modeling & visualization,

Graphical application construction

Component test automation & mgmnt

WebSphere: AST

WAS app creation, assembly,

Eclipse: Base & WTP

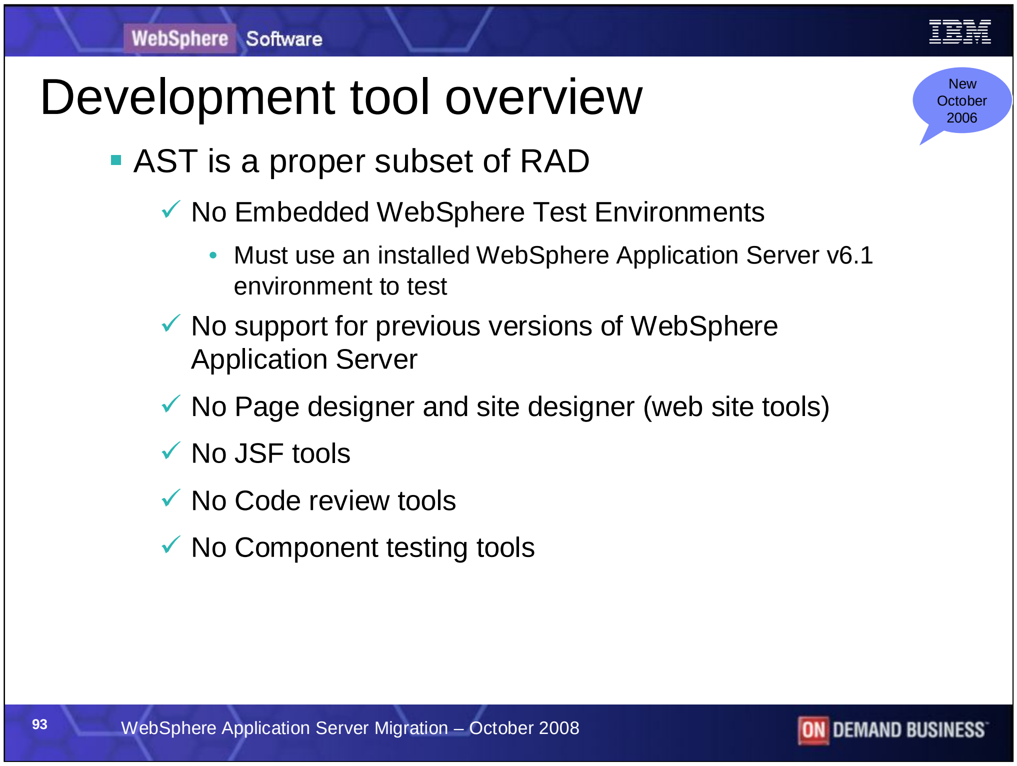
Basic framework, Web

Tools, Models & Wizards

- 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 (new v6.1)
- 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

The main thing to take away from this slide is that WebSphere Application Server v6.1 ships with AST which provides complete WebSphere Application Server v6.1 support. This tool has a subset of the functionality found in RAD and RSA. As stated, this tool is a subset, so this tool does not contain the modeling and visualization features of RAD and RSA and few other features, but this tool is adequate for many development needs.

More information on this chart can be found in the following article: http://www-128.ibm.com/developerworks/websphere/library/techarticles/0606_petersonr/0606_petersonr.html



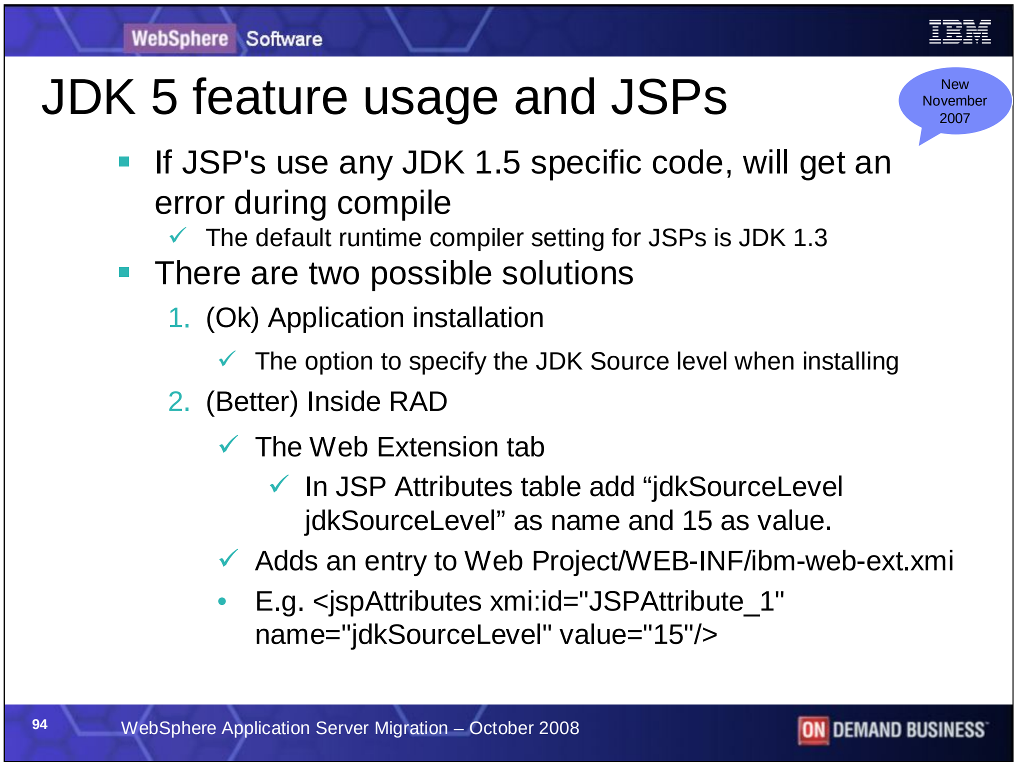
Development tool overview

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

As noted in prior slides AST is a subset of RAD’s features. AST is shipped with WebSphere Application Server v6.1 and is meant to work against that release. There are no embedded WebSphere Test Environments, it is shipped with the full installation of WebSphere Application Server v6.1 and needs to be tested against it. AST that shipped with WebSphere Application Server v6.0 was primarily concerned with assembling and deploying the applications, but come v6.1 AST is a full fledge development environment minus a few missing features only found in RAD or RSA.

For online overview of what AST and what it has to offer see the following presentation: http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/DevelopmentTools/WASv61_ASTOverview/player.html

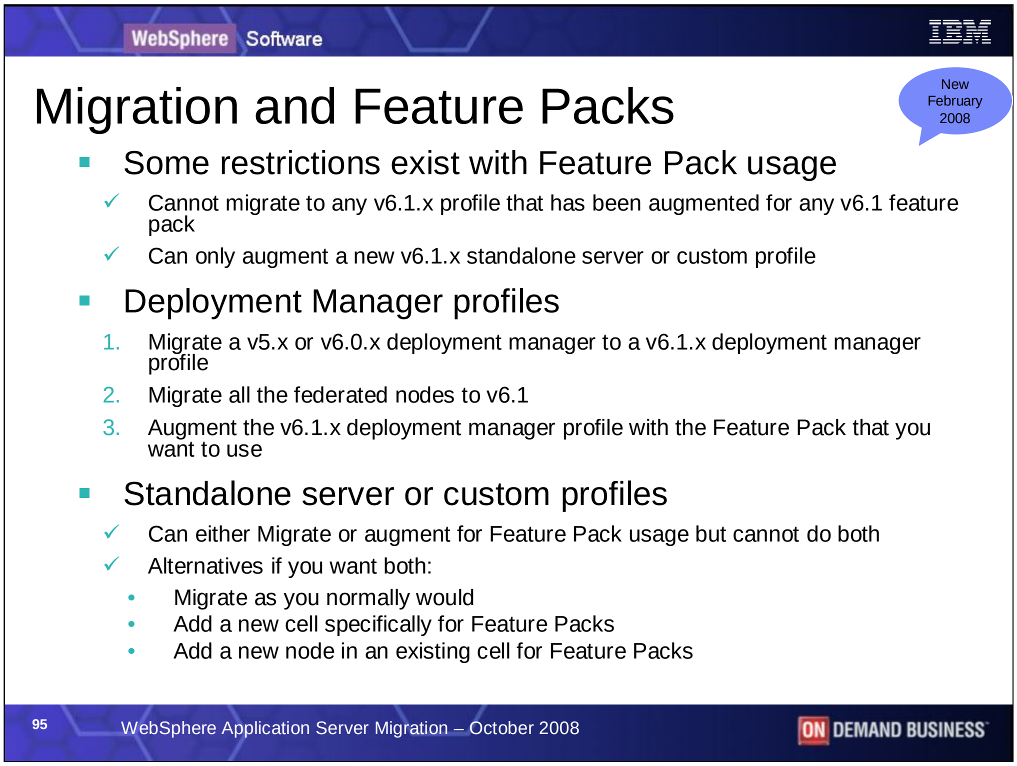


JDK 5 feature usage and JSPs

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 possible 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"/>

The default JDK assumption for JSPs is backward compatibility and not the latest JDK support included in JDK 1.5. This is to provided backward compatibility. As you can see this is tailorable by setting the compiler level to support JDK 1.5 constructs. The second technique is more portable and is the recommended approach.



WebSphere Software

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

WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

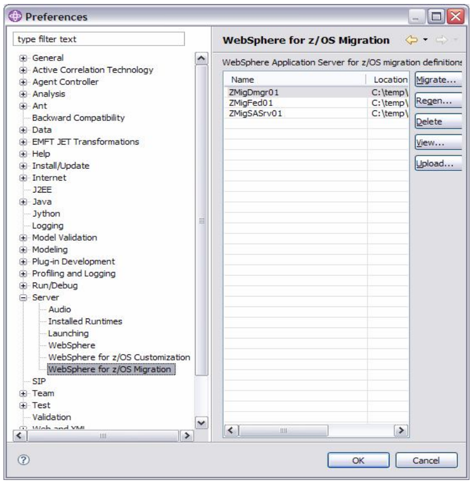
These restrictions exist for all v6.1 feature packs. We understand that this may be painful for some customers but some restrictions exist in order to support the flexibility of delivering Feature Packs.

See http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.wsfep.multiplatform.doc/info/ae/ae/tmig_admin.html for more details.

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



WebSphere Application Server for z/OS migration definitions

Name	Location
ZMigDmgr01	C:\temp\
ZMigFed01	C:\temp\
ZMigSASrv01	C:\temp\

This support was added as an additional Migration path when migrating zOS configurations. This support is available with Application Server Toolkit (AST) 6.1.1 or later.

WebSphere Software

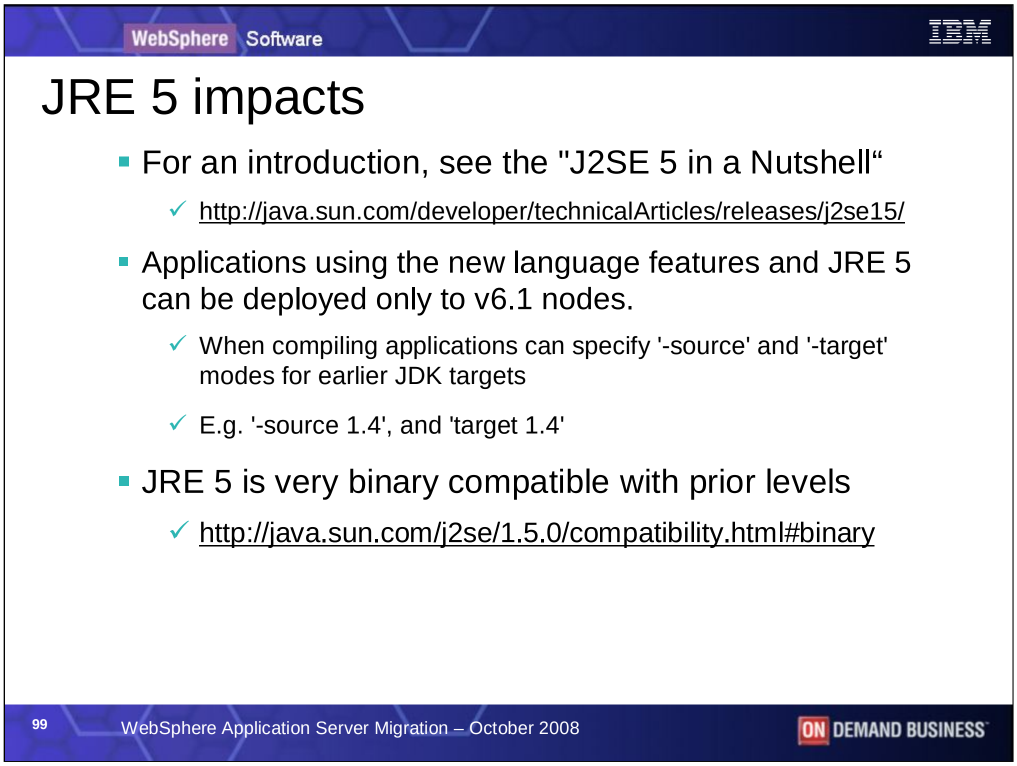
v6.1 cumulative changes

October 2008

ON DEMAND BUSINESS

v6.1 cumulative

Moving to JRE 5



WebSphere Software

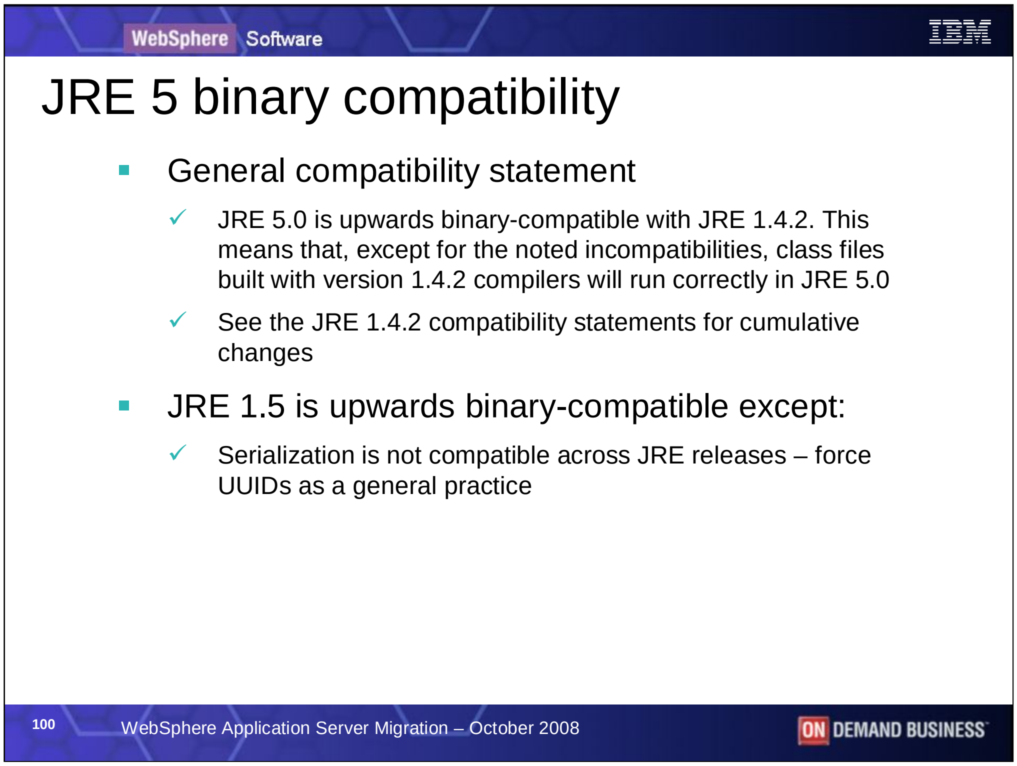
JRE 5 impacts

- For an introduction, see the "J2SE 5 in a Nutshell“
  - ✓ http://java.sun.com/developer/technicalArticles/releases/j2se15/
- 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'
- JRE 5 is very binary compatible with prior levels
  - ✓ http://java.sun.com/j2se/1.5.0/compatibility.html#binary

99 WebSphere Application Server Migration – October 2008

J2SE 5 introduce a number of new features and WebSphere Application Server v6.1 recognizes that you may want to make use of these features so added support for J2SE 5. But keep in mind that any applications that make use of these new language features can be deployed only to v6.1 nodes. This details needs to be taken into account during your migration planning if you are planning for a extended period of time where you are in a mixed cell state. If you have no intentions of using the new features and running these applications on v6.0/v5.x nodes is part of your plan then you need to take special steps when compiling your applications(need to set –source and –target).

In general JDK levels are very compatible with prior levels and JDK 5 is no different. There are however non compatible issues between the versions that you need to be aware of and a complete listing of those can be found in the link provided in the slide.



WebSphere Software

JRE 5 binary compatibility

- General compatibility statement
  - ✓ JRE 5.0 is upwards binary-compatible with JRE 1.4.2. This means that, except for the noted incompatibilities, class files built with version 1.4.2 compilers will run correctly in JRE 5.0
  - ✓ See the JRE 1.4.2 compatibility statements for cumulative changes
- JRE 1.5 is upwards binary-compatible except:
  - ✓ Serialization is not compatible across JRE releases – force UUIDs as a general practice

100 WebSphere Application Server Migration – October 2008

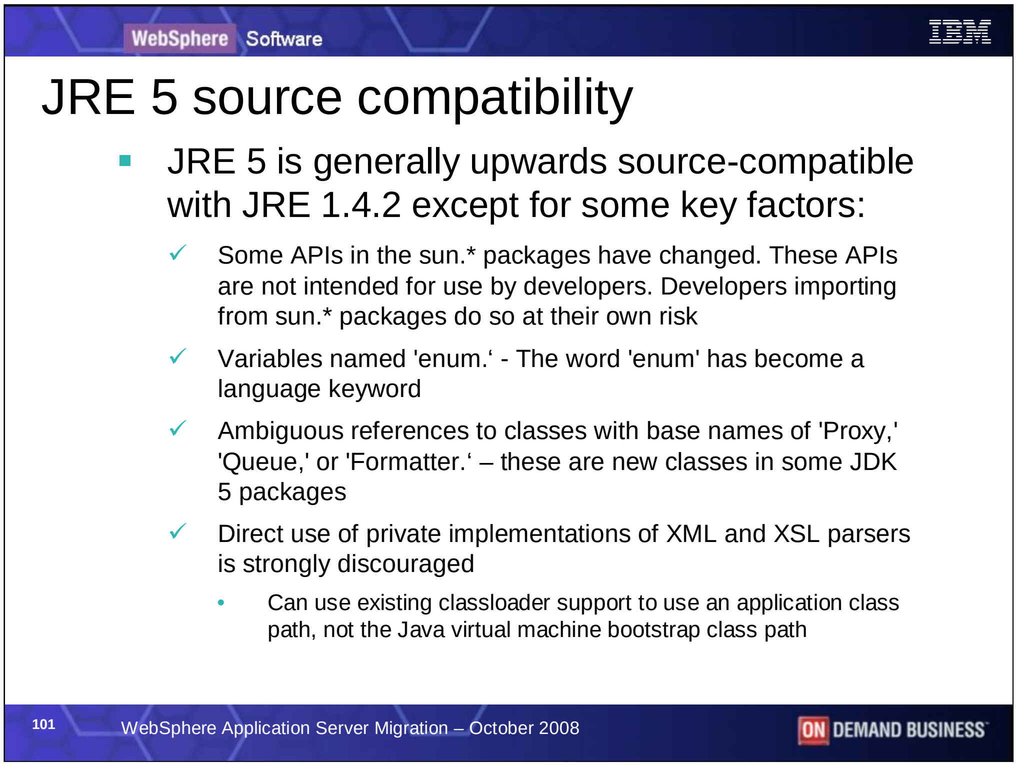
ON DEMAND BUSINESS

JDK has a large customer base as you can imagine, so whenever and wherever possible the JDK tries to maintain compatibility. The JDK follows a few guidelines to address this issue. Maintenance releases within a family(example v1.5.1, v1.5.2…) will maintain both upward and downward binary-compatibility with the exception of the defined incompatibles in the specification. Functional releases(example 1.3, 1.4, 1.5) within a family will maintain upward but not necessarily downward binary-compatibility with each other.

The full list of incompatibilities and guidelines can be found at: http://java.sun.com/javase/technologies/compatibility.jsp

Some earlier compilers violated the class format rules specified by the JDK specification. These class files will not run on JDK v5 and must be recompiled.

Differences in the way JDK v5 and prior versions auto generate serial version UIDs will cause incompatibles in your application. To correct this issue you need to explicitly set a serial version UID for each serializable class. This and other breaking potentially breaking changes in JDK v5 can be found at: http://java.sun.com/javase/technologies/compatibility.jsp



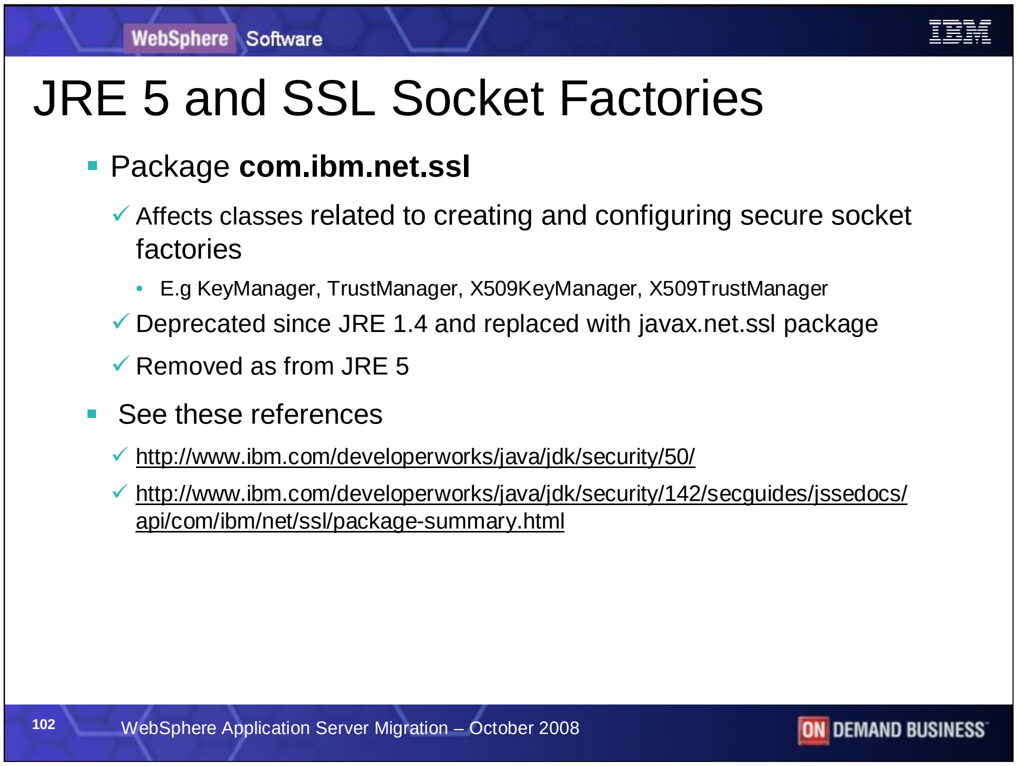
JRE 5 source compatibility

- JRE 5 is generally upwards source-compatible with JRE 1.4.2 except for some key 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
  - ✓ Variables named 'enum.' - The word 'enum' has become a language keyword
  - ✓ Ambiguous references to classes with base names of 'Proxy,' 'Queue,' or 'Formatter.' – these are new classes in some JDK 5 packages
  - ✓ 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

JDK has a large customer base as you can imagine, so whenever and wherever possible the JDK tries to maintain compatibility. The JDK follows a few guidelines to address this issue. Maintenance releases do not introduce any new language features or APIs, so they maintain source-compatibility in both directions. Functional releases and major releases maintain upwards but not downwards source-compatibility. There are a few situations where source-compatibility has not been maintained which are defined in the following link: http://java.sun.com/javase/technologies/compatibility.jsp

One item worth mentioning is that some of the APIs in the sun.* packages have changed. These APIs are private and not intended for use by development. But it is understood that good programming practices are not always followed and thus this should be part of your migration planning.

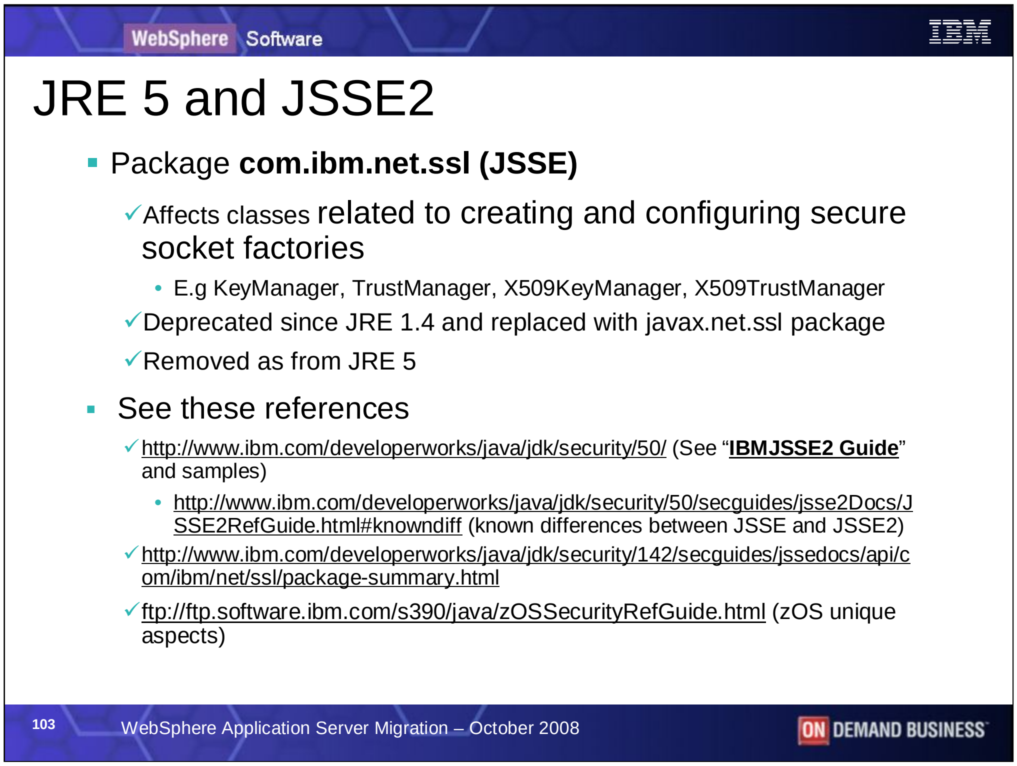
enum is now a java language keyword. If your application is making use enum your application will no longer compile under JDK 5 unless you specify the “-source 1.4” on the javac command.



JRE 5 and SSL Socket Factories

- Package **com.ibm.net.ssl**
  - ✓ 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
- See these references
  - ✓ http://www.ibm.com/developerworks/java/jdk/security/50/
  - ✓ http://www.ibm.com/developerworks/java/jdk/security/142/secguides/jssedocs/api/com/ibm/net/ssl/package-summary.html

Package com.ibm.net.ssl has been deprecated since JKD v1.4 and has been removed in WebSphere Application Server v6.1. It has been replaced with javax.net.ssl package which is part of the JDK since v1.4. Classes related to creating and configuration secure socket factories are effected(KeyManager, TrustManager, X509KeyManager, X509TrustManager). You can obtained a full listing of the details in the links provided in the slide.



JRE 5 and JSSE2

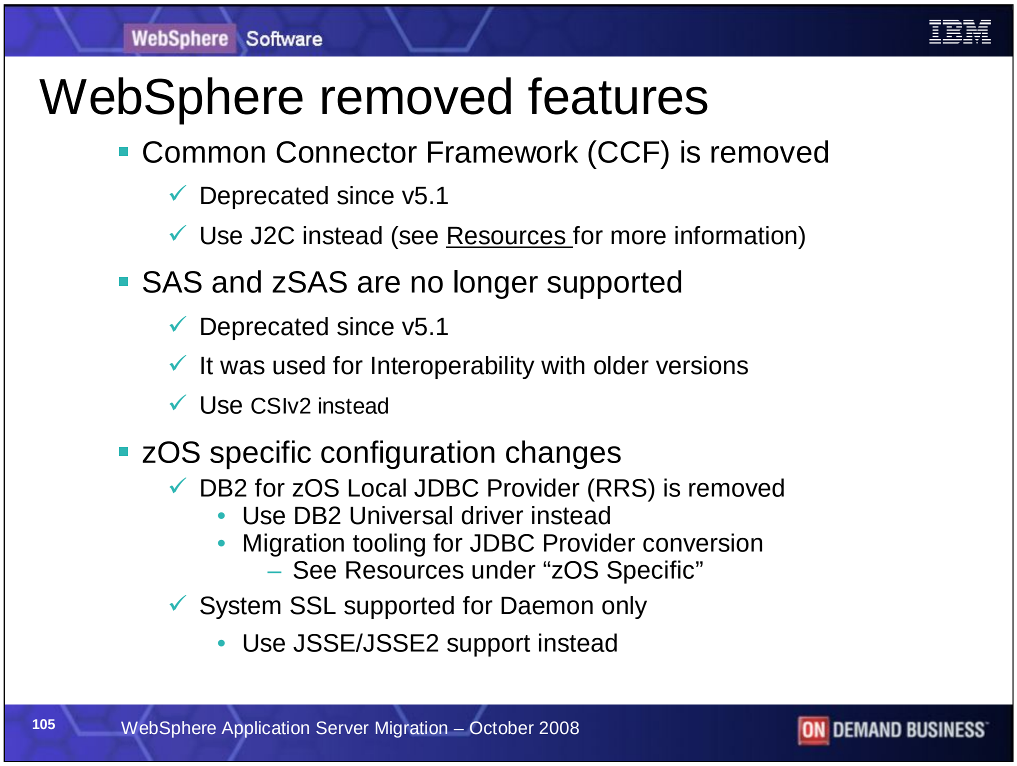
- Package **com.ibm.net.ssl (JSSE)**
  - ✓Affects classes related to creating and configuring secure socket factories
    - E.g KeyManager, TrustManager, X509KeyManager, X509TrustManager
  - ✓Deprecated since JRE 1.4 and replaced with javax.net.ssl package
  - ✓Removed as from JRE 5
- See these references
  - ✓http://www.ibm.com/developerworks/java/jdk/security/50/ (See "**IBMJSSE2 Guide**" and samples)
    - http://www.ibm.com/developerworks/java/jdk/security/50/secguides/jsse2Docs/JSSE2RefGuide.html#knowndiff (known differences between JSSE and JSSE2)
  - ✓http://www.ibm.com/developerworks/java/jdk/security/142/secguides/jssedocs/api/com/ibm/net/ssl/package-summary.html
  - ✓ftp://ftp.software.ibm.com/s390/java/zOSSecurityRefGuide.html (zOS unique aspects)

Package com.ibm.net.ssl has been deprecated since JKD v1.4 and has been removed in WebSphere Application Server v6.1. It has been replaced with javax.net.ssl package which is part of the JDK since v1.4. Classes related to creating and configuration secure socket factories are effected(KeyManager, TrustManager, X509KeyManager, X509TrustManager). You can obtained a full listing of the details in the links provided in the slide.

The first reference "http://www.ibm.com/developerworks/java/jdk/security/50/ (See "**IBMJSSE2 Guide**") contains a good introductory paragraph, including a pointer to samples.

v6.1 cumulative

WebSphere API migration details



WebSphere removed features

- Common Connector Framework (CCF) is removed
  - ✓ Deprecated since v5.1
  - ✓ Use J2C instead (see Resources for more information)
- 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

Generally, IBM will not remove a feature until at least two major releases or three full years (whichever is longer) after the release in which it is deprecated.

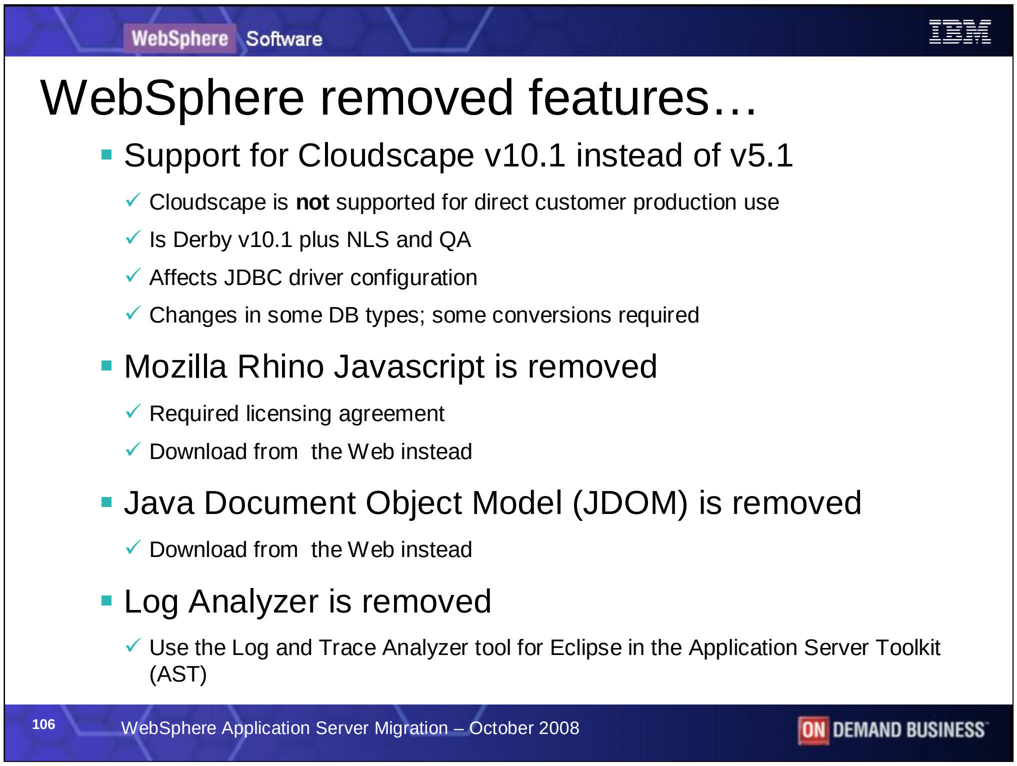
The Common Connector Framework (CCF) has been deprecated since v5.1 and has been removed in v6.1. You should use the industry-standard Java Connector Architecture (J2C) implementation in its place. Further details about CCF to J2C conversion can be found in the following article: http://www-128.ibm.com/developerworks/websphere/library/techarticles/0403_searle/0403_searle1.html

SAS and zSAS has been deprecated since v5.1 and has been removed in v6.1. If you are using the SAS APIs you should modify your code to make use of the Java Authentication and Authorization Service (JAAS) APIs.

The following link provides mappings of the deprecated SAS APIs to their JAAS API equivalents and additional links to additional information on this topic: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tsec_migratecorba.html

Supported of “DB2 for zOS Local JDBC Provider (RRS)” has been removed in WebSphere Application Server v6.1 and it is recommend that you use “DB2 Universal JDBC Driver Provider” in place of it.

The following link contains a whitepaper with more detailed description of the issue and a utility that can aid in your migration: http://www-1.ibm.com/support/docview.wss?uid=swg27007826


WebSphere Software
IBM

WebSphere removed features…

- 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
- Log Analyzer is removed
  - ✓ Use the Log and Trace Analyzer tool for Eclipse in the Application Server Toolkit (AST)

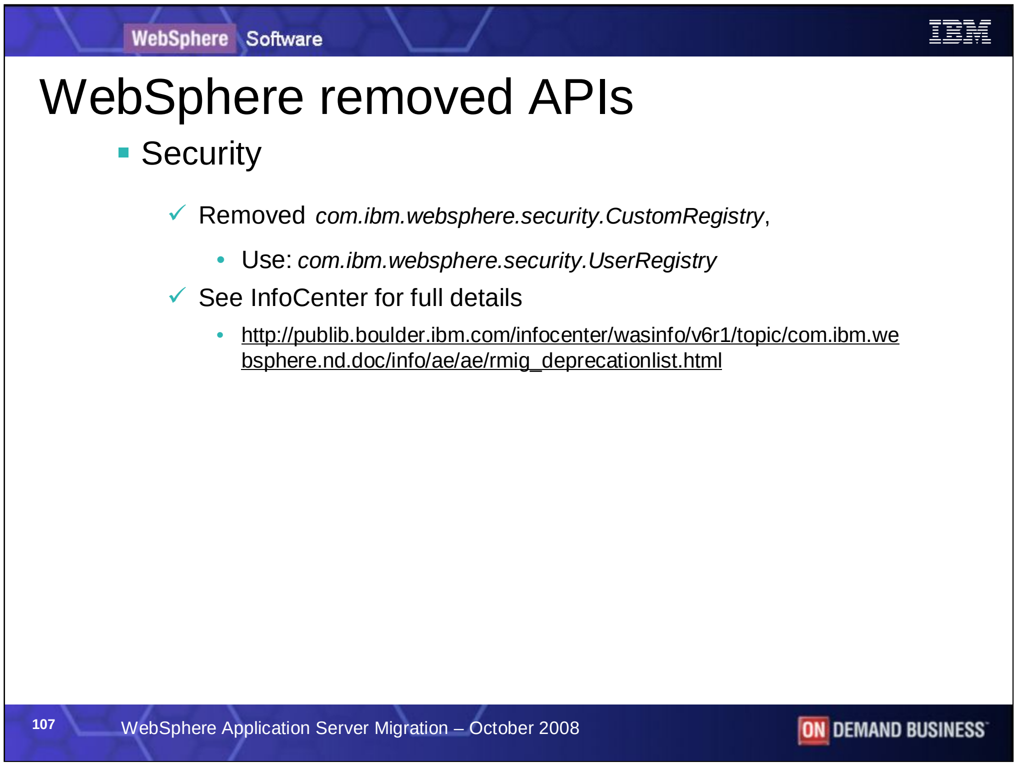
WebSphere Application Server v6.1 has dropped Cloudscape v5.1 and replace it with Cloudscape v10.1. Just like in releases prior to WebSphere Application Server v6.1 Cloudscape is not supported in a production environment and should only be used in your test and development environments. This change does have an impact on the JDBC driver configuration so your configuration needs to be modified and since the Cloudscape DBs are not compatible they must also be migrated. The Migration tooling(WASPostUpgrade) does offer some relief in that it will do much(but not in some circumstances) of the migration for you.

The following InfoCenter document gives you more details on the Migration tooling end as well as links to the manual steps for the cases that the Migration tooling does not support:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tdat_autoCloudSmig.html

Due to a licensing agreement Mozilla Rhino Javascript has been removed from WebSphere Application Server v6.1. You can still use it, but you will need to download it yourself from the web. Mozilla Rhino Javascript can be found at the following site: http://www.mozilla.org/rhino/

JDOM has been deprecated since Websphere Application Server v5.1 and has been removed in v6.1. If you need to make use of JDOM you will need to manually download and install this feature yourself. JDOM can be found at the following site: http://www.jdom.org/

Log Analyzer has been removed in v6.1 instead use Log and Trace Analyzer tool for Eclipse in the AST. For more information, go to Application Server Toolkit > Detecting and analyzing runtime problems > Log and Trace Analyzer in this information center.



WebSphere Software

WebSphere removed APIs

- Security
  - ✓ Removed *com.ibm.websphere.security.CustomRegistry*,
    - Use: *com.ibm.websphere.security.UserRegistry*
  - ✓ See InfoCenter for full details
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_deprecationlist.html

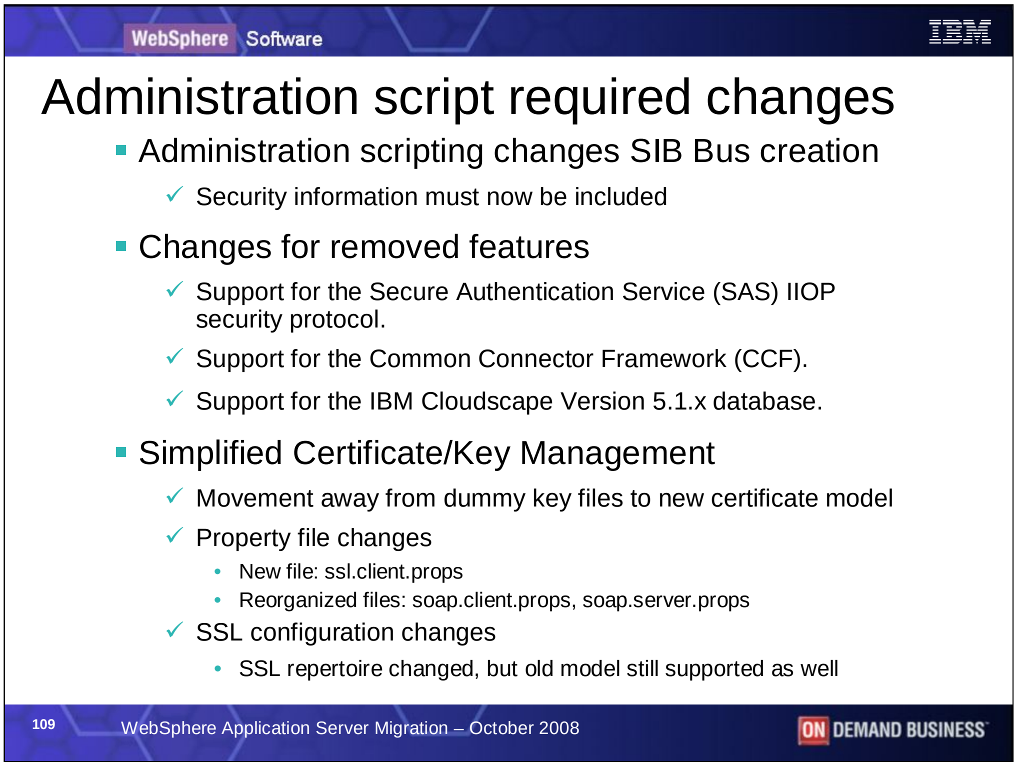
107 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

As noted in slide, CustomerRegistry has been removed and UserRegistry should be used in its place. You can find instructions on this conversion in InfoCenter document:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tsec_migratecustom.html

v6.1cumulative

Administration Scripting changes details


WebSphere Software

Administration script required changes

- Administration scripting changes SIB Bus creation
  - ✓ Security information must now be included
- 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

109 WebSphere Application Server Migration – October 2008

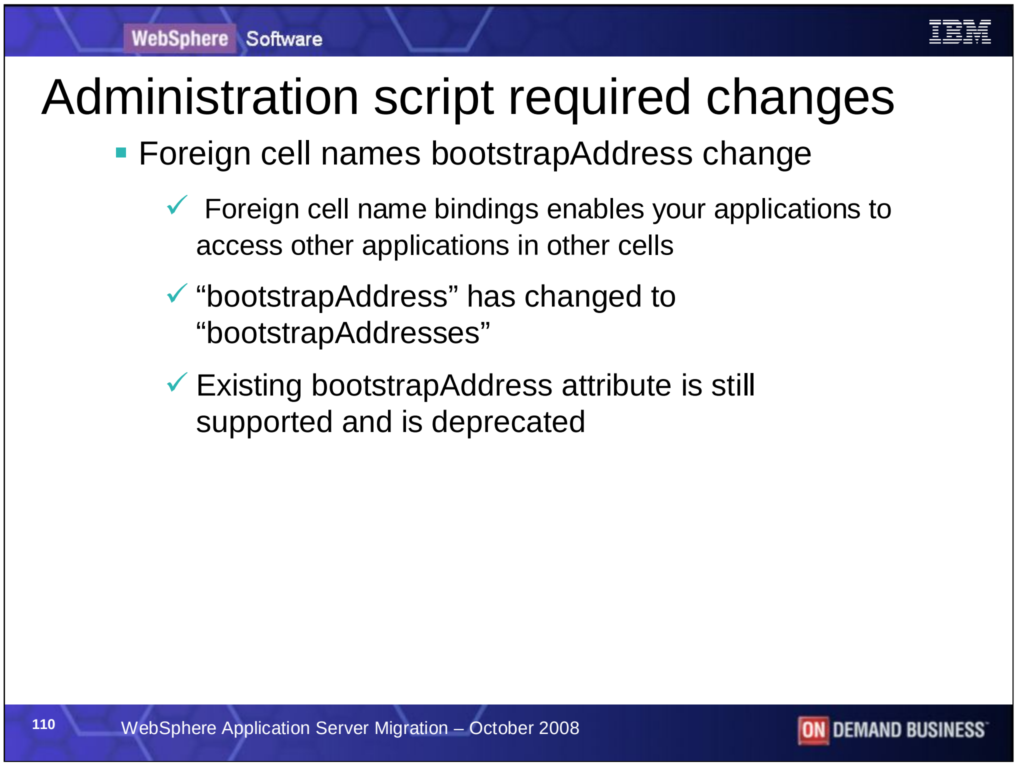
A new parameter is on the createSIBus command related to Security. This change is a side-effect of the other changes to the Security model in WebSphere Application Server v6.1. The –secure parameter has been deprecated and is ignored. It controlled whether or not global security controlled the SIBus Security. The busSecurity parameter has been added. Set this option to TRUE to enforce the authorization policy for the bus, which also requires administrative security to be enabled. Set this option to FALSE if you always want to disable bus security. If administrative security is disabled the bus is insecure.

More Information can be found in InfoCenter document:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.pmc.nd.iseries.doc/ref/rjj_cli_bus_create.html

Both CCF and SAS have been removed from WebSphere Application Server v6.1 after proper deprecation. Any resources that were configured to use these features will need to be changed to use the new functions. For CCF this is J2C, for SAS it is JAAS. The exception is the Daemon process of zOS. This will still use System SSL.

A new way of configuring SSL was introduced in WebSphere Application Server v6.1. When migrating to v6.1, you can update the format for SSL configuration or you can continue to use the v6.0 format. If you encounter errors with your existing administration scripts for SSL configurations, you can manually convert your SSL configuration to the v6.1 format. If you create a new profile without doing Migration you will be using the new SSL model and you must modify your scripts.

More information on this file and how to set this file up can be found in InfoCenter document:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rsec_sslclientpropsfile.html



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

The ability to configure foreign cell name bindings enables your applications to access other applications in other cells. A *foreign cell* is a cell other than the one in which an application resides. A foreign cell name binding is a context binding that resolves to the Cell Root context of a foreign cell.

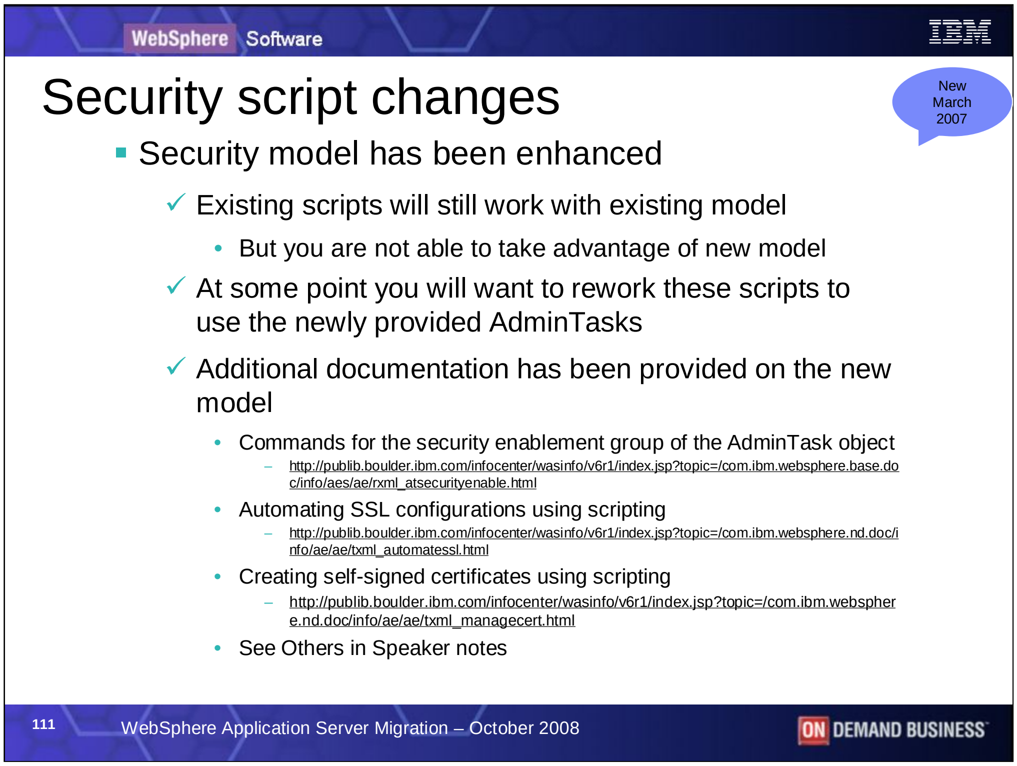
The format of this has changed in the v6.1 default configuration. Edit your version 5.x or 6.0.x foreign cell binding scripts, replacing bootstrapAddress with bootstrapAddresses, the list property introduced in version 6.1. The bootstrap address properties are in the context of the ForeignCell class. The fully qualified name for the version 5.x or 6.0.x property now deprecated is the following:

topology.cell:Cell/foreignCells/bootstrapAddress

The fully qualified name for the bootstrap address list property introduced in version 6.1 is the following:

topology.cell:Cell/foreignCells/bootstrapAddresses

Version 6.1 and later foreign cell binding scripts should use the bootstrap address list property, bootstrapAddresses.



WebSphere Software

IBM

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
      - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rxml_atsecurityenable.html
    - Automating SSL configurations using scripting
      - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_automatessl.html
    - Creating self-signed certificates using scripting
      - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_managecert.html
    - See Others in Speaker notes

111 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

As has been noted the new Security model is in place for v6.1 to provided enhanced support for ease of use and maintainability. The standard approach for WebSphere Application Server compatibility has been followed. Namely, existing scripts will still work but if you want to take advantage of the newest features you will need to update your scripts. In addition to the links noted, these are also available:

Commands for the security enablement group of the AdminTask object

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rxml_atsecurityenable.html

Virtual Member Manager (VMM) management commands using scripting

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rxml_atwimmgt.html

Automating SSL configurations using scripting

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_automatessl.html

SPNEGO TAI configurations using scripting

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rxml_atspnego.html

Creating self-signed certificates using scripting

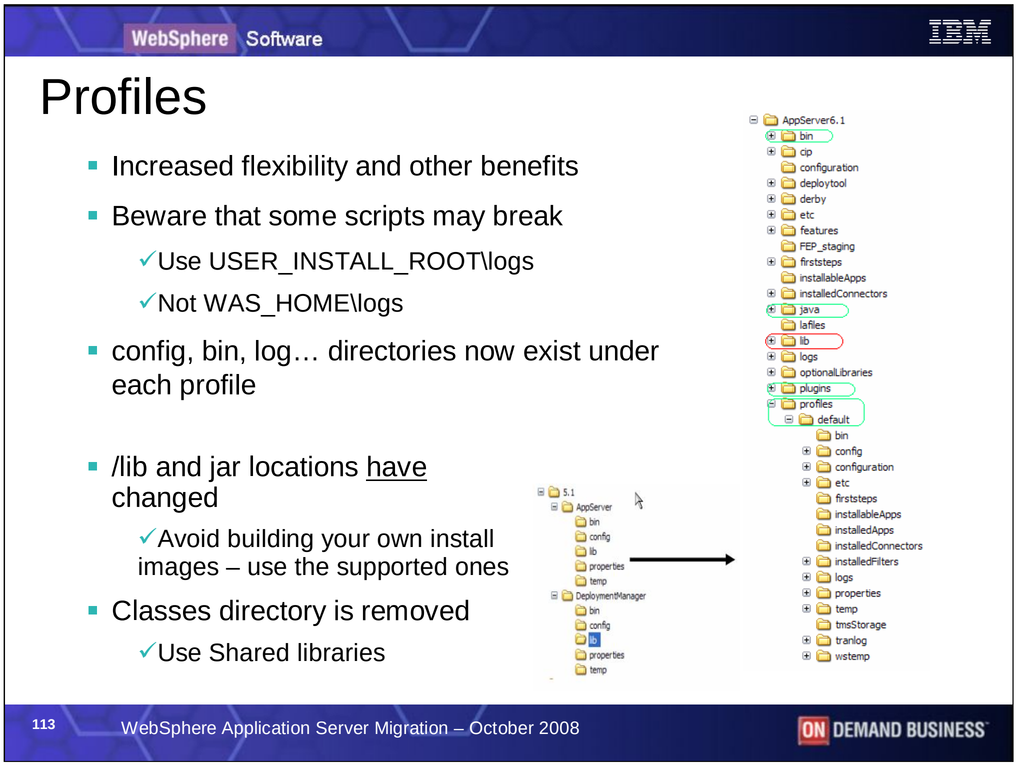
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_managecert.html

Other security tasks documented

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rxml_admintask.html

v6.1cumulative

Other Miscellaneous changes details



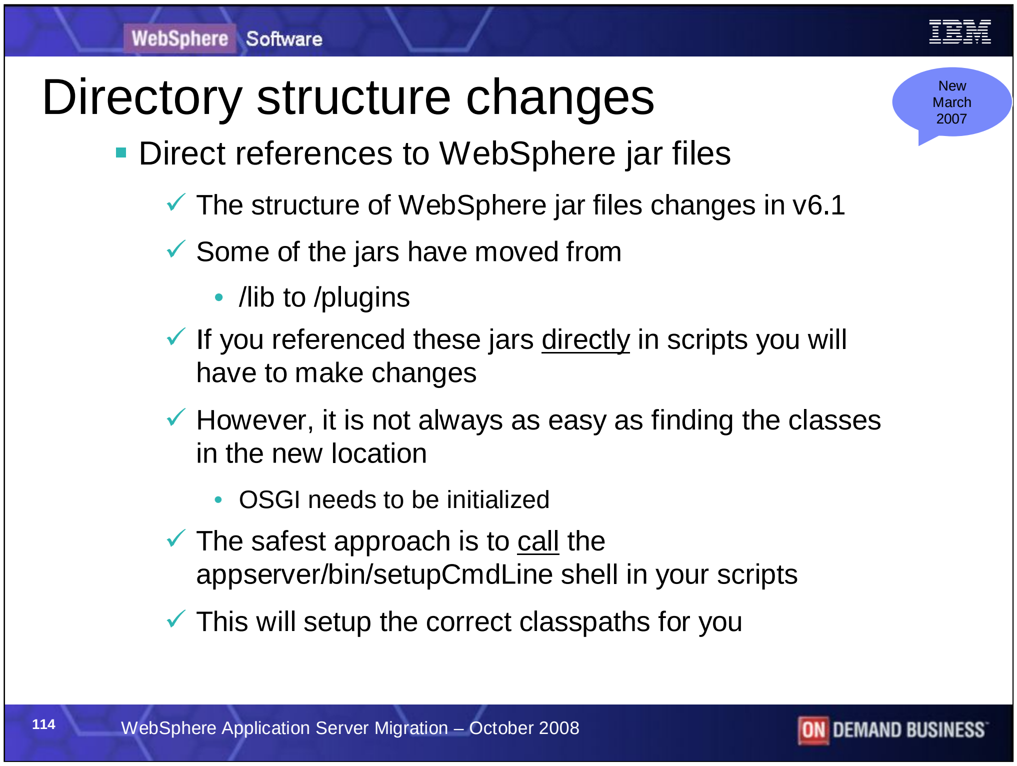
Profiles

- 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

The introduction of Profiles in v6 may cause some migration issues. Profiles in a nutshell is the separation of the product binaries from user data. In pre v6 environments the product binaries(WAS_INSTALL_ROOT, WAS_HOME…) and the user data(USER_INSTALL_ROOT, USER_HOME…) were set to the same value, because there was no clean separation. WebSphere Application Server v6 was changed in such a way that these variables no longer point to the same location. If your scripts were incorrectly using USER_INSTALL_ROOT for example to point to the product binaries then your scripts will no longer work and will need to be updated.

A more detailed explanation of Profiles can be found at: http://www-128.ibm.com/developerworks/websphere/techjournal/0501_williamson/0501_williamson.html

Also worth noting is just about anything else can change as well. If you are dependant on the executable structures of any of the WebSphere Application Server directory structures then you will have a risk that these can change on version boundaries. This is especially true for cases such as building your own client images. For these cases it is much safer to used the installation images that are shipped with and supported by WebSphere Application Server.

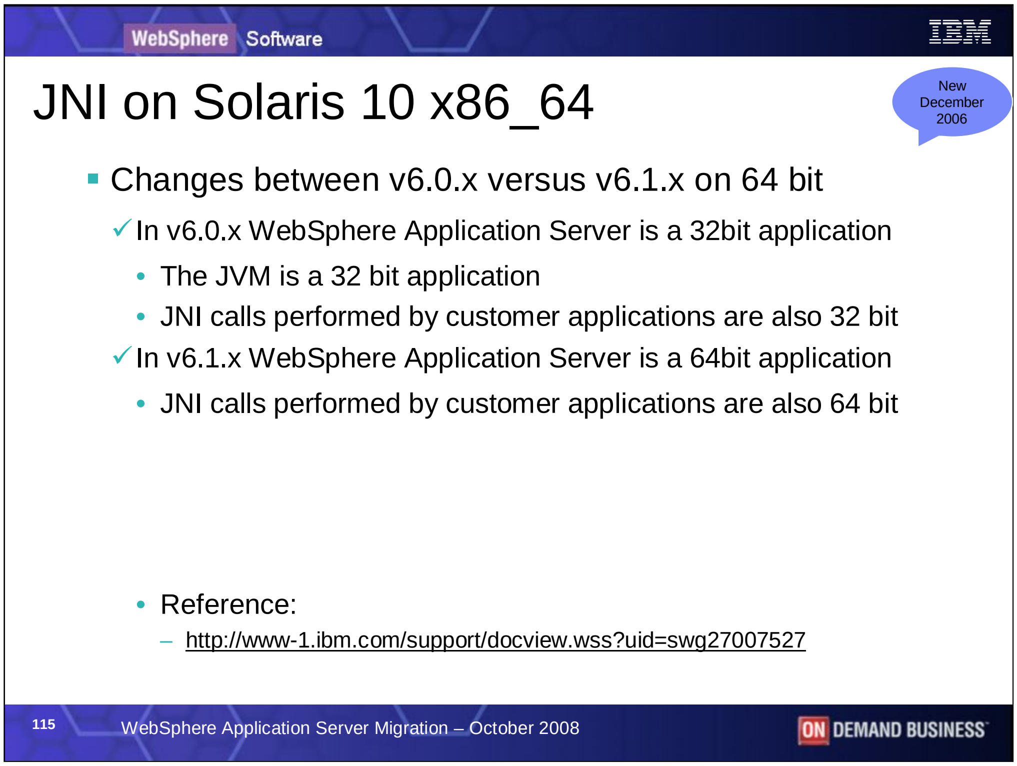


Directory structure changes

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

A change was made in v6.1 to use OSGI for packaging the WebSphere Application Server. This moved some of the classes that were used by customer scripts into the jars in the /plugins directory. It is not as straight forward as finding the new location for the classes. The OSGI bundles must be initialized. This is done by the setupCmdLine.bat or setupCmdLine.sh scripts. The correct approach is to call the appropriate script in your script.



WebSphere Software

New December 2006

JNI on Solaris 10 x86_64

- 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

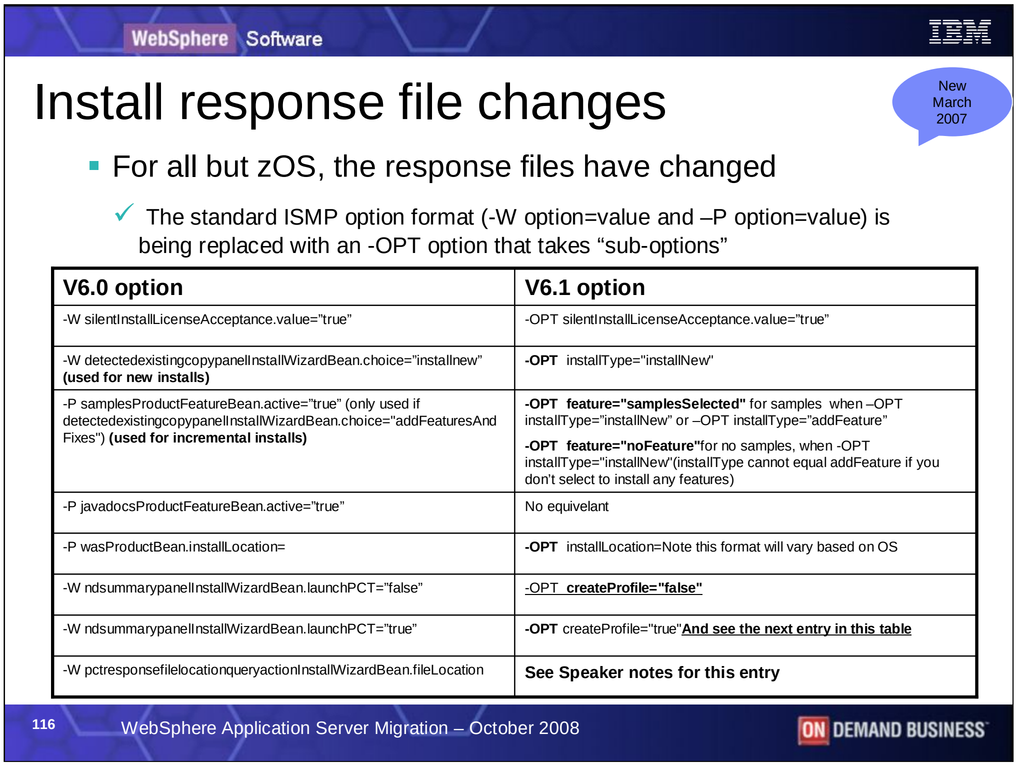
- Reference:
  - http://www-1.ibm.com/support/docview.wss?uid=swg27007527

115 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

WebSphere Application Server v6.0 on Solaris/x86_64 was built as a 32 bit application (32 bit Java, 32 bit JNI). Customers may have their own applications which rely on JNI that is not a part of WAS; for example a special database driver. To run in this environment the JNI must be 32 bit.

WebSphere Application Server v6.1 on Solaris/x86_64 was built as a 64 bit application (64 bit Java, 64 bit JNI). If you have applications on v6.0 that used 32 bit JNI that wasn't a part of WAS those applications cannot be used in v6.1. 32 bit JNI cannot be loaded by a 64 bit JVM. You will have to either rebuild your JNI to be 64 bit, or contact the vendor of that JNI to obtain a 64 bit version.


WebSphere Software
IBM
New
March
2007

Install response file changes

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

V6.0 option	V6.1 option
-W silentInstallLicenseAcceptance.value=”true”	-OPT silentInstallLicenseAcceptance.value=”true”
-W detectedexistingcopypanelInstallWizardBean.choice=”installnew” **(used for new installs)**	**-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 equivelant
-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**

116 WebSphere Application Server Migration – October 2008

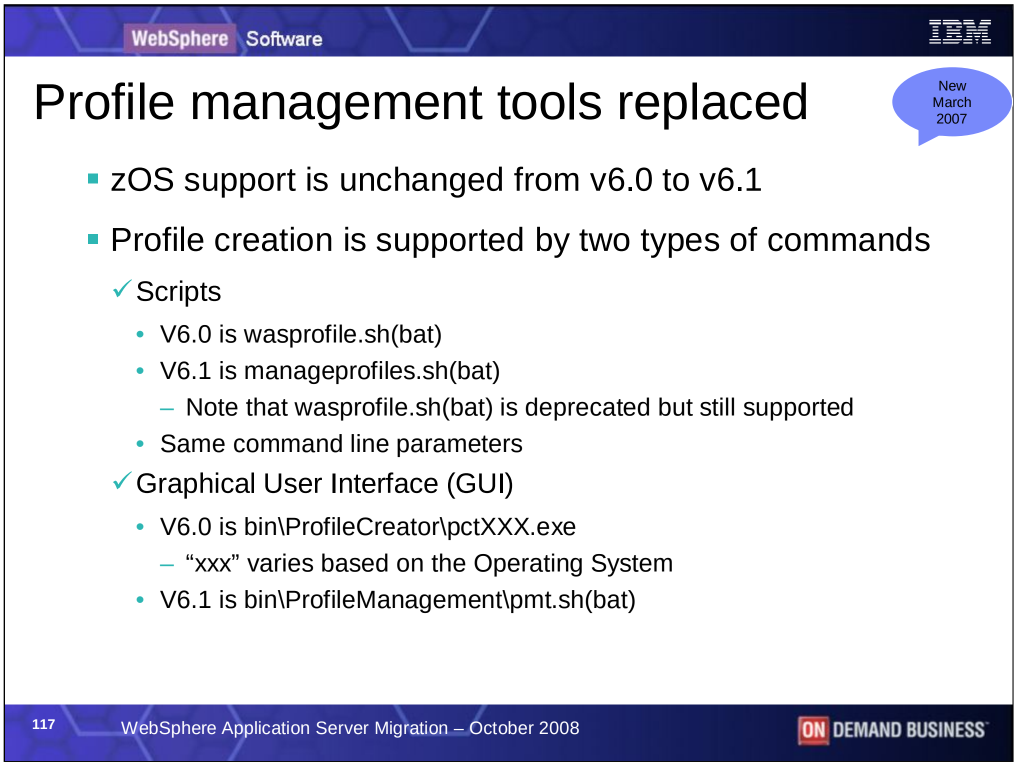
Simpler command line installation are new in Version 6.1. The standard ISMP option format (-W option=value and –P option=value) is being replaced with an -OPT option that takes “sub-options” whose names are simpler and more meaningful, and less subject to change from one release to another. The same options are used in the command line invocation of the wizard and in the silent installation parameters in the response files.

See the following for more information on response files:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/tins_silentn.html

Speaker Note for:” W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation”

If you did launch PCT via install then you used to have a response file. Instead use the following options for profileCreation:profileType, PROF_* - this set will vary based on which profile type you want.Note that all the options for all profiles exist in the one responsefile. Each profile type has its own section, so move to that section when making profile specific changes.

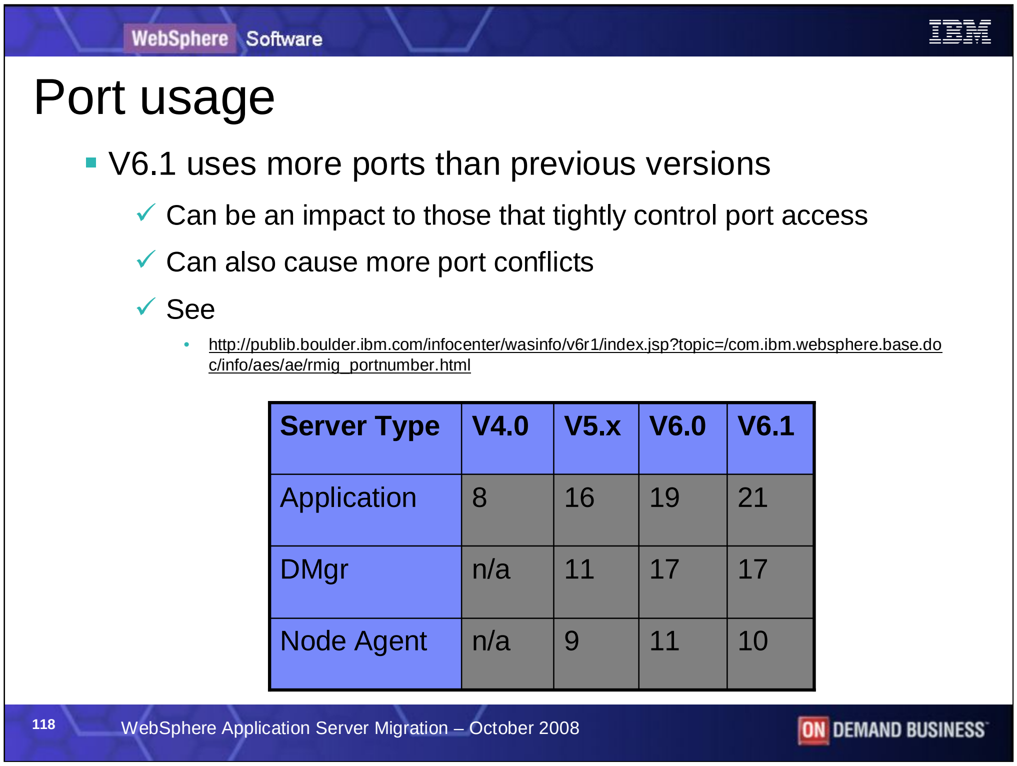


Profile management tools replaced

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)

This is a fairly straight forward mapping as long as you know where to look. The PMT GUI in v6.1 is still for creation only. You will continue to need to use the command line scripts for other tasks, like deletion.



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
  - ✓ See
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rmig_portnumber.html

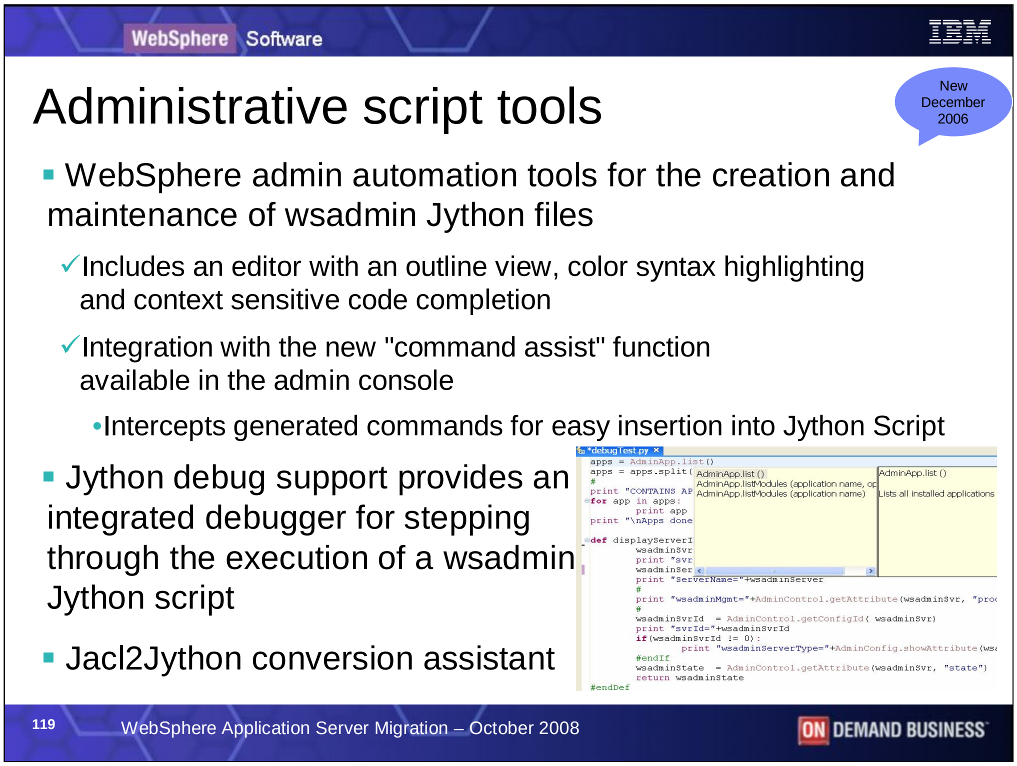
Server Type	**V4.0**	**V5.x**	**V6.0**	**V6.1**
Application	8	16	19	21
DMgr	n/a	11	17	17
Node Agent	n/a	9	11	10

The number of ports reserved out of the box for each successive version of WebSphere Application Server has been steadily increasing due to new feature requirements or changes in existing features.

This increase in the number of ports being reserved can have an impact on your environment in the following ways:

- Your Firewall will need to be updated to reflect these additional ports.
- More ports means there is more potential for port conflicts within WebSphere Application Server and with external services residing on the same ports.

The link provided in this slide gives you a summation of the reserved ports for the different versions of WebSphere Application Server. This summation can be used to give you a rough idea of the port ranges that will be in use out of the box, but note these ports may vary slightly when dealing with a migrated environment.



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

```
apps = AdminApp.list()
apps = apps.split(
#
print "CONTAINS AP
for app in apps:
        print app
print "\nApps done

def displayServerI
        wsadminSvr
        print "svr
        wsadminSer
        print "ServerName="+wsadminServer
        #
        print "wsadminMgmt="+AdminControl.getAttribute(wsadminSvr, "proc
        #
        wsadminSvrId  = AdminControl.getConfigId( wsadminSvr)
        print "svrId="+wsadminSvrId
        if(wsadminSvrId != 0):
                print "wsadminServerType="+AdminConfig.showAttribute(ws
        #endIf
        wsadminState  = AdminControl.getAttribute(wsadminSvr, "state")
        return wsadminState
#endDef
```

AdminApp.list ()
AdminApp.listModules (application name, op
AdminApp.listModules (application name)
AdminApp.list ()
Lists all installed applications

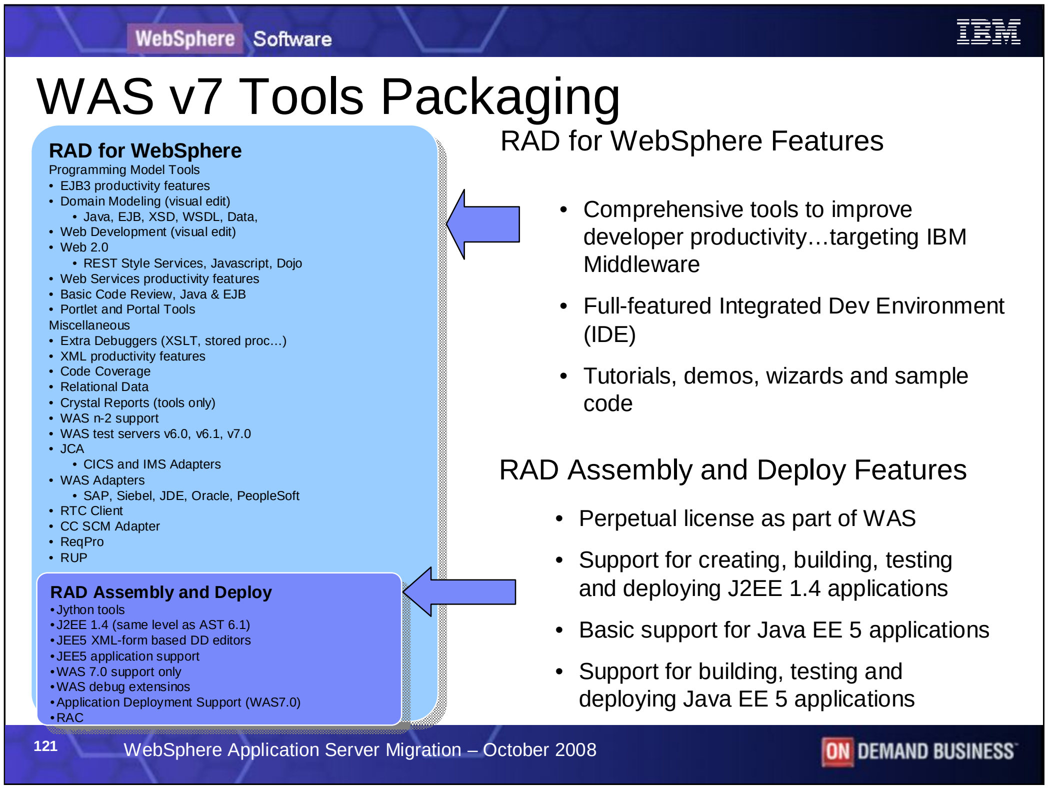
AST v6.1 offers assistance with creating your Jython scripts. This assistance is done in much the same way as prior java development assistance tools(syntax highlighting, auto completion…). WebSphere Application Server is moving in the direction of Jython, so any future scripts you may write should be done in Jython and not jacl. There currently are no plans to drop support for jacl, so there is no need to rush out and change your scripts. There are many reasons for this change, among those are performance and industry support. If you decide that you are going to change your existing scripts then it is worth looking into the “Jacl2Jython” tool which will automatically convert most of your jacl code to jython code and if it cannot convert something it will document that so that you can revisit later. This tool can also be used as a teaching aid for those that only know jacl(ex: create your script in jacl and run the tool and inspect/study the mappings). All in all this tool is pretty slick and worth trying out before manual attempts are made.

This tool can be found at the following link: http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012144

WebSphere Software

v7.0 specific changes

October 2008



WAS v7 Tools Packaging

RAD for WebSphere

Programming Model Tools
- EJB3 productivity features
- Domain Modeling (visual edit)
  - Java, EJB, XSD, WSDL, Data,
- Web Development (visual edit)
- Web 2.0
  - REST Style Services, Javascript, Dojo
- Web Services productivity features
- Basic Code Review, Java & EJB
- Portlet and Portal Tools

Miscellaneous
- Extra Debuggers (XSLT, stored proc…)
- XML productivity features
- Code Coverage
- Relational Data
- Crystal Reports (tools only)
- WAS n-2 support
- WAS test servers v6.0, v6.1, v7.0
- JCA
  - CICS and IMS Adapters
- WAS Adapters
  - SAP, Siebel, JDE, Oracle, PeopleSoft
- RTC Client
- CC SCM Adapter
- ReqPro
- RUP

RAD Assembly and Deploy
- Jython tools
- J2EE 1.4 (same level as AST 6.1)
- JEE5 XML-form based DD editors
- JEE5 application support
- WAS 7.0 support only
- WAS debug extensinos
- Application Deployment Support (WAS7.0)
- RAC

RAD for WebSphere Features

- 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 Features

- 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

“RAD for WebSphere” will be generally available at the same time as WAS V7

RAD for Websphere content will be included with WAS V7, named “RAD Assembly and Deploy module for WebSphere”

RAD Assembly and Deploy content is fully licensed and supported with WAS license

Remaining content is optionally installable as a 60-day trial and becomes perpetually licensed on purchase of RAD for Websphere or priced modules

RAD Assembly and deploy includes:
- J2EE 1.4 support
- Java EE 5 support
- Java EE 5 XML form-based DD editors
- WAS V7 support only
- WAS debug extensions
- Application deployment to WAS V7
- SIP tools

RAD for WebSphere includes:
- Visual editing
- EJB3 and web services productivity features
- WAS 6.0 & 6.1 support
- Adapters: CICS, IMS, SAP, Siebel, JDE, Oracle, PeopleSoft
- Much more

“RAD for WebSphere” will be generally available at the same time as WAS V7

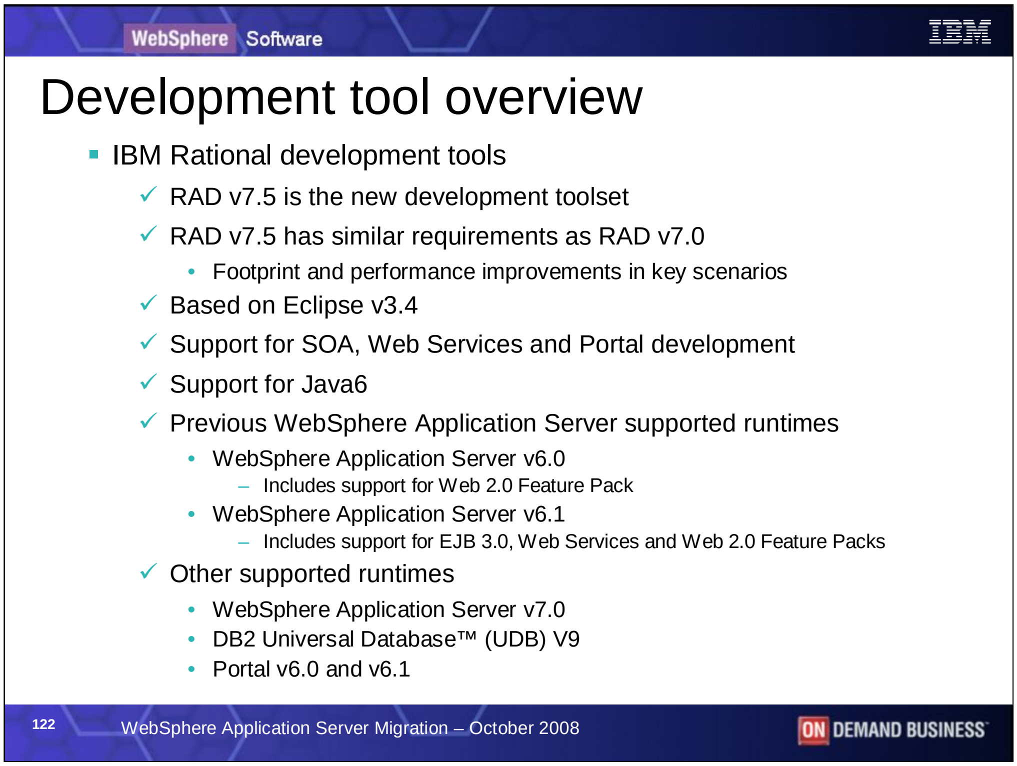
RAD for Websphere content will be included with WAS V7, named “RAD Assembly and Deploy module for WebSphere”

RAD Assembly and Deploy content is fully licensed and supported with WAS license

Remaining content is optionally installable as a 60-day trial and becomes perpetually licensed on purchase of RAD for Websphere or priced modules

RAD Assembly and deploy includes:
- J2EE 1.4 support
- Java EE 5 support
- Java EE 5 XML form-based DD editors
- WAS V7 support only
- WAS debug extensions
- Application deployment to WAS V7
- SIP tools

RAD for WebSphere includes:
- Visual editing
- EJB3 and web services productivity features
- WAS 6.0 & 6.1 support
- Adapters: CICS, IMS, SAP, Siebel, JDE, Oracle, PeopleSoft
- Much more



Development tool overview

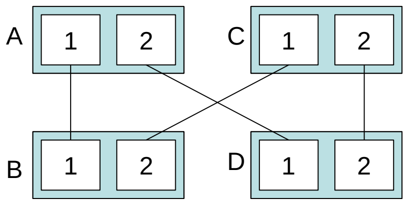
- IBM Rational development tools
  - ✓ RAD v7.5 is the new development toolset
  - ✓ RAD v7.5 has similar requirements as RAD v7.0
    - Footprint and performance improvements in key scenarios
  - ✓ Based on Eclipse v3.4
  - ✓ Support for SOA, Web Services and Portal development
  - ✓ Support for Java6
  - ✓ 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

RAD v7.5 is the newest development toolset designed to work with the latest WebSphere Application Server v7.0.

SIP Migration steps

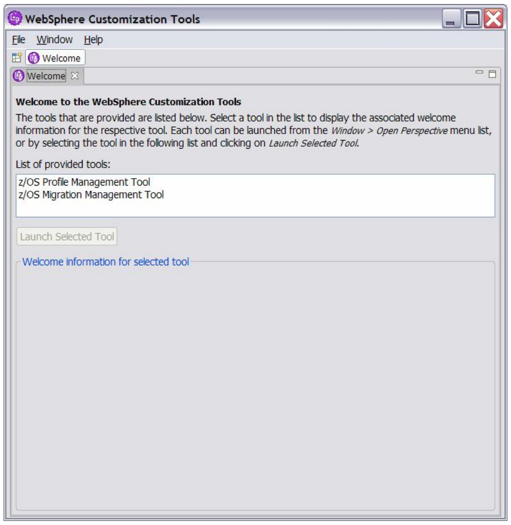
1. Target node A for migration. Quiesce A1, A2 and their replication partners B1 and D1.
   - This leaves node A ready for migration because it has no replication partners active (so we won't hit our replication bytes versioning issue)
   - Stop those 4 servers once quiesce is complete
   - Migrate node 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

Initially SIP will require additional steps for live migration



z/OS Migration Management Tool

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


WebSphere Customization Tools
File
Window
Help
Welcome
Welcome
Launch Selected Tool
Welcome information for selected tool

Welcome to the WebSphere Customization Tools

The tools that are provided are listed below. Select a tool in the list to display the associated welcome information for the respective tool. Each tool can be launched from the *Window > Open Perspective* menu list, or by selecting the tool in the following list and clicking on *Launch Selected Tool.*

List of provided tools:

z/OS Profile Management Tool
z/OS Migration Management Tool

This support was added as an additional Migration path when migrating zOS configurations. This support is available with Application Server Toolkit (AST) 6.1.1 or later.

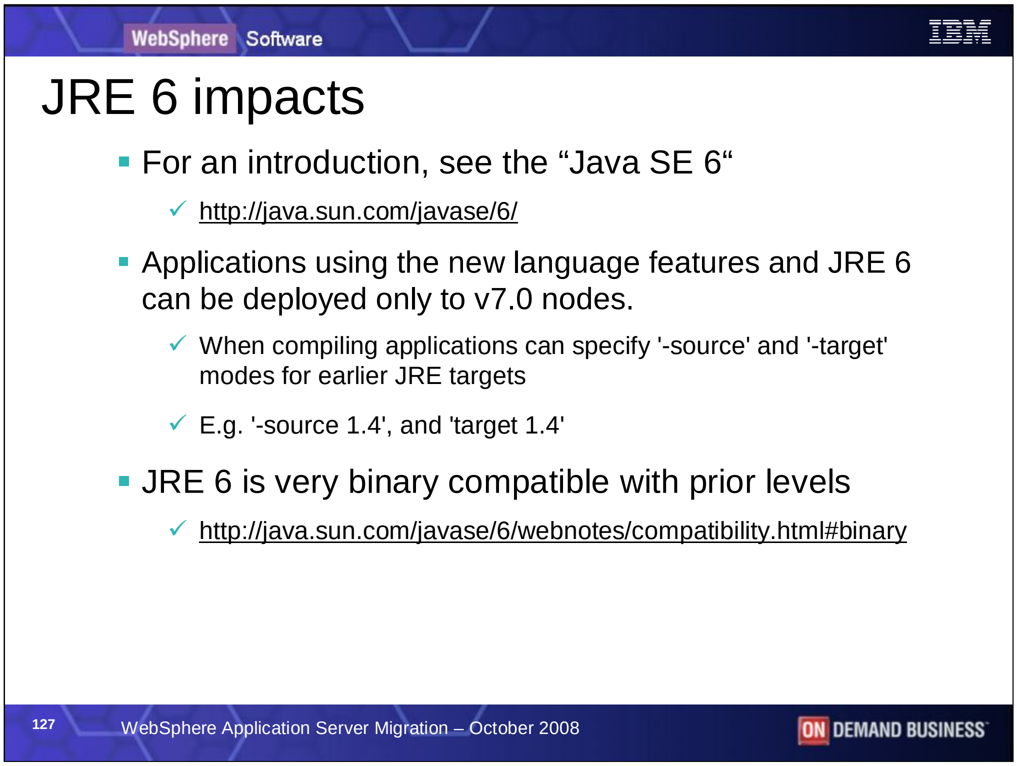
WebSphere Software

v7.0 cumulative changes

October 2008

v7.0 cumulative

Moving to JRE 6

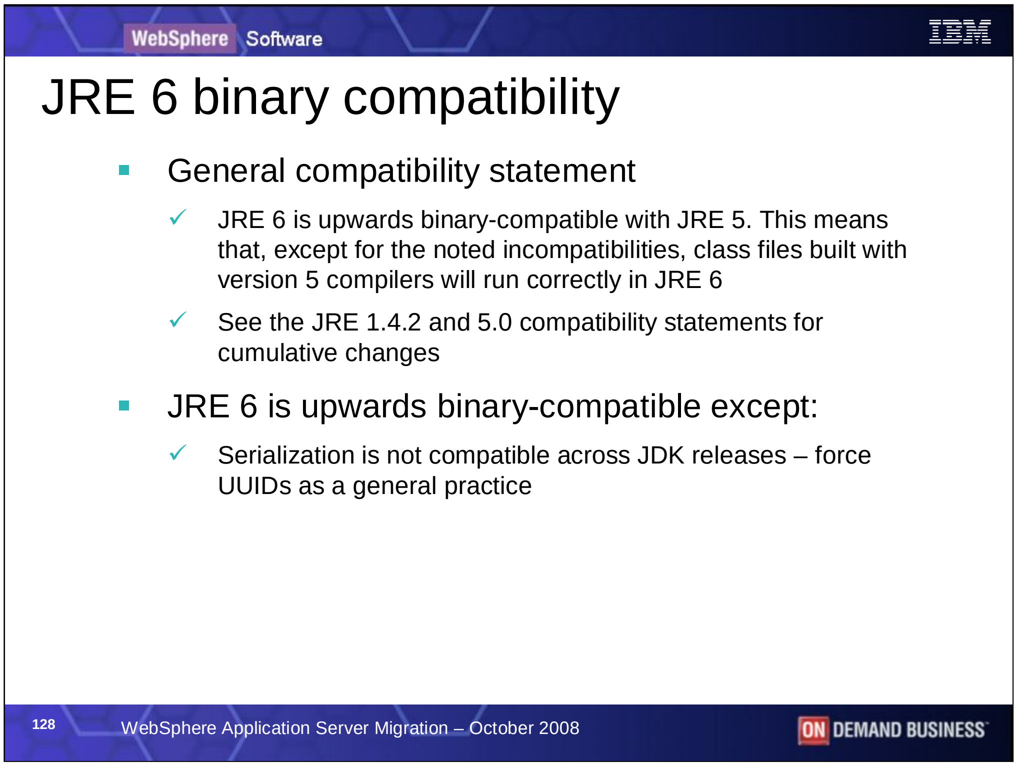

WebSphere Software
IBM

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'
- JRE 6 is very binary compatible with prior levels
  - ✓ http://java.sun.com/javase/6/webnotes/compatibility.html#binary

J2SE 6 introduce a number of new features and WebSphere Application Server v7.0 recognizes that you may want to make use of these features so added support for J2SE 6. But keep in mind that any applications that make use of these new language features can be deployed only to v7.0 nodes. This details needs to be taken into account during your migration planning if you are planning for a extended period of time where you are in a mixed cell state. If you have no intentions of using the new features and running these applications on v6.X/v5.1 nodes is part of your plan then you need to take special steps when compiling your applications(need to set –source and –target).

In general JRE levels are very compatible with prior levels and JRE 6 is no different. There are however non compatible issues between the versions that you need to be aware of and a complete listing of those can be found in the link provided in the slide.



JRE 6 binary compatibility

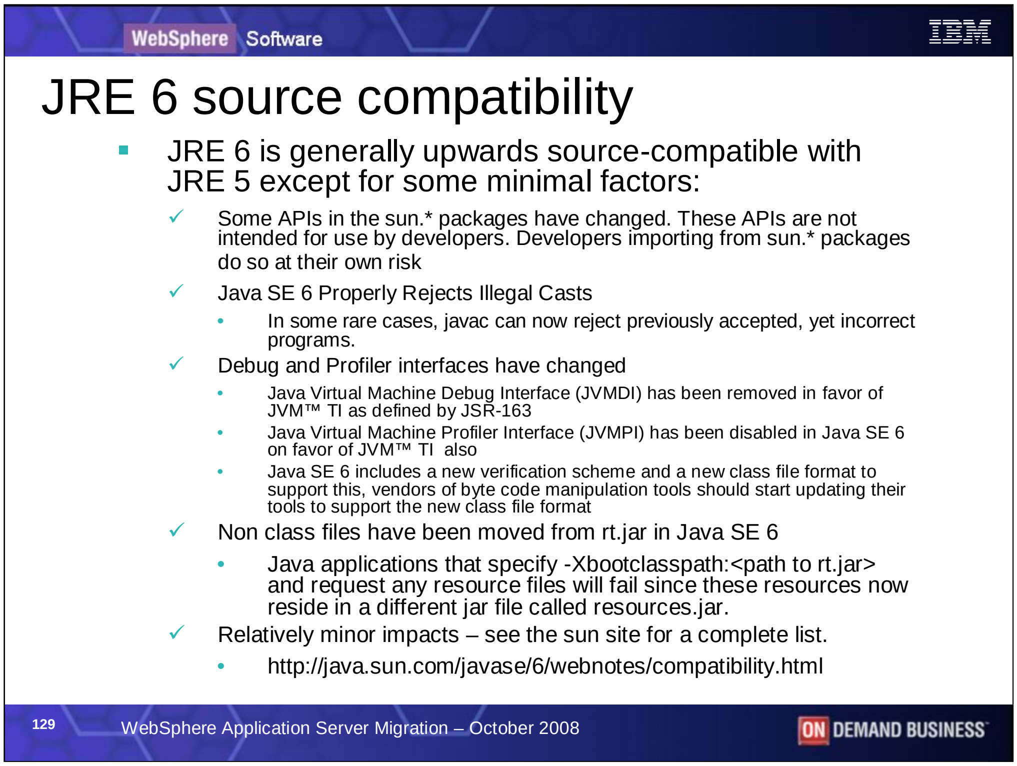
- General compatibility statement
  - ✓ JRE 6 is upwards binary-compatible with JRE 5. This means that, except for the noted incompatibilities, class files built with version 5 compilers will run correctly in JRE 6
  - ✓ See the JRE 1.4.2 and 5.0 compatibility statements for cumulative changes
- JRE 6 is upwards binary-compatible except:
  - ✓ Serialization is not compatible across JDK releases – force UUIDs as a general practice

JRE has a large customer base as you can imagine, so whenever and wherever possible the JRE tries to maintain compatibility. The JRE follows a few guidelines to address this issue. Maintenance releases within a family(example v1.5.1, v1.5.2…) will maintain both upward and downward binary-compatibility with the exception of the defined incompatibles in the specification. Functional releases(example 1.3, 1.4, 1.5) within a family will maintain upward but not necessarily downward binary-compatibility with each other.

The full list of incompatibilities and guidelines can be found at: http://java.sun.com/javase/technologies/compatibility.jsp

Some earlier compilers violated the class format rules specified by the JDK specification. These class files will not run on JDK v5 and must be recompiled.

Differences in the way JRE v6 and prior versions auto generate serial version UIDs will cause incompatibles in your application. To correct this issue you need to explicitly set a serial version UID for each serializable class. This and other breaking potentially breaking changes in JRE 6 can be found at: http://java.sun.com/javase/technologies/compatibility.jsp



JRE 6 source compatibility

- 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 Properly Rejects Illegal Casts
    - In some rare cases, javac can now reject previously accepted, yet incorrect programs.
  - ✓ Debug and Profiler interfaces have changed
    - Java Virtual Machine Debug Interface (JVMDI) has been removed in favor of JVM™ TI as defined by JSR-163
    - Java Virtual Machine Profiler Interface (JVMPI) has been disabled in Java SE 6 on favor of JVM™ TI also
    - Java SE 6 includes a new verification scheme and a new class file format to support this, vendors of byte code manipulation tools should start updating their tools to support the new class file format
  - ✓ 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.
  - ✓ Relatively minor impacts – see the sun site for a complete list.
    - http://java.sun.com/javase/6/webnotes/compatibility.html

JRE has a large customer base as you can imagine, so whenever and wherever possible the JRE tries to maintain compatibility. The JRE follows a few guidelines to address this issue. Maintenance releases do not introduce any new language features or APIs, so they maintain source-compatibility in both directions. Functional releases and major releases maintain upwards but not downwards source-compatibility. There are a few situations where source-compatibility has not been maintained which are defined in the following link: http://java.sun.com/javase/6/webnotes/compatibility.html

One item worth mentioning is that some of the APIs in the sun.* packages have changed. These APIs are private and not intended for use by development. But it is understood that good programming practices are not always followed and thus this should be part of your migration planning.

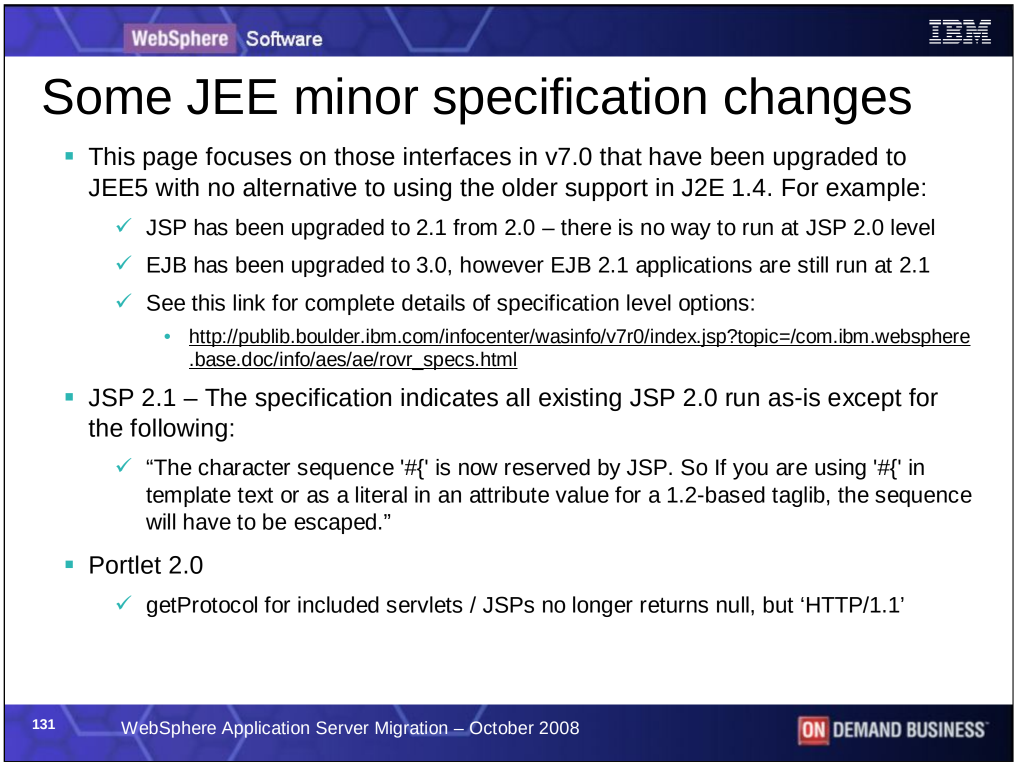
enum is now a java language keyword. If your application is making use enum your application will no longer compile under JDK 6 unless you specify the “-source 1.4” on the javac command.

See this for the JVMPI Transition to JVMTI

http://java.sun.com/developer/technicalArticles/Programming/jvmpitransition/

v7.0 cumulative

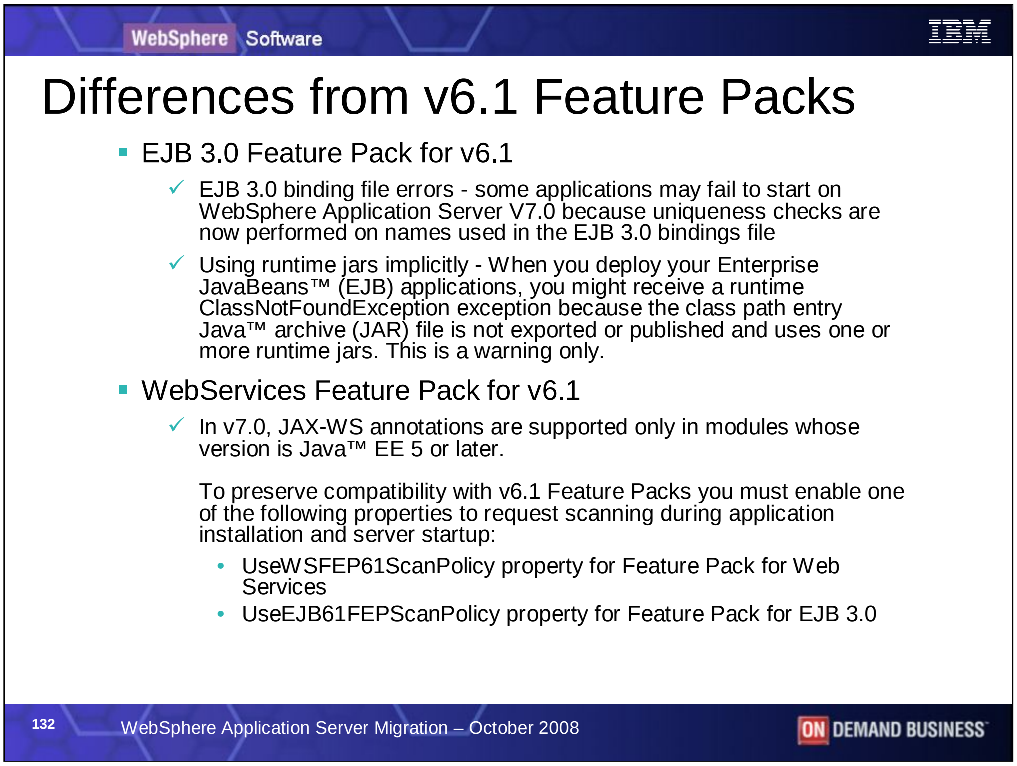
Moving to JEE 5



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
  - ✓ See this link for complete details of specification level options:
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rovr_specs.html
- JSP 2.1 – The specification indicates all existing JSP 2.0 run as-is except for the following:
  - ✓ “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.”
- Portlet 2.0
  - ✓ getProtocol for included servlets / JSPs no longer returns null, but ‘HTTP/1.1’

This page just focuses on APIs where there is no alternative but to use the new version of the specification as noted. There are a number of specifications with newer alternatives in JEE but they are provided as alternatives –not a required move. Information on moving to these new, alternative choices has not been included because this would be considered an upgrade to the application, as opposed to a migration, which is the scope of this document.



Differences from v6.1 Feature Packs

- 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.
- 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.

    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

See the following for more information on the EJB 3.0 items

•http://www-01.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q1=V7RNotes&uid=swg21316530&loc=en_US&cs=utf-8&lang=en

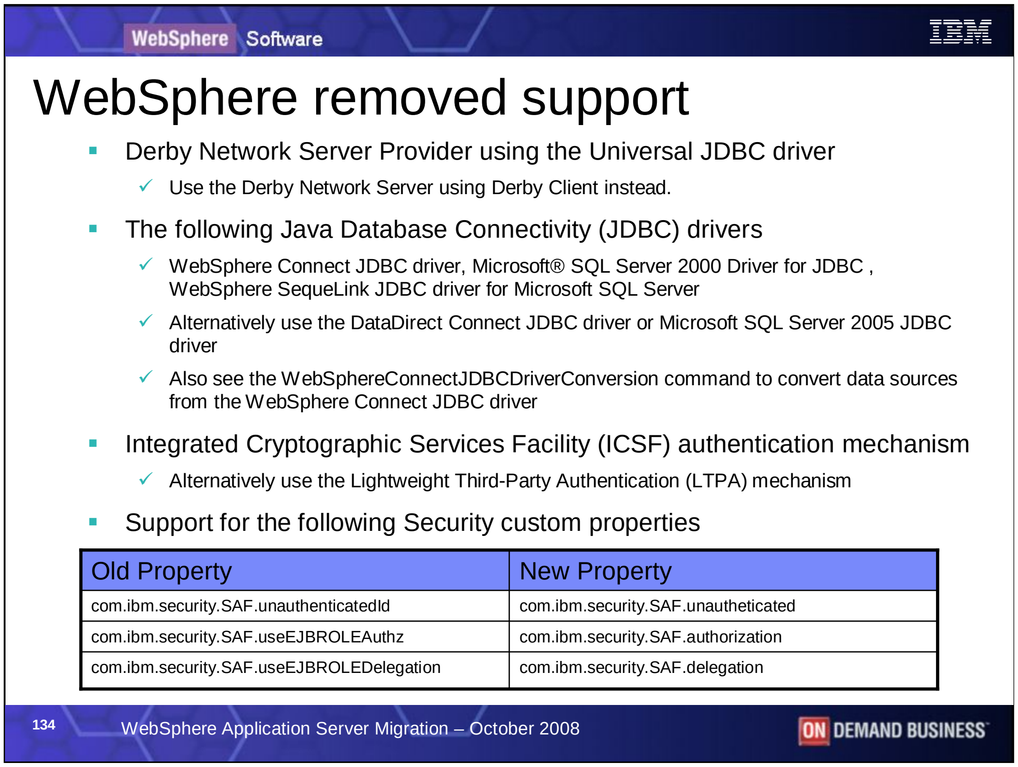
•http://www-01.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q1=V7RNotes&uid=swg21316909&loc=en_US&cs=utf-8&lang=en

See the following for more information on the WebServices item

•http://www-01.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q1=V7RNotes&uid=swg21317708&loc=en_US&cs=utf-8&lang=en

v7.0 cumulative

WebSphere API migration details

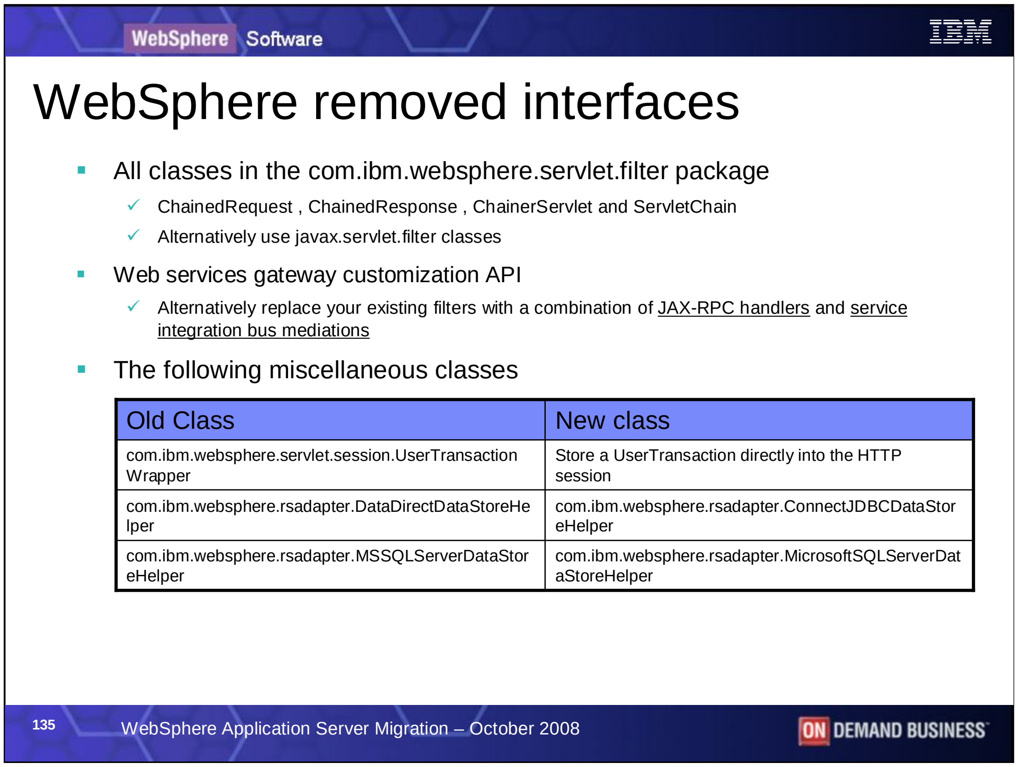


WebSphere removed support

- Derby Network Server Provider using the Universal JDBC driver
  - ✓ Use the Derby Network Server using Derby Client instead.
- 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.unautheticated
com.ibm.security.SAF.useEJBROLEAuthz	com.ibm.security.SAF.authorization
com.ibm.security.SAF.useEJBROLEDelegation	com.ibm.security.SAF.delegation

These features have been removed for the most part using proper WebSphere deprecation processes. See http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_remfeat.html for further information

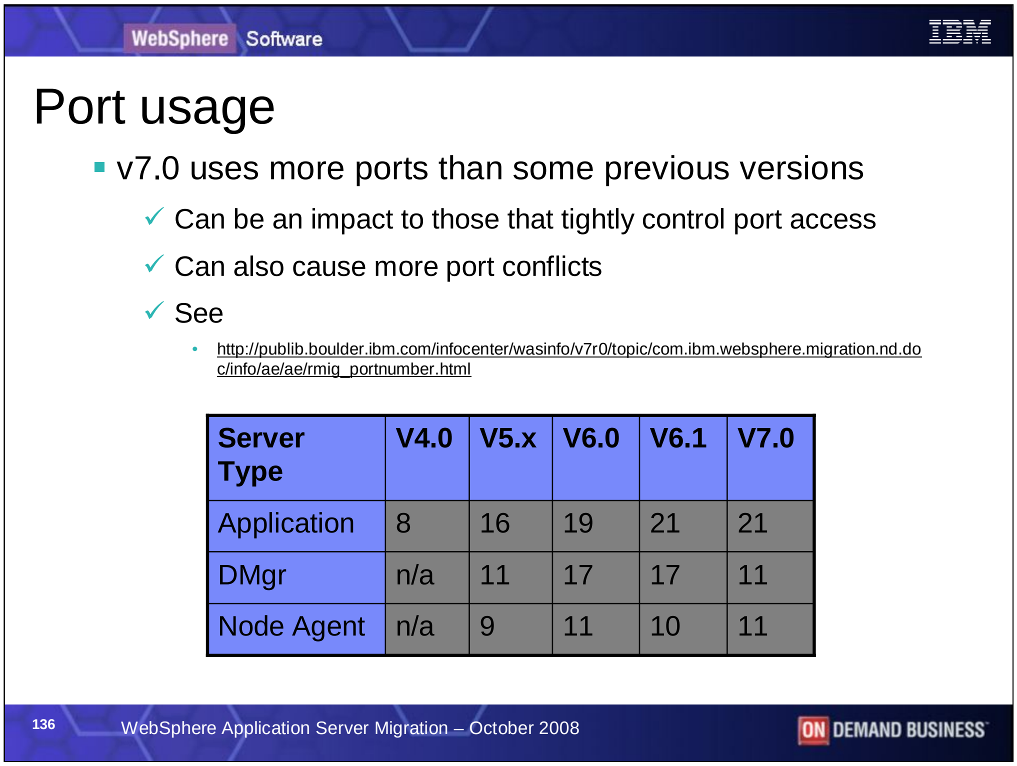


WebSphere removed interfaces

- 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.MSSQLServerDataStoreHelper	com.ibm.websphere.rsadapter.MicrosoftSQLServerDataStoreHelper

These features have been removed for the most part using proper WebSphere deprecation processes. See http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_remfeat.html for further information



WebSphere Software

IBM

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
  - ✓ See
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.migration.nd.doc/info/ae/ae/rmig_portnumber.html

Server Type	V4.0	V5.x	V6.0	V6.1	V7.0
Application	8	16	19	21	21
DMgr	n/a	11	17	17	11
Node Agent	n/a	9	11	10	11

136 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

The number of ports reserved out of the box for each successive version of WebSphere Application Server has been steadily increasing due to new feature requirements or changes in existing features.

This increase in the number of ports being reserved can have an impact on your environment in the following ways:

- •Your Firewall will need to be updated to reflect these additional ports.
- •More ports means there is more potential for port conflicts within WebSphere Application Server and with external services residing on the same ports.

The link provided in this slide gives you a summation of the reserved ports for the different versions of WebSphere Application Server. This summation can be used to give you a rough idea of the port ranges that will be in use out of the box, but note these ports may vary slightly when dealing with a migrated environment.

Security

- 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
    - com.ibm.ws.security.ssoInteropModeEnabled
- WebSeal TAI is no longer supported

WebSphere Software

IBM®

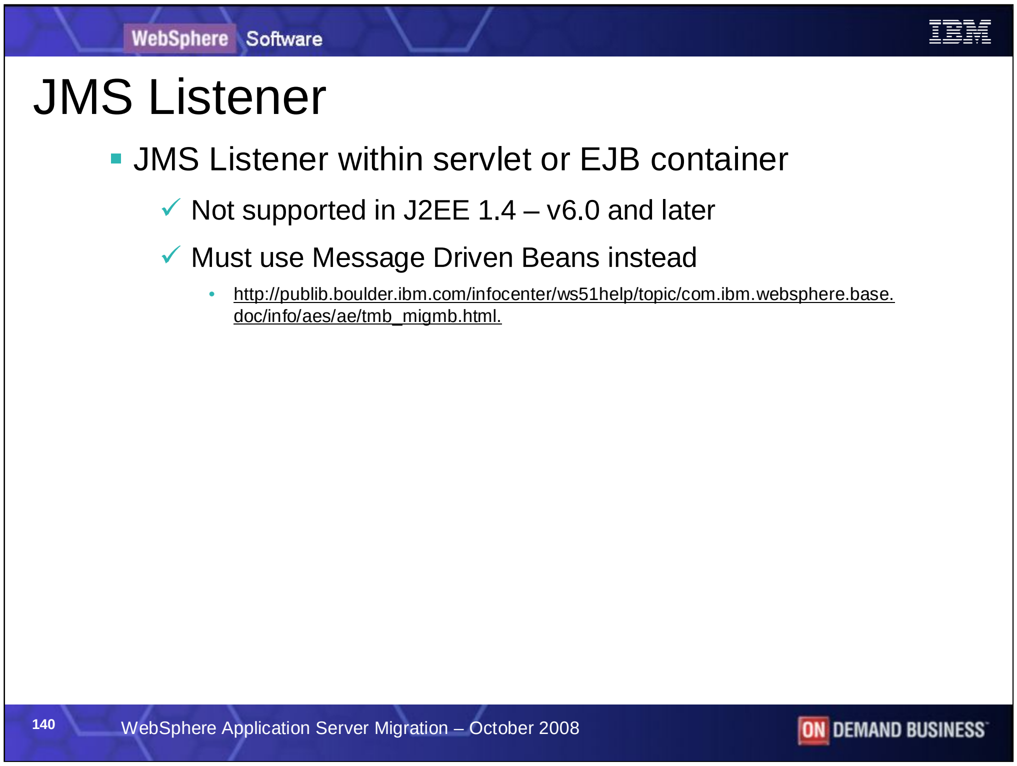
Moving from v4.0

October 2008

ON DEMAND BUSINESS™

Moving from v4.0

Application changes

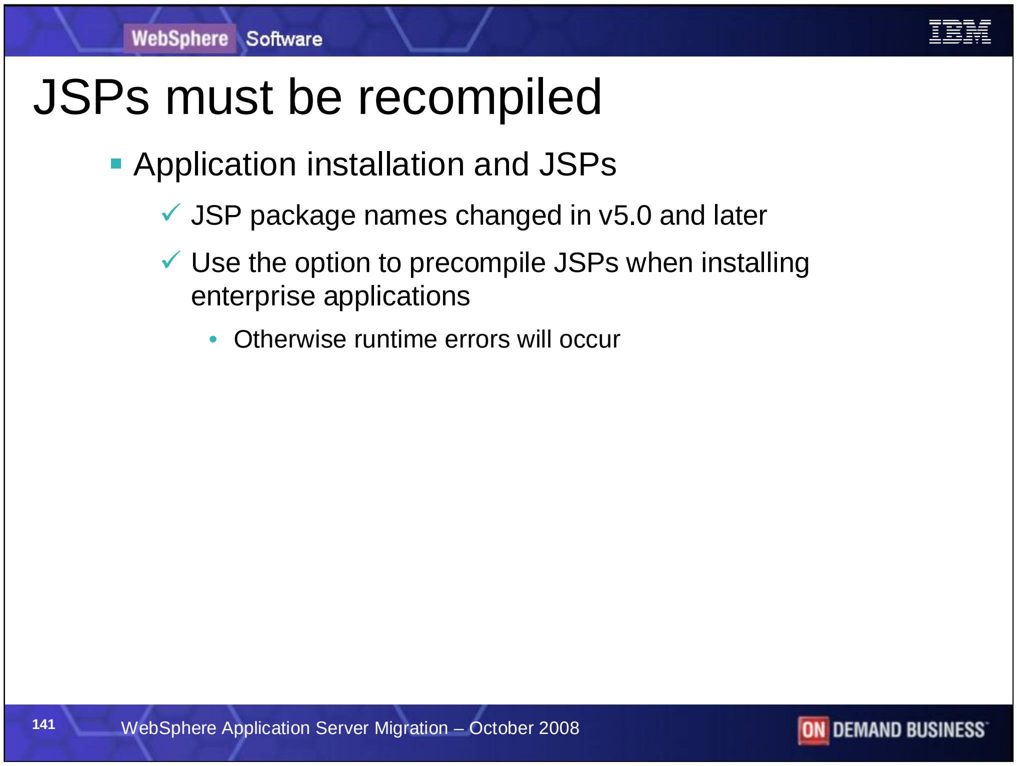


JMS Listener

- JMS Listener within servlet or EJB container
  - ✓ Not supported in J2EE 1.4 – v6.0 and later
  - ✓ Must use Message Driven Beans instead
    - http://publib.boulder.ibm.com/infocenter/ws51help/topic/com.ibm.websphere.base.doc/info/aes/ae/tmb_migmb.html.

Support for JMS Listener applications has been removed in WebSphere Application Server v6.0. You should migrate your application to make use of Message Driven Beans instead. There is a command line utility called “mb2mdb” that can aid you in this conversion. The InfoCenter document linked to in this slide gives you usage instructions for the “mb2mdb” utility.

Further information on developing an deploying Message Driven Beans application can be found in InfoCenter document: http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tmb_devap.html



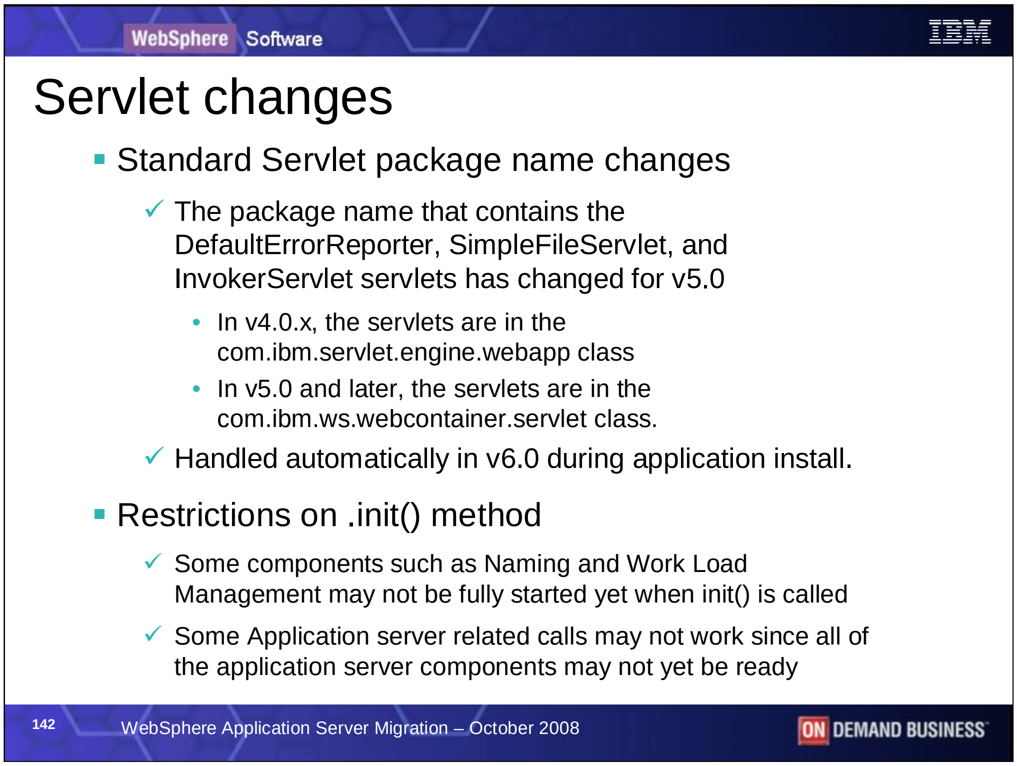
JSPs must be recompiled

- Application installation and JSPs
  - ✓ JSP package names changed in v5.0 and later
  - ✓ Use the option to precompile JSPs when installing enterprise applications
    - Otherwise runtime errors will occur

In WebSphere Application Server v4.0.x, the classes generated from JSP pages are in a package based on the directory structure of the WAR file. Any JSP at the top of the context root is in the unnamed package. JSP pages in subdirectories of the root are in packages named after the subdirectories. In v5.x, the classes generated from JSP pages are all in the org.apache.jsp package. Therefore, the class files are not compatible between versions.

When migrating an enterprise application from v4.0.x to v5.x, recompile the JSP pages to regenerate the class files into the correct packages.

The migration tools(WASPostUpgrade) provide this support, by using the -preCompileJSPs option of the wsadmin tool during the installation of the application. If you are manually reinstalling your applications in v5.x then you will want to use the same option(-preCompileJSPs)



WebSphere Software

IBM

Servlet changes

- Standard Servlet package name changes
  - ✓ The package name that contains the DefaultErrorReporter, SimpleFileServlet, and InvokerServlet servlets has changed for v5.0
    - In v4.0.x, the servlets are in the com.ibm.servlet.engine.webapp class
    - In v5.0 and later, the servlets are in the com.ibm.ws.webcontainer.servlet class.
  - ✓ Handled automatically in v6.0 during application install.
- Restrictions on .init() method
  - ✓ Some components such as Naming and Work Load Management may not be fully started yet when init() is called
  - ✓ Some Application server related calls may not work since all of the application server components may not yet be ready

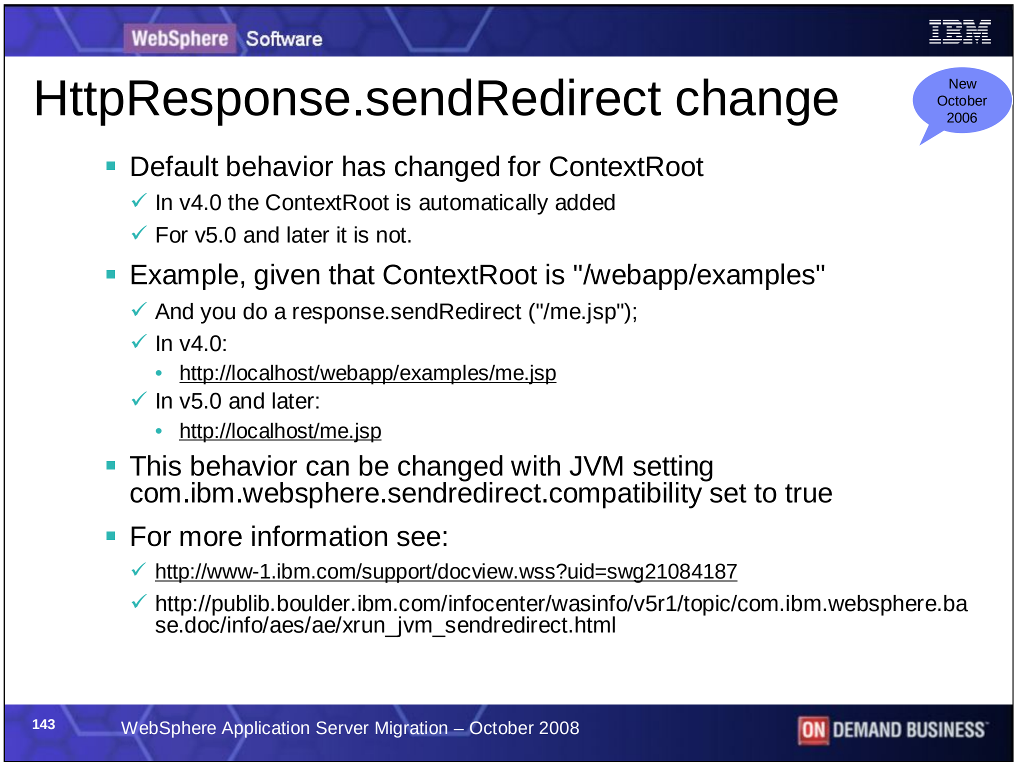
The package that contains the DefaultErrorReporter servlet has changed for WebSphere Application Server v5.x. In v3.5.x and v4.0.x, the servlet was in the com.ibm.servlet.engine.webapp class. In v5.x, the servlet is in the com.ibm.ws.webcontainer.servlet class. Direct use of the DefaultErrorReporter servlet has been deprecated in v5.x and other plans should be made.

The InvokerServlet and SimpleFileServlet servlets are internal servlets that have not been public since WebSphere Application Server v3.5. If you referenced these servlets in any version after v3.5 through the web.xml file, you should remove these entries from the web.xml file and use the ibm-web-ext.xmi file to enable or disable these servlets using serveServletsByClassnameEnabled and fileServingEnabled.

The following InfoCenter document shows an example of this:
http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cins_migconf.html

There are restrictions to keep in mind when depending on the behavior of .init() when it relates to the WebContainer startup. If you enable your servlets to start up when the application server starts up you may run into dependency issues. Naming and Work Load Management services may not be fully initialized during the execution of the .init() methods, so your code should account for this and make other arrangements.

More details on this topic and be found in InfoCenter document:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tovr_migratingjava.html



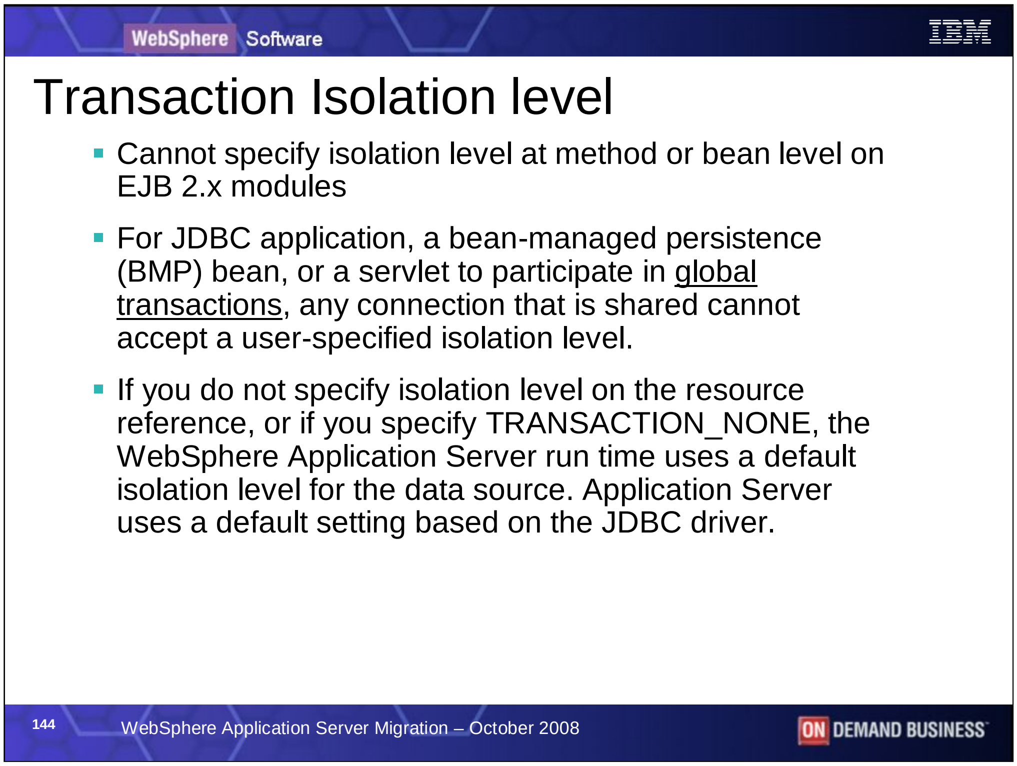
HttpResponse.sendRedirect change

New October 2006

- Default behavior has changed for ContextRoot
  - ✓ In v4.0 the ContextRoot is automatically added
  - ✓ For v5.0 and later it is not.
- Example, given that ContextRoot is "/webapp/examples"
  - ✓ And you do a response.sendRedirect ("/me.jsp");
  - ✓ In v4.0:
    - http://localhost/webapp/examples/me.jsp
  - ✓ In v5.0 and later:
    - http://localhost/me.jsp
- This behavior can be changed with JVM setting com.ibm.websphere.sendredirect.compatibility set to true
- For more information see:
  - ✓ http://www-1.ibm.com/support/docview.wss?uid=swg21084187
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.base.doc/info/aes/ae/xrun_jvm_sendredirect.html

Behavior of HttpResponse.sendRedirect(…) has changed starting in WebSphere Application Server v5.x. In v4, if your .sendRedirect(…) contains a leading “/”, it will append the ContextRoot by default. This default behavior can be overridden by specifying the JVM system property “com.ibm.websphere.sendredirect.compliance” to “true”. In v5.x and later if your .sendRedirect(…) contains a leading “/” it will not append the ContextRoot by default. This behavior change can be reverted back to v4 default of appending the ContextRoot by specifying the JVM system property “com.ibm.websphere.sendredirect.compatibility” to “true”.

The links provided in the slide describe the problem in more detail and show you how to modify the property.



Transaction Isolation level

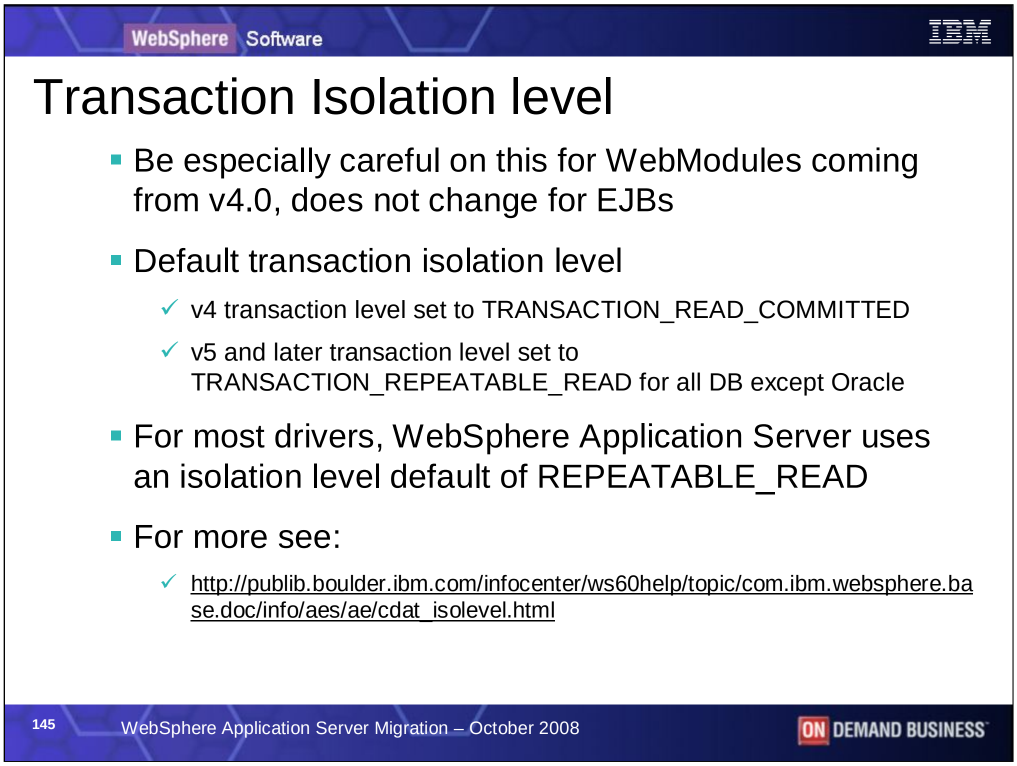
- Cannot specify isolation level at method or bean level on EJB 2.x modules
- For JDBC application, a bean-managed persistence (BMP) bean, or a servlet to participate in global transactions, any connection that is shared cannot accept a user-specified isolation level.
- If you do not specify isolation level on the resource reference, or if you specify TRANSACTION_NONE, the WebSphere Application Server run time uses a default isolation level for the data source. Application Server uses a default setting based on the JDBC driver.

In an Enterprise JavaBean (EJB) 1.1 module, you can set the isolation level at the method level or bean level. This capability also applies to container-managed persistence (CMP) 1.1 beans that you assemble into *EJB 2.x modules*. (WebSphere Application Server permits the deployment descriptor of a CMP bean to declare the version level of 1.1, regardless of the overall module version.) However, the ability to set isolation level at the method or bean level does **not** apply to other enterprise beans within an EJB 2.x module, including *CMP 2.x beans*. WebSphere Application Server Version 5.0 removed this capability from EJB 2.0 modules to deliver an architecture that ultimately provides more efficient connection use.

In a EJB 2.x module, when a CMP 2.x bean uses a new data source to access a backend database, the isolation level is determined by the WebSphere Application Server run time, based on the type of access intent assigned to the bean or the calling method. Other non-CMP connection users can access this same data source and also use the access intent and application profile support to manage their concurrency control.

The product does not require you to set the isolation level on a data source resource reference for a non-CMP application module. If you do not specify isolation level on the resource reference, or if you specify TRANSACTION_NONE, the WebSphere Application Server run time uses a default isolation level for the data source. Application Server uses a default setting based on the JDBC driver. For most drivers, WebSphere Application Server uses an isolation level default of TRANSACTION_REPEATABLE_READ. For Oracle drivers, however, Application Server uses an isolation level of TRANSACTION_READ_COMMITTED.

The following InfoCenter document contains more details on the topics covered in this slide:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cdat_isolevel.html



Transaction Isolation level

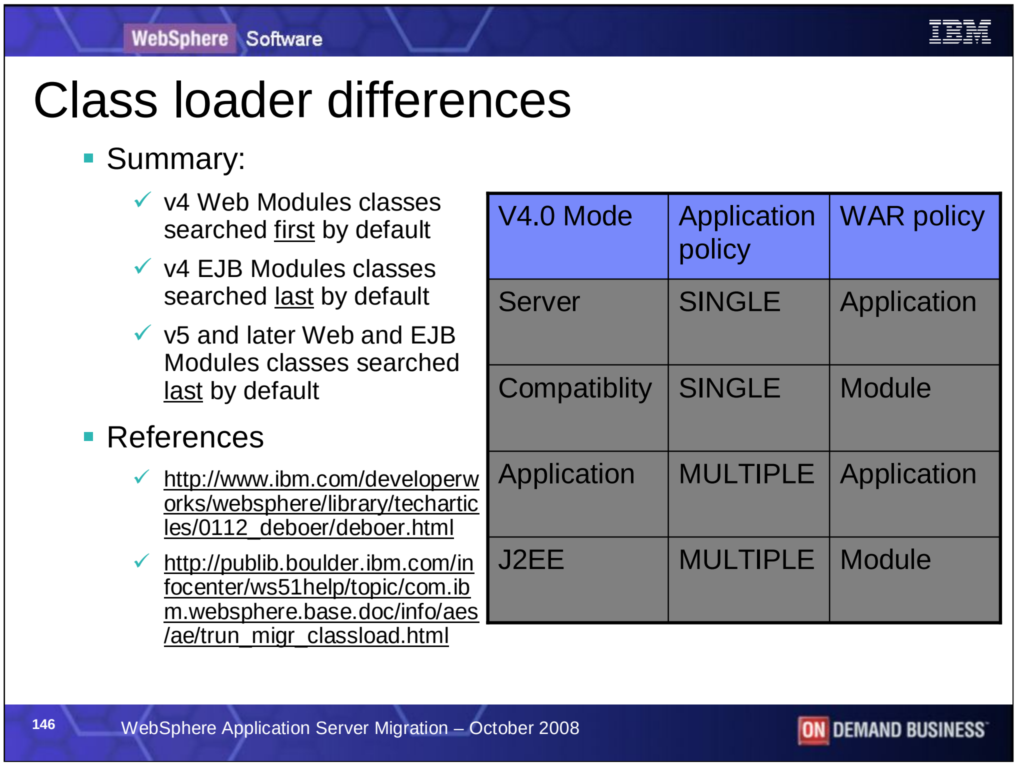
- Be especially careful on this for WebModules coming from v4.0, does not change for EJBs
- Default transaction isolation level
  - ✓ v4 transaction level set to TRANSACTION_READ_COMMITTED
  - ✓ v5 and later transaction level set to TRANSACTION_REPEATABLE_READ for all DB except Oracle
- For most drivers, WebSphere Application Server uses an isolation level default of REPEATABLE_READ
- For more see:
  - ✓ http://publib.boulder.ibm.com/infocenter/ws60help/topic/com.ibm.websphere.base.doc/info/aes/ae/cdat_isolevel.html

Applications may experience a loss of performance or in worst case database deadlocks.

Modify the isolation level of the datasource resource reference in the Application to what your Application expects.

The default transaction isolation levels have changed, between WebSphere Application Server v4.x and v5.x. v4.x transaction isolation level was set to TRANSACTION_READ_COMMITTED and in v5.x and later versions it is set to TRANSACTION_REPEATABLE_READ for all DBs except for Oracle in which it remains TRANSACTION_READ_COMMITTED. Worst case scenario if this information is overlooked or not part of your migration planning is that your applications that previously worked may now end up with deadlocks.

The link to the InfoCenter document found in the slide has additional details.



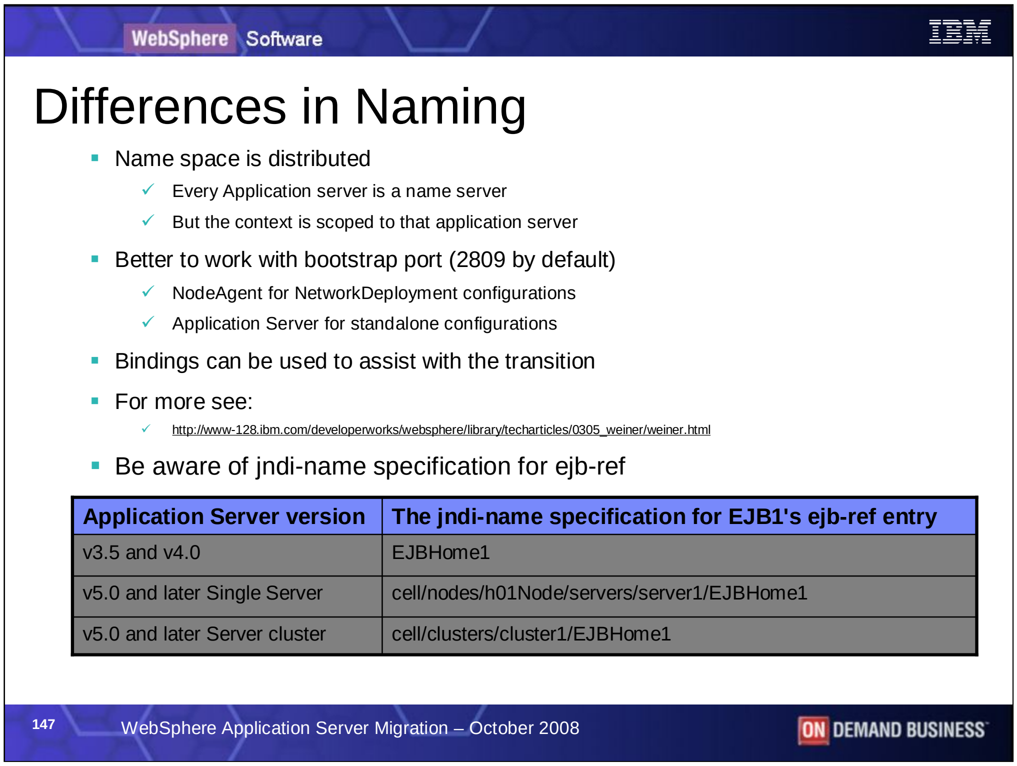
Class loader differences

- Summary:
  - ✓ v4 Web Modules classes searched first by default
  - ✓ v4 EJB Modules classes searched last by default
  - ✓ v5 and later Web and EJB Modules classes searched last by default
- References
  - ✓ http://www.ibm.com/developerworks/websphere/library/techarticles/0112_deboer/deboer.html
  - ✓ http://publib.boulder.ibm.com/infocenter/ws51help/topic/com.ibm.websphere.base.doc/info/aes/ae/trun_migr_classload.html

V4.0 Mode	Application policy	WAR policy
Server	SINGLE	Application
Compatiblity	SINGLE	Module
Application	MULTIPLE	Application
J2EE	MULTIPLE	Module

Classloader default load order has changed between WebSphere Application Server v4.x and v5.x. By default in v4.x Web Modules classes were search first and in v5.x Web Modules classes are search last by default. This change can cause some in your face errors or some odd errors that are hard to debug. For an example, in another slide we stated that JDK started shipping with it’s own XML libraries in v5.x. If your v4.x application includes it’s own XML libraries and you migrate this application to v5.x and do no further action, then your application will no longer be using your XML libraries included in your application, instead it will be using the XML libraries included in the JDK.

There currently there is no J2EE specification that dictates how classloaders will operate, and thus all application servers have their own implementation and each operates differently. The article referenced in the slide does a good job of detailing the WebSphere Application Server classloading architiecture.



Differences in Naming

- Name space is distributed
  - ✓ Every Application server is a name server
  - ✓ But the context is scoped to that application server
- Better to work with bootstrap port (2809 by default)
  - ✓ NodeAgent for NetworkDeployment configurations
  - ✓ Application Server for standalone configurations
- Bindings can be used to assist with the transition
- For more see:
  - ✓ http://www-128.ibm.com/developerworks/websphere/library/techarticles/0305_weiner/weiner.html
- Be aware of jndi-name specification for ejb-ref

Application Server version	The jndi-name specification for EJB1's ejb-ref entry
v3.5 and v4.0	EJBHome1
v5.0 and later Single Server	cell/nodes/h01Node/servers/server1/EJBHome1
v5.0 and later Server cluster	cell/clusters/cluster1/EJBHome1

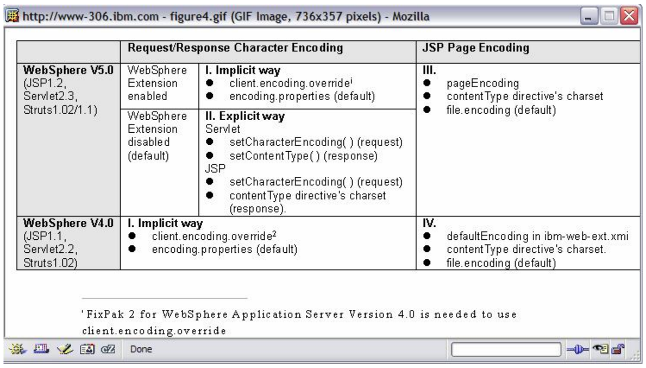
The Naming service in WebSphere Application Server v5x and above is implemented as a distributed Naming service. What this means is that all server processes(servers, nodeagents, DMgr) have a Naming service running and all these Naming services are linked together to form a distributed Naming service. This is in contrast to the way v3.5 and v4.x implemented the Naming service, which was a flat Naming service(all servers running in a single InitialContext). This difference can be seen in the table in the slide which shows that the “jndi-name” in v3.5/v4.x was just EJBHome1 and in v5.x and greater it is located in a specific IntialContext(ex: cell/nodes/h01Node/servers/server1/EJBHome1).

The article linked to in this slide goes into greater detail on the topics provided in this slide.

Webcontainer auto-encoding change

New
May
2008

- In v4.0 automatically sets request and response encodings, and response content types
- In v5.0 and later need to set these values using available methods in the Servlet 2.3 Specification or later.
  - ✓ Or you can specify the autoRequestEncoding and autoResponseEncoding extensions
- For more information see:
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cweb_autoreq.html
  - ✓ http://www.ibm.com/software/globalization/j2ee/encoding.jsp


http://www-306.ibm.com - figure4.gif (GIF Image, 736x357 pixels) - Mozilla
Done

	Request/Response Character Encoding		JSP Page Encoding
WebSphere V5.0 (JSP1.2, Servlet2.3, Struts1.02/1.1)	WebSphere Extension enabled	**I. Implicit way** ● client.encoding.override[1] ● encoding.properties (default)	**III.** ● pageEncoding ● contentType directive's charset ● file.encoding (default)
	WebSphere Extension disabled (default)	**II. Explicit way** Servlet ● setCharacterEncoding( ) (request) ● setContentType( ) (response) JSP ● setCharacterEncoding( ) (request) ● contentType directive's charset (response).	
WebSphere V4.0 (JSP1.1, Servlet2.2, Struts1.02)	**I. Implicit way** ● client.encoding.override[2] ● encoding.properties (default)		**IV.** ● defaultEncoding in ibm-web-ext.xmi ● contentType directive's charset. ● file.encoding (default)

---

[1] FixPak 2 for WebSphere Application Server Version 4.0 is needed to use client.encoding.override

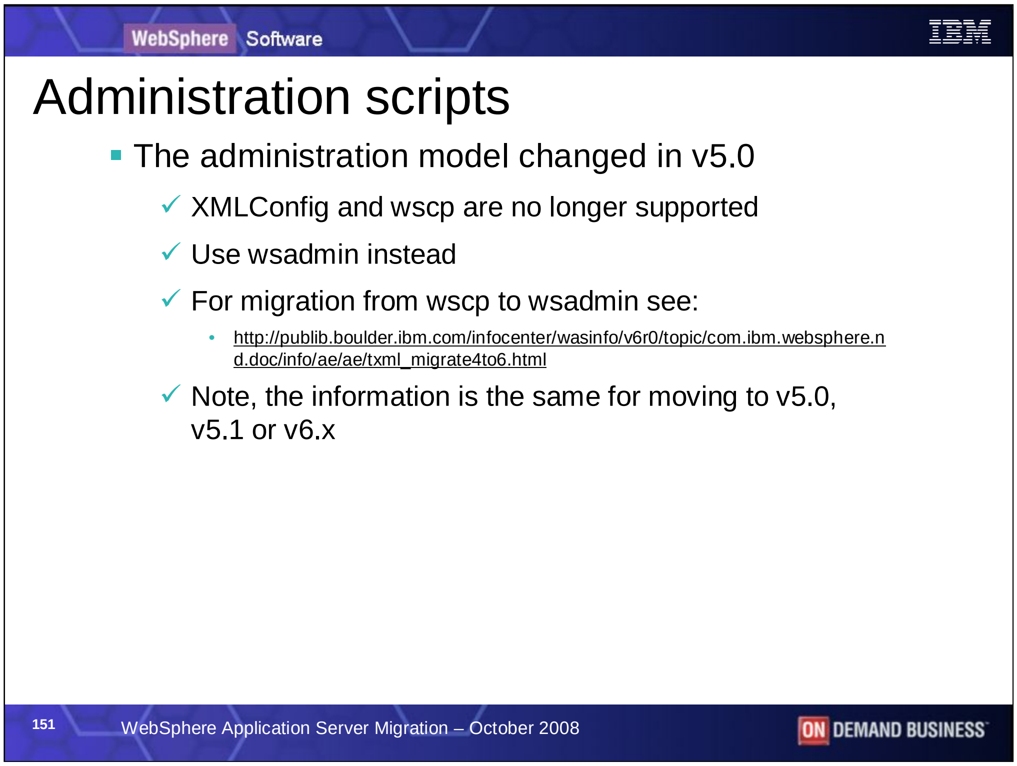
See the following for a more detailed explanation:

References

- Transaction Isolation level change
  - http://publib.boulder.ibm.com/infocenter/ws60help/topic/com.ibm.websphere.base.doc/info/aes/ae/cdat_isolevel.html
- Class loader behavioral difference
  - www.ibm.com/developerworks/websphere/ library/techarticles/0112_deboer/deboer.html
  - http://publib.boulder.ibm.com/infocenter/ws51help/topic/com.ibm.websphere.base.doc/info/aes/ae/trun_migr_classload.htm
- Connection sharing and new datasource sharing
  - http://publib.boulder.ibm.com/infocenter/wasinfo/topic/com.ibm.websphere.base.doc/info/aes/ae/rdat_conmig.html
  - http://www-1.ibm.com/support/docview.wss?rs=0&q1=ConnectionWaitTimeoutException&uid=swg21192666&loc=en_US&cs=utf-8&cc=us&lang=en

Moving from v4.0

Administration changes

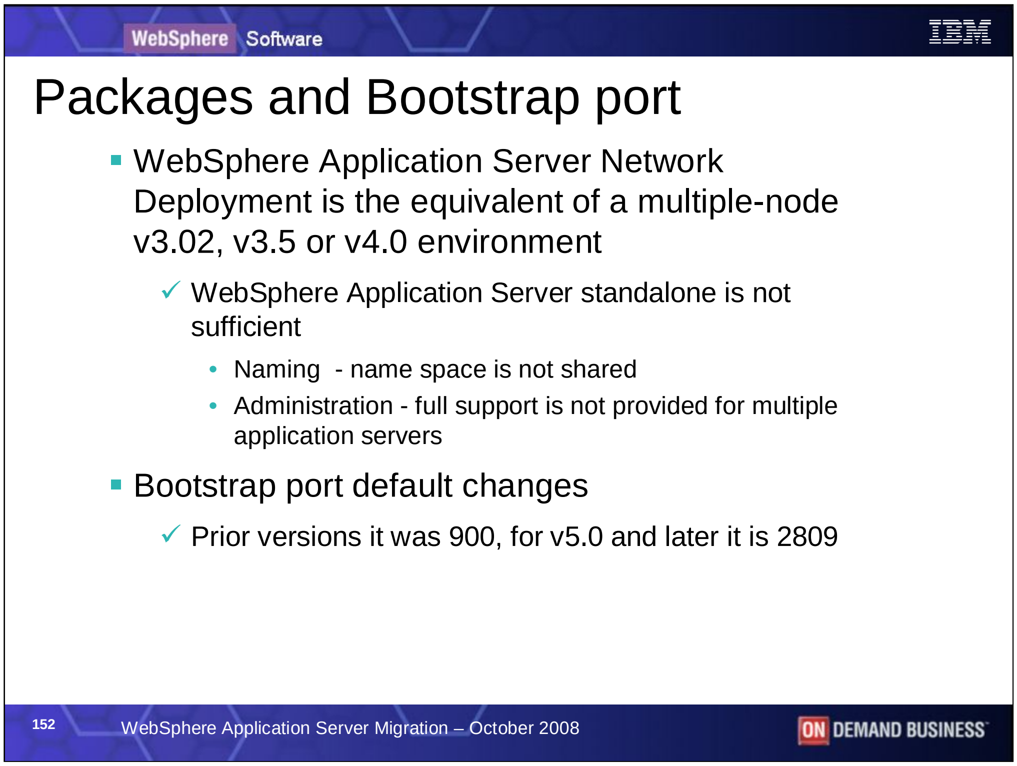


Administration scripts

- The administration model changed in v5.0
  - ✓ XMLConfig and wscp are no longer supported
  - ✓ Use wsadmin instead
  - ✓ For migration from wscp to wsadmin see:
    - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate4to6.html
  - ✓ Note, the information is the same for moving to v5.0, v5.1 or v6.x

The repository that was implemented in WebSphere Application Server v3.x and later no longer exists and thus the tools that manipulate it are no longer supported. wsadmin is a replacement for wscp. Whatever you can do in wscp you can do in wsadmin, but the same does not hold true the other way around.

In v4.0, wscp commands are used for both configuration queries or updates, and operational commands. In wsadmin, a distinction is made between configuration and operational commands. The link provided in the slide contains a table of mappings from wscp commands to wsadmin commands that can be used in your migration.



WebSphere Software

Packages and Bootstrap port

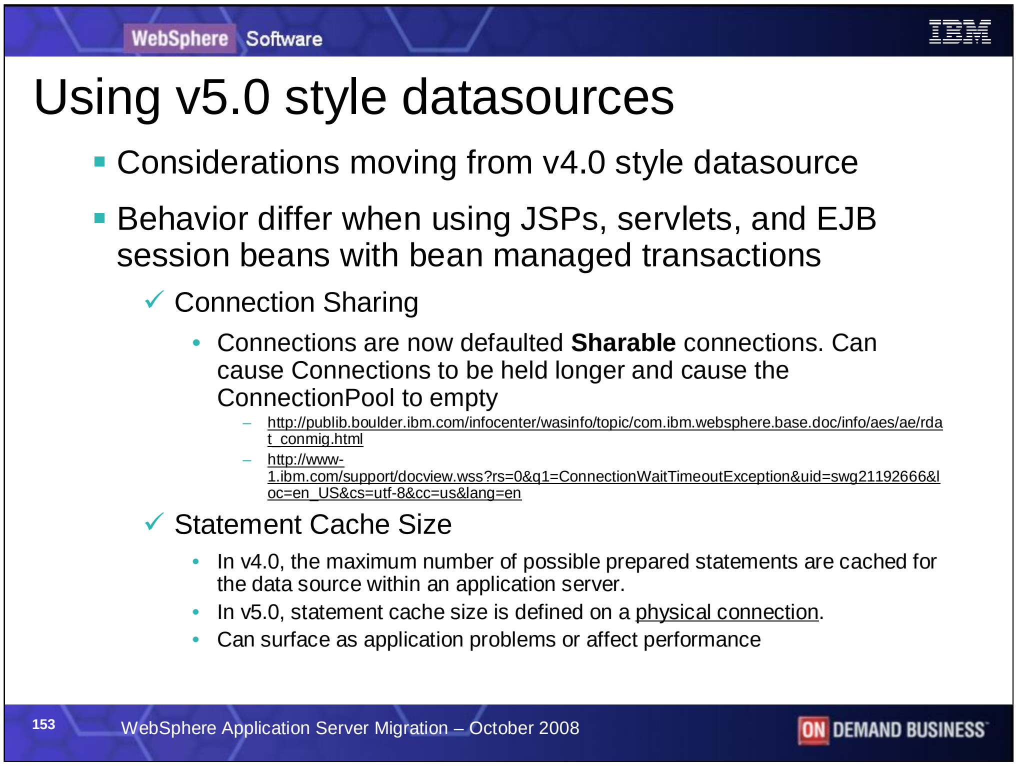
- WebSphere Application Server Network Deployment is the equivalent of a multiple-node v3.02, v3.5 or v4.0 environment
  - ✓ WebSphere Application Server standalone is not sufficient
    - Naming - name space is not shared
    - Administration - full support is not provided for multiple application servers
- Bootstrap port default changes
  - ✓ Prior versions it was 900, for v5.0 and later it is 2809

152 WebSphere Application Server Migration – October 2008

During your migration planning you need to determine which package of WebSphere Application Server will be sufficient for your needs. In WebSphere Application Server v5x and later there are 2 main packages for you to consider(There are various flavors of these packages that you should investigate as well to determine your needs), which are WebSphere Application Server Network Deployment and WebSphere Application Server Standalone. If your v3.02/v3.5/v4.0 environment consists of more than one node then your migration path should take you to WebSphere Application Server Network Deployment. If your v3.0.2/v3.5/v4.0 environment consists of only one node then your migration path should take you to WebSphere Application Server Standalone.

A couple of the big differences between Network Deployment and Standalone is that in Standalone the name space is not shared and you do not have full administrative support for multiple application servers. What this means for you is that your applications cannot share resources via the Naming Service and that you cannot centrally manage your application servers.

Another Naming-related change in v5.0 and later is that the default bootstrap port has changed from 900 to 2809. This change was implemented as part of the CORBA Interoperable Naming Service (INS) support, which is required for J2EE 1.3 compliance. Since the provider URL is already set in server environments, then this change should primarily impact the launching of pure clients, where a provider URL specification is most likely needed.



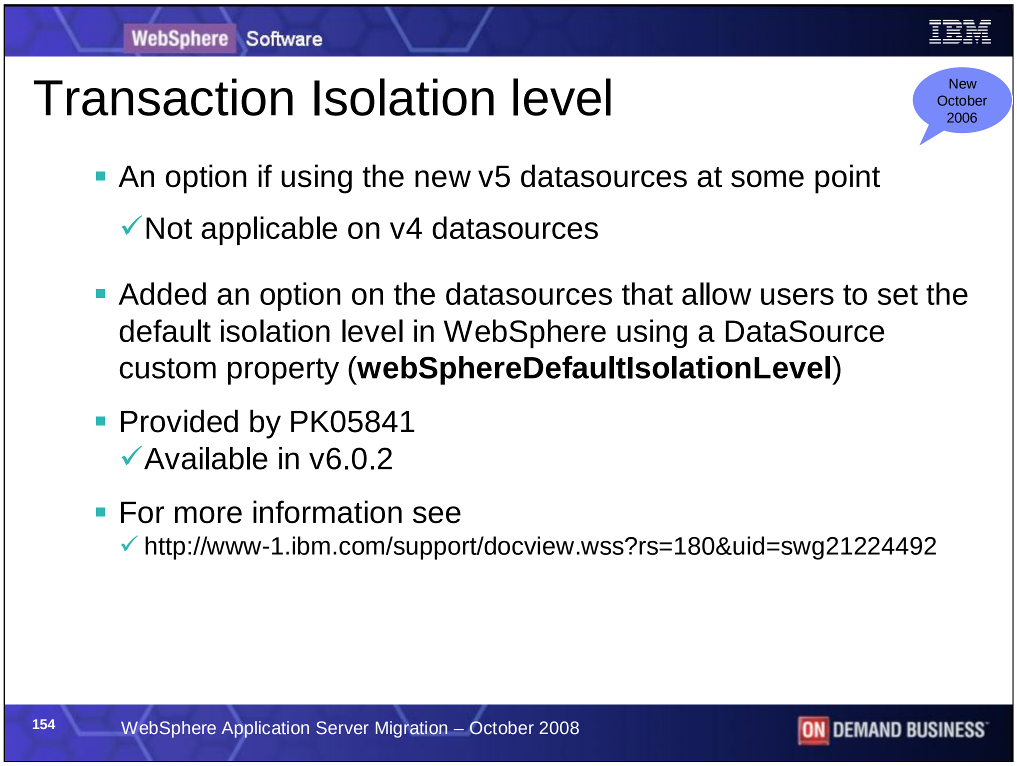
Using v5.0 style datasources

- Considerations moving from v4.0 style datasource
- Behavior differ when using JSPs, servlets, and EJB session beans with bean managed transactions
  - ✓ Connection Sharing
    - Connections are now defaulted **Sharable** connections. Can cause Connections to be held longer and cause the ConnectionPool to empty
      - http://publib.boulder.ibm.com/infocenter/wasinfo/topic/com.ibm.websphere.base.doc/info/aes/ae/rdat_conmig.html
      - http://www-1.ibm.com/support/docview.wss?rs=0&q1=ConnectionWaitTimeoutException&uid=swg21192666&loc=en_US&cs=utf-8&cc=us&lang=en
  - ✓ Statement Cache Size
    - In v4.0, the maximum number of possible prepared statements are cached for the data source within an application server.
    - In v5.0, statement cache size is defined on a physical connection.
    - Can surface as application problems or affect performance

If you are migrating from WebSphere Application Server v3.5 or v4.0 and using JSPs, Servlets or EJB session beans with bean managed transactions you need to be aware of a few behavior changes. Symptoms of these behavior changes include session hangs, session timeouts and running out of connections. The reason for these behavior changes is that in v3.5 and v4.0 connection sharing was defaulted to not-shareable and in v5.x and above they are defaulted to shareable. This behavioral change is a J2EE v1.3 specification requirement.

Statement Cache Size has changed the location where it is defined, from application server in WebSphere Application Server v4.0 to being defined on a per physical connection in v5.x. This change can surface in performance problems.

The links provided in the slide give you further details and examples of potential outcomes if nothing is done on your part.


WebSphere Software
IBM

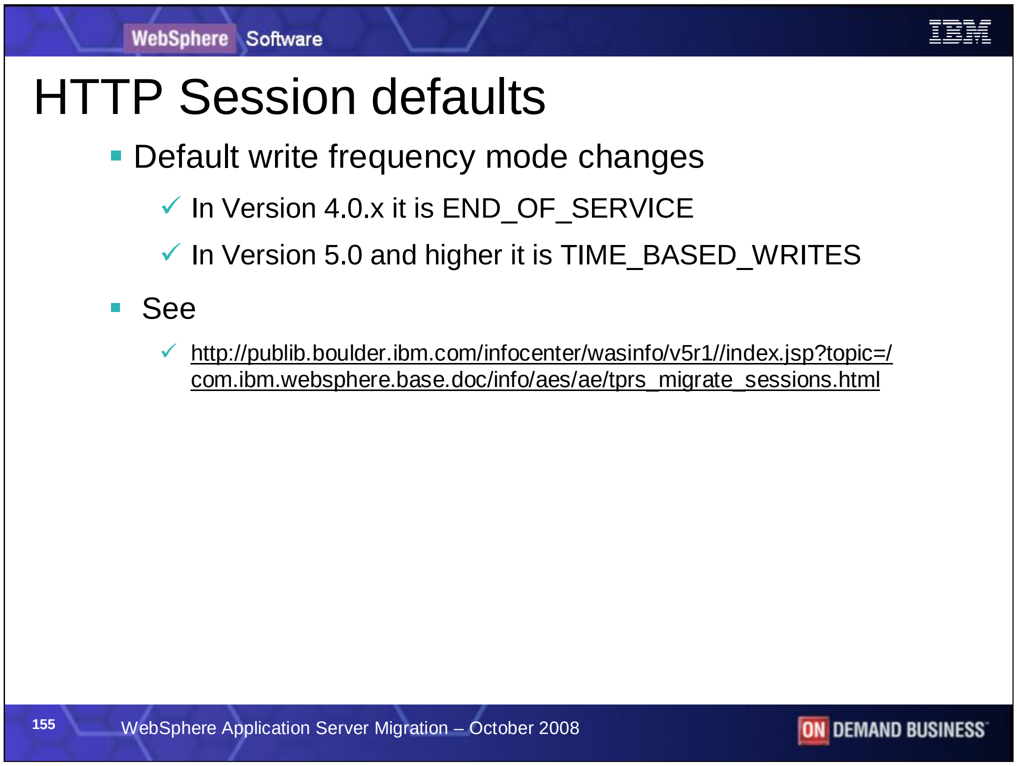
Transaction Isolation level

New October 2006

- An option if using the new v5 datasources at some point
  - ✓Not applicable on v4 datasources
- Added an option on the datasources that allow users to set the default isolation level in WebSphere using a DataSource custom property (**webSphereDefaultIsolationLevel**)
- Provided by PK05841
  - ✓Available in v6.0.2
- For more information see
  - ✓ http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21224492

A custom property “webspshereDefaultIsolationLevel” was introduced in v6.0.2 with APAR PK05841 that allows you to change the isolation level of a datasource. Changing the isolation level of a datasource is not a new concept and mechanisms are already in place to perform this step prior to the introduction of APAR PK05841. What this APAR does do is that it allows you to change the isolation level externally from your application, thus avoiding the need to make application changes on your part. If you do not set the isolation level, WebSphere Application Server will use the most restrictive setting to avoid deadlocks and this may have the potential to degrade your applications performance needlessly if you do not require such a restrictive isolation level.

The link provided in the slide will bring you the techdoc that describes the enhancement.



HTTP Session defaults

- Default write frequency mode changes
  - ✓ In Version 4.0.x it is END_OF_SERVICE
  - ✓ In Version 5.0 and higher it is TIME_BASED_WRITES
- See
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tprs_migrate_sessions.html

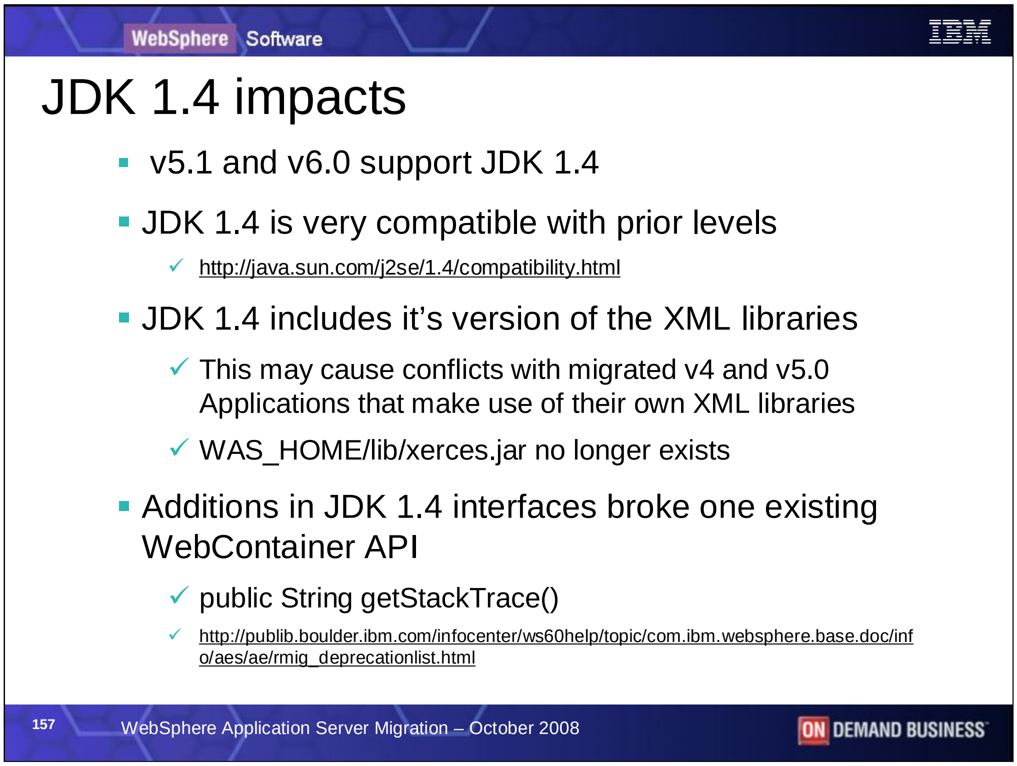
WebSphere Application Server v5.x has changed the default write frequency mode to TIME_BASED_WRITES from END_OF_SERVICE in versions prior to v5.0. These settings can be modified through the Session Manager. If you are coming from v4.x it will require no changes to your application. If you are coming from v3.5 then applications changes may need to be done.

Further information on possible v3.5 application changes can be found at the following InfoCenter document: http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tprs_migrate_sessions.html

WebSphere Software

Moving to JDK 1.4 or later

October 2008



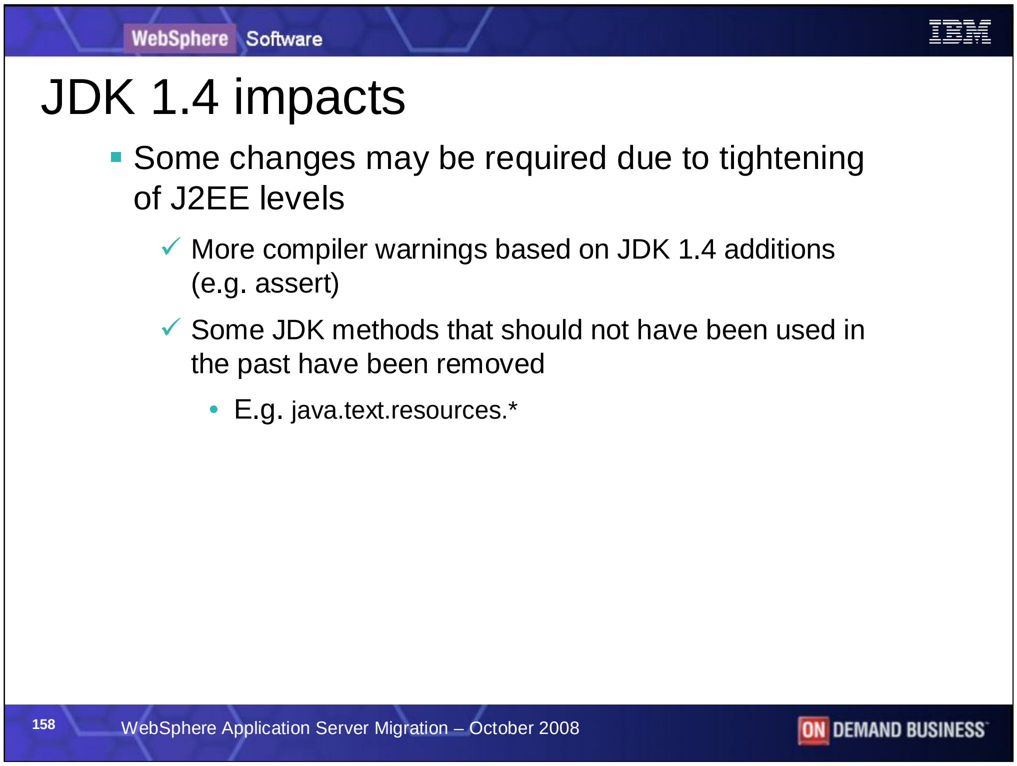
JDK 1.4 impacts

- v5.1 and v6.0 support JDK 1.4
- JDK 1.4 is very compatible with prior levels
  - ✓ http://java.sun.com/j2se/1.4/compatibility.html
- JDK 1.4 includes it's version of the XML libraries
  - ✓ This may cause conflicts with migrated v4 and v5.0 Applications that make use of their own XML libraries
  - ✓ WAS_HOME/lib/xerces.jar no longer exists
- Additions in JDK 1.4 interfaces broke one existing WebContainer API
  - ✓ public String getStackTrace()
  - ✓ http://publib.boulder.ibm.com/infocenter/ws60help/topic/com.ibm.websphere.base.doc/info/aes/ae/rmig_deprecationlist.html

Both v5.1 and v6.0 of Websphere Application Server support JDK v1.4. In general JDK v1.4 is compatible with prior levels of the JDK. There are however incompatibilities with prior versions. The number of incompatibilities are small and they more than likely will not effect your scenario, but they are definitely worth investigating during your Migration planning phase. The slide provides a link to the incompatibles identified in the JDK v1.4 specification.

JKD v1.4 started including a version of the XML libraries. Due to this inclusion the xerces.jar located under the WAS_HOME/lib directory has been removed. This change may cause some applications to operate incorrectly or not at all when migrating applications from v4 or v.5.0. If your application includes its own XML libraries and you have your class loading policy set to load the PARENT_FIRST after migration your application will be using the XML libraries included in JKD v1.4 rather than your own(classloading is a first load win algorithm) and this may have unpredictable results.

A change in one of the JDK v1.4 interfaces broke an existing WebContainer API. "com.ibm.websphere.servlet.error.ServletErrorReport" previously returned a "String", it now returns "StringTraceElement[]" due to a change in the java.lang.Throwable interface. WebSphere Application Server introduced "getStackTraceAsString()" for those customers that require the old behavior



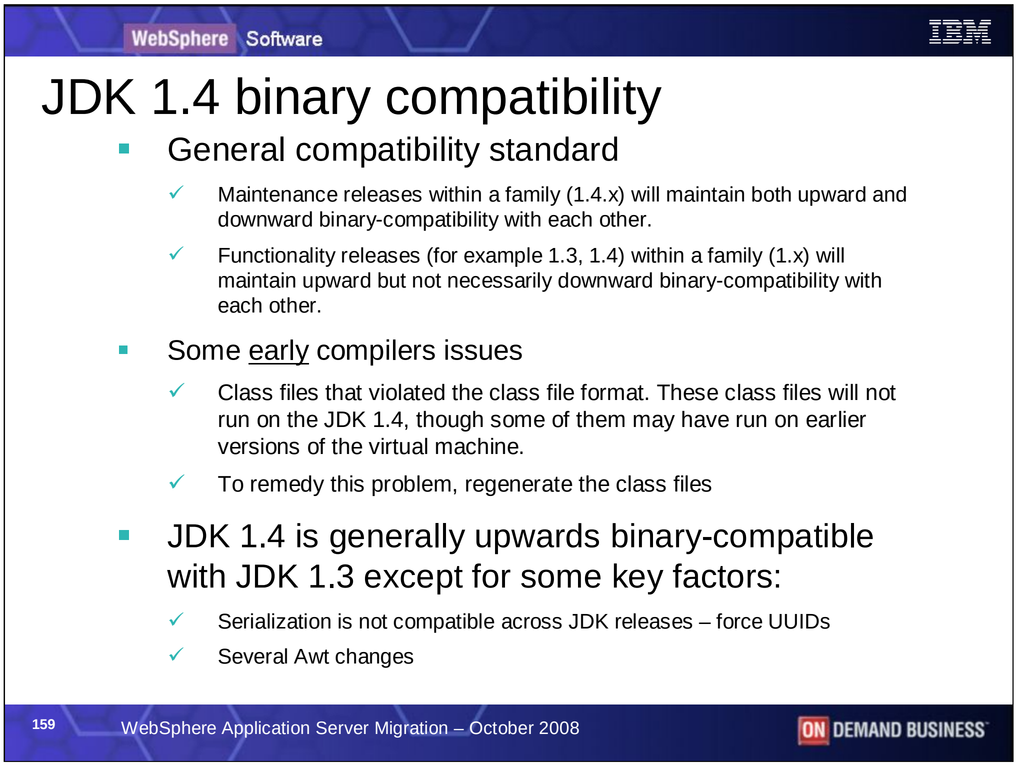
JDK 1.4 impacts

- Some changes may be required due to tightening of J2EE levels
  - ✓ More compiler warnings based on JDK 1.4 additions (e.g. assert)
  - ✓ Some JDK methods that should not have been used in the past have been removed
    - E.g. java.text.resources.*

Each version of the J2EE specification that is released contains many additions and refinements to the existing specification and the release of J2EE 1.4 is no different.

J2EE is forward compatible, so if your application worked in v1.2 or v1.3 it should work in v1.4 with a few exceptions. The authors of the J2EE specification have made every effort to be clear on all points, but due to the size and complexity of the specification things slip through and thus we are left with ambiguities. Each Application Server will interpret these ambiguities differently. J2EE v1.4 resolved many of these ambiguities and thus your applications should be checked for consistent behavior between your current version of WebSphere Application Server and your target version of WebSphere Application Server.

The best source of information for all of the known changes or refinements to the current version of the specification is in the “compatibility” section of the individual specifications.



JDK 1.4 binary compatibility

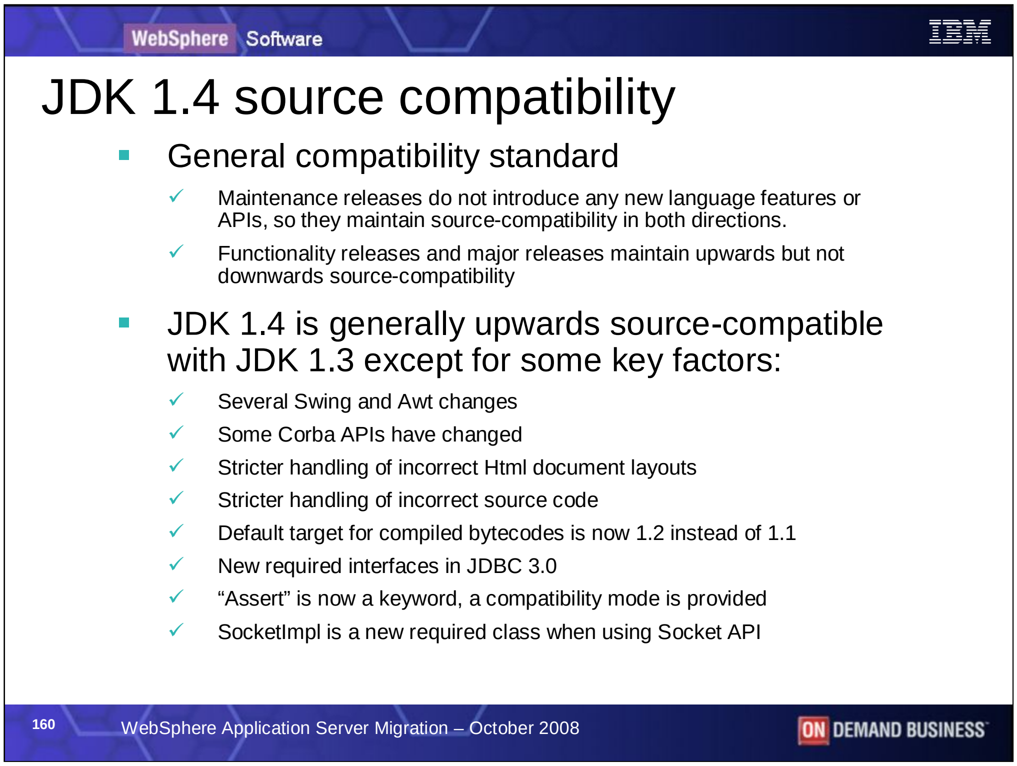
- General compatibility standard
  - ✓ Maintenance releases within a family (1.4.x) will maintain both upward and downward binary-compatibility with each other.
  - ✓ Functionality releases (for example 1.3, 1.4) within a family (1.x) will maintain upward but not necessarily downward binary-compatibility with each other.
- Some early compilers issues
  - ✓ Class files that violated the class file format. These class files will not run on the JDK 1.4, though some of them may have run on earlier versions of the virtual machine.
  - ✓ To remedy this problem, regenerate the class files
- JDK 1.4 is generally upwards binary-compatible with JDK 1.3 except for some key factors:
  - ✓ Serialization is not compatible across JDK releases – force UUIDs
  - ✓ Several Awt changes

JDK has a large customer base as you can imagine, so whenever and wherever possible the JDK tries to maintain compatibility. The JDK follows a few guidelines to address this issue. Maintenance releases within a family(example v1.4.1, v1.4.2…) will maintain both upward and downward binary-compatibility with the exception of the defined incompatibles in the specification. Functional releases(example 1.2, 1.3, 1.4) within a family will maintain upward but not necessarily downward binary-compatibility with each other.

The full list of incompatibilities and guidelines can be found at: http://java.sun.com/j2se/1.4/compatibility.html

Some earlier compilers violated the class format rules specified by the JDK specification. These class files will not run on JDK v1.4 and must be recompiled.

Differences in the way JDK v1.4 and v1.3 auto generate serial version UIDs will cause incompatibles in your application. To correct this issue you need to explicitly set a serial version UID for each serializable class. This and other breaking potentially breaking changes in JDK v1.4 can be found at: http://java.sun.com/j2se/1.4/compatibility.html



JDK 1.4 source compatibility

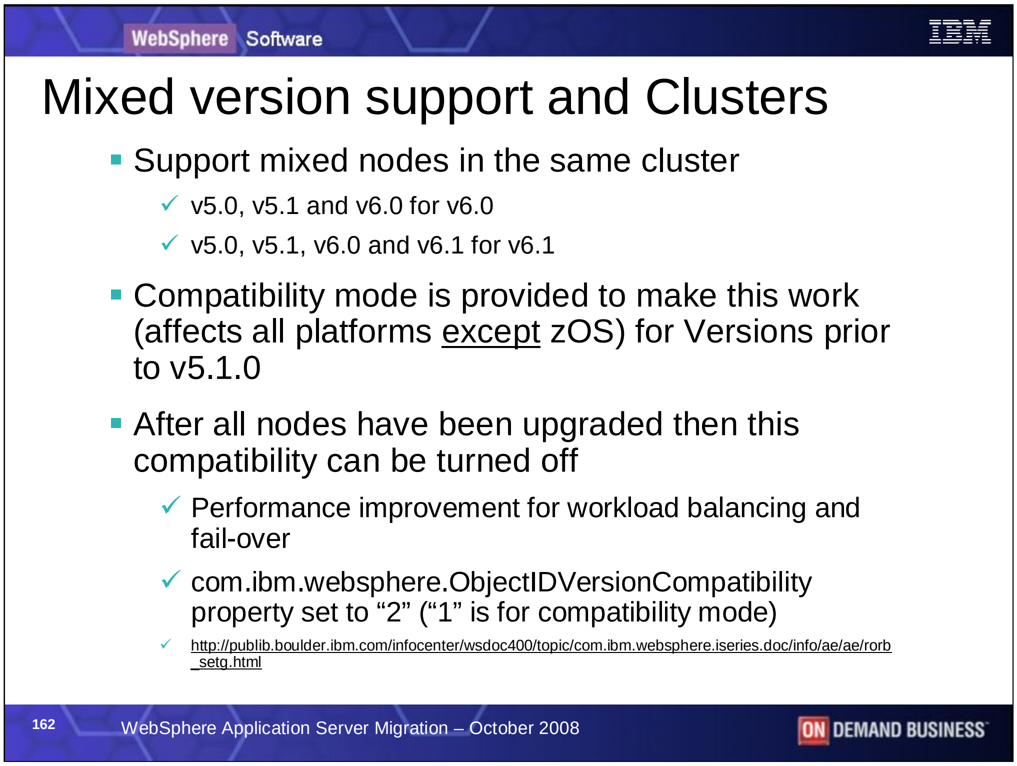
- General compatibility standard
  - ✓ Maintenance releases do not introduce any new language features or APIs, so they maintain source-compatibility in both directions.
  - ✓ Functionality releases and major releases maintain upwards but not downwards source-compatibility
- JDK 1.4 is generally upwards source-compatible with JDK 1.3 except for some key factors:
  - ✓ Several Swing and Awt changes
  - ✓ Some Corba APIs have changed
  - ✓ Stricter handling of incorrect Html document layouts
  - ✓ Stricter handling of incorrect source code
  - ✓ Default target for compiled bytecodes is now 1.2 instead of 1.1
  - ✓ New required interfaces in JDBC 3.0
  - ✓ “Assert” is now a keyword, a compatibility mode is provided
  - ✓ SocketImpl is a new required class when using Socket API

JDK has a large customer base as you can imagine, so whenever and wherever possible the JDK tries to maintain compatibility. The JDK follows a few guidelines to address this issue. Maintenance releases do not introduce any new language features or APIs, so they maintain source-compatibility in both directions. Functional releases and major releases maintain upwards but not downwards source-compatibility. There are a few situations where source-compatibility has not been maintained which are defined in the following link: http://java.sun.com/j2se/1.4/compatibility.html

WebSphere Software

Mixed version cell support

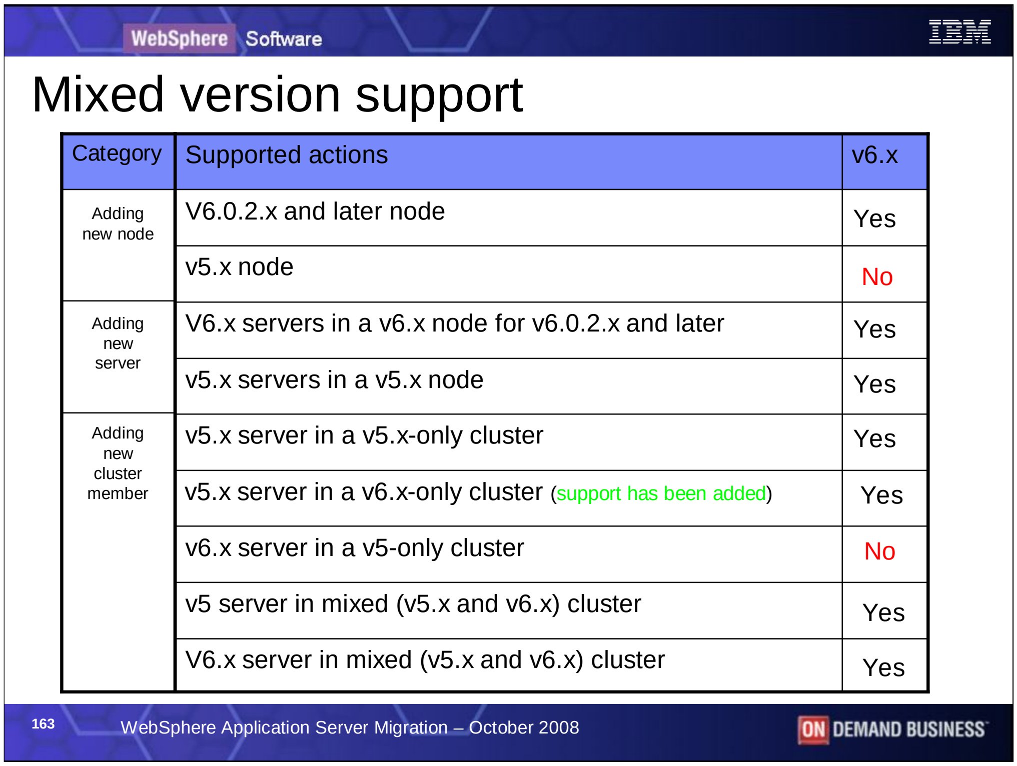
October 2008



Mixed version support and Clusters

- Support mixed nodes in the same cluster
  - ✓ v5.0, v5.1 and v6.0 for v6.0
  - ✓ v5.0, v5.1, v6.0 and v6.1 for v6.1
- Compatibility mode is provided to make this work (affects all platforms <u>except</u> zOS) for Versions prior to v5.1.0
- After all nodes have been upgraded then this compatibility can be turned off
  - ✓ Performance improvement for workload balancing and fail-over
  - ✓ com.ibm.websphere.ObjectIDVersionCompatibility property set to “2” (“1” is for compatibility mode)
  - ✓ http://publib.boulder.ibm.com/infocenter/wsdoc400/topic/com.ibm.websphere.iseries.doc/info/ae/ae/rorb_setg.html

WebSphere Application Server v6 supports a mix environment(meaning that your cell can contain v5.0, v5.1 as well as v6 nodes). This support extends to clusters as well, so for example ClusterX could contain a v5.0, v5.1 and a v6.0 node and workload management would route requests to all nodes seamlessly. Workload Management introduced new features in v6. This results in a limitation in that mix clusters cannot use these new features until all cluster members have been migrated to the v6 level. These new features are disabled via a flag property(see slide) and this property must be reset once all nodes in the cluster have been migrated to take advantage of the new Workload Management features.



Mixed version support

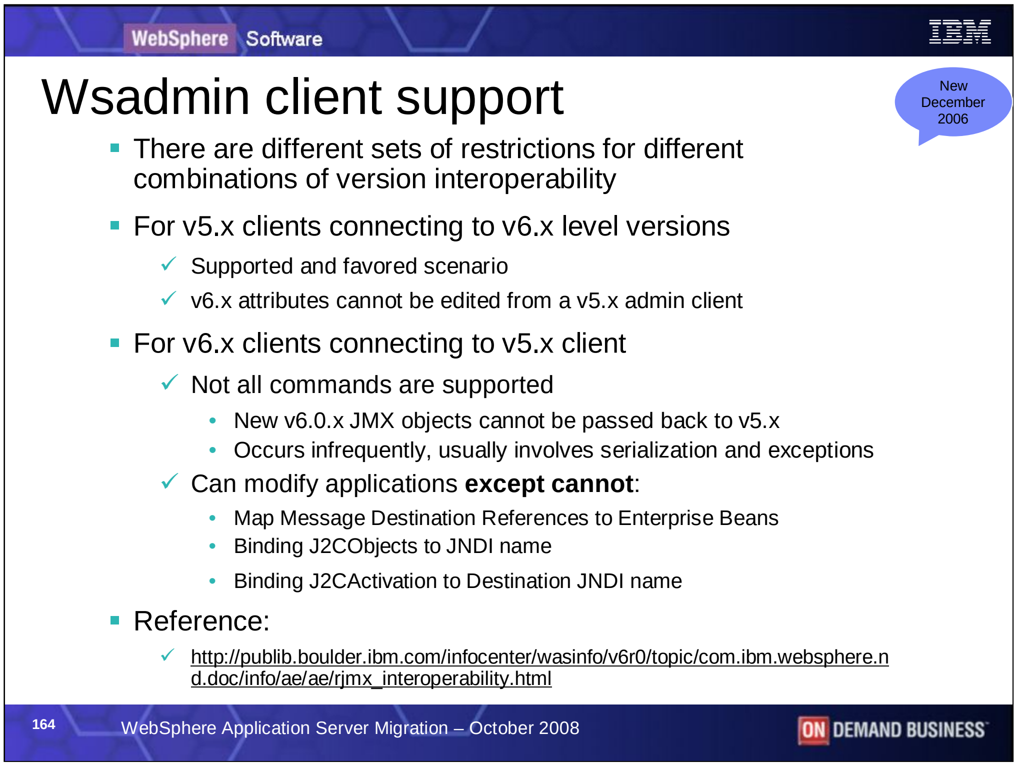
Category	Supported actions	v6.x
Adding new node	V6.0.2.x and later node	Yes
	v5.x node	No
Adding new server	V6.x servers in a v6.x node for v6.0.2.x and later	Yes
	v5.x servers in a v5.x node	Yes
Adding new cluster member	v5.x server in a v5.x-only cluster	Yes
	v5.x server in a v6.x-only cluster (support has been added)	Yes
	v6.x server in a v5-only cluster	No
	v5 server in mixed (v5.x and v6.x) cluster	Yes
	V6.x server in mixed (v5.x and v6.x) cluster	Yes

Permanent restrictions

- v6.0 attributes cannot be edited from a v5.x admin client
- A v5.x node cannot be directly added to a v6.0 cell.
- A v5.x node cannot be directly removed from a v6.0 cell.
  - However, it can be done indirectly
- Some application install options cannot be used for applications to be installable on both v5.x and v6.0
  - deployejb
  - deployws
  - precompileJSP
  - useMetaDataFromBinary (Note, this is fixed in v6.1.0.1 and later via PK27243)
- zOS v5.0.x nodes are not supported in the same cell and in the same LPAR

Available in v6.0.2+

- Clusters with v5.x nodes can have v5.x members added
- New servers may be defined to v5.x nodes



WebSphere Software

Wsadmin client support

New
December
2006

- There are different sets of restrictions for different combinations of version interoperability
- For v5.x clients connecting to v6.x level versions
  - ✓ Supported and favored scenario
  - ✓ v6.x attributes cannot be edited from a v5.x admin client
- For v6.x clients connecting to v5.x client
  - ✓ Not all commands are supported
    - New v6.0.x JMX objects cannot be passed back to v5.x
    - Occurs infrequently, usually involves serialization and exceptions
  - ✓ Can modify applications **except cannot**:
    - Map Message Destination References to Enterprise Beans
    - Binding J2CObjects to JNDI name
    - Binding J2CActivation to Destination JNDI name
- Reference:
  - ✓ http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rjmx_interoperability.html

164 WebSphere Application Server Migration – October 2008

You can use 6.0.x administration to modify 5.x and 6.0.x applications

All applications may be updated by reinstallation, editing, or remapping modules to new targets. When remapping a module, the new target for a Version 6.0.x module may not be a Version 5.x target. When editing an application from a Version 6.0.x client, all editing functions are available for all versions of applications.

When editing a Version 6.0.x application from a Version 5.x client, the functions that are provided exclusively by the Version 6.0.x runtime are not available. They include:

Map Message Destination References to Enterprise Beans

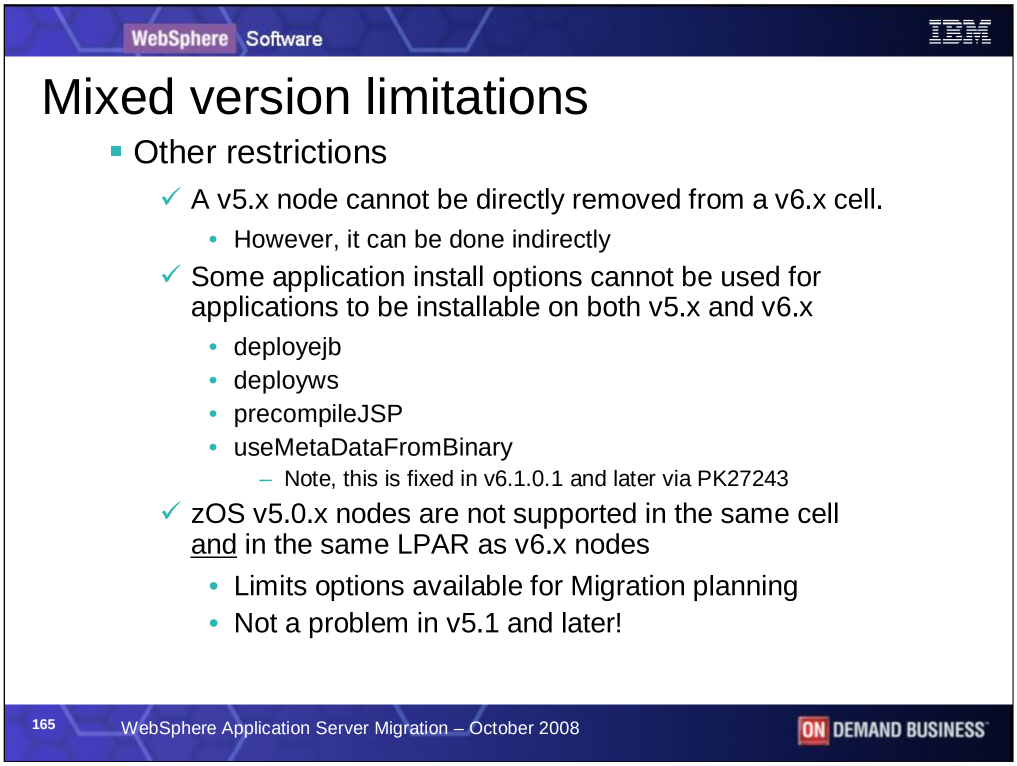
Binding J2CObjects to JNDI name

Binding J2CActivation to Destination JNDI name

Version 6.0.x implements JMX 1.2, while Version 5.x implements JMX 1.1. Due to the evolution of the JMX specification, the serialization format for JMX objects, such as javax.management.ObjectName, differs between the two specifications.

The Version 6.0.x JMX run time has been enhanced to be aware of the version of the client with which it is communicating. It makes appropriate transformations on these incompatible serialized formats so as to allow the different version run times to communicate with each other. This makes it possible for a Version 5.x administrative client can call a Version 6.0.x deployment manager, node, or server. Similarly, a Version 6.0.x administrative client can call a Version 5.x node or server.

Instances of JMX classes new in Version 6.0.x cannot be passed back into Version 5.x. Problems usually are signaled by a particular exception. Also, note that you might see different exceptions from your Version 6.0.x and Version 5.x administrative programs.



Mixed version limitations

- Other restrictions
  - ✓ A v5.x node cannot be directly removed from a v6.x cell.
    - However, it can be done indirectly
  - ✓ Some application install options cannot be used for applications to be installable on both v5.x and v6.x
    - deployejb
    - deployws
    - precompileJSP
    - useMetaDataFromBinary
      - Note, this is fixed in v6.1.0.1 and later via PK27243
  - ✓ zOS v5.0.x nodes are not supported in the same cell and in the same LPAR as v6.x nodes
    - Limits options available for Migration planning
    - Not a problem in v5.1 and later!

A restriction has been imposed which does not allow for the direct removal of a v5.x node from a v6 cell. There are workarounds to this issue. Workaround#1 is that you upgrade the v5.x node to v6.x using the Migration tooling and then perform and then run the removeNode.sh command. Workaround#2 is that you uninstall v5.x and then run cleanupNode.sh on the Deployment Manager node to remove any relics.

Application install options deployejb, deployws and precompileJSP are not supported when targeting a v5.x environment, the reason behind this is the v6 deployment tools are not aware of the target environment and thus cannot do a correct job deploying the application.

z/OS v5.0 nodes cannot coexist in the same cell AND in the same LPAR as v6 nodes. This limitation has been lifted in v5.1 and later. This limits your options when planning for migration.

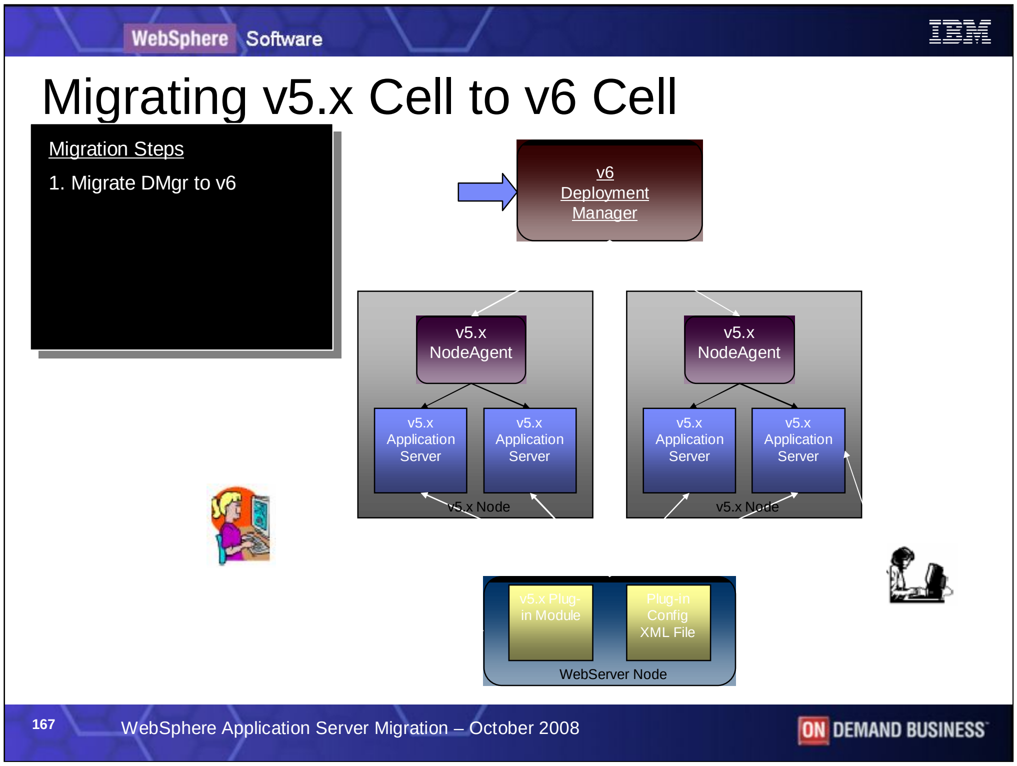
Further information on these topic can be found here: http://www-128.ibm.com/developerworks/websphere/techjournal/0502_cheng/0502_cheng.html

WebSphere Software

Using the runtime Migration tools

October 2008

ON DEMAND BUSINESS



Migrating v5.x Cell to v6 Cell

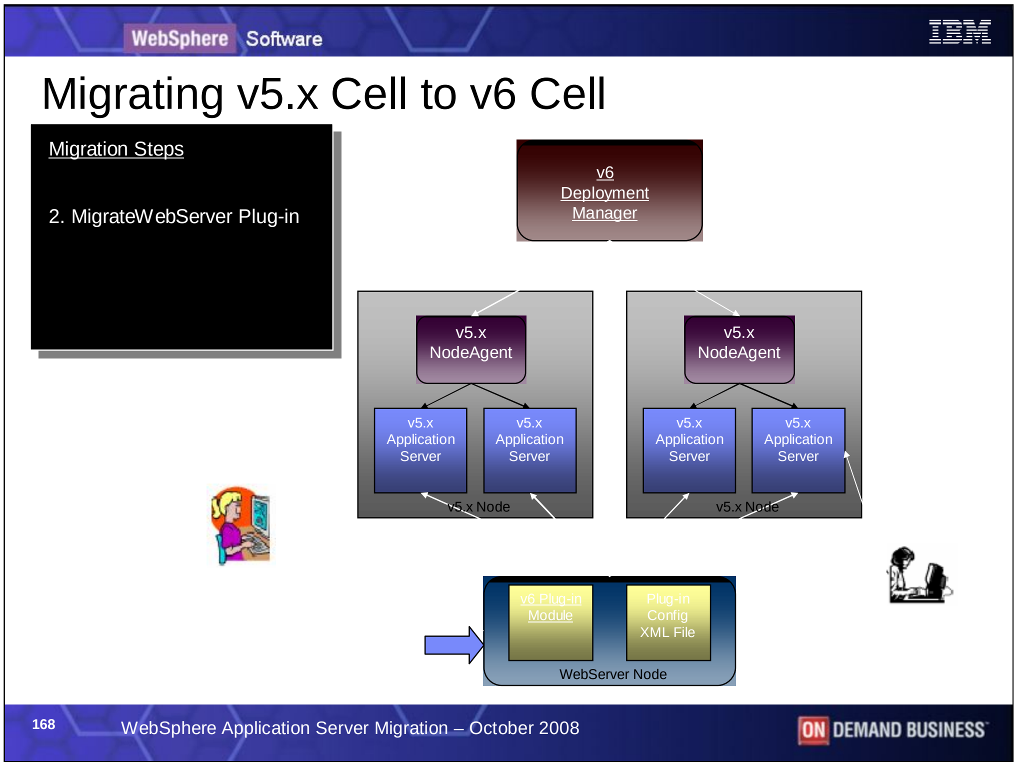
Migration Steps

1. Migrate DMgr to v6

When migrating or updating an ND configuration you need to keep in mind that the Deployment Manager node must always be at the highest service level.

The following slides show a high level view of the migration steps and in which order you need to perform them to ensure a successful migration.

Step#1: Deployment Manager must be the first node to be migrated. A v6 Deployment Manager can communicate with v5 Federated nodes, but not vice versa. During this step the Migration tooling will migrate the cell scoped configuration data and the Deployment Manager node and server scoped data. Since Applications are considered to be part of the cell these will get migrated forward, this is done redeploying them in your v6 WebSphere Application Server environment. Once this step has been completed you will be left with a v6 Deployment Manager managing your v5 Federated nodes.


WebSphere Software

IBM

Migrating v5.x Cell to v6 Cell

Migration Steps

2. MigrateWebServer Plug-in

v6
Deployment
Manager

v5.x
NodeAgent

v5.x
Application
Server

v5.x
Application
Server

v5.x Node

v5.x
NodeAgent

v5.x
Application
Server

v5.x
Application
Server

v5.x Node

v6 Plug-in
Module

Plug-in
Config
XML File

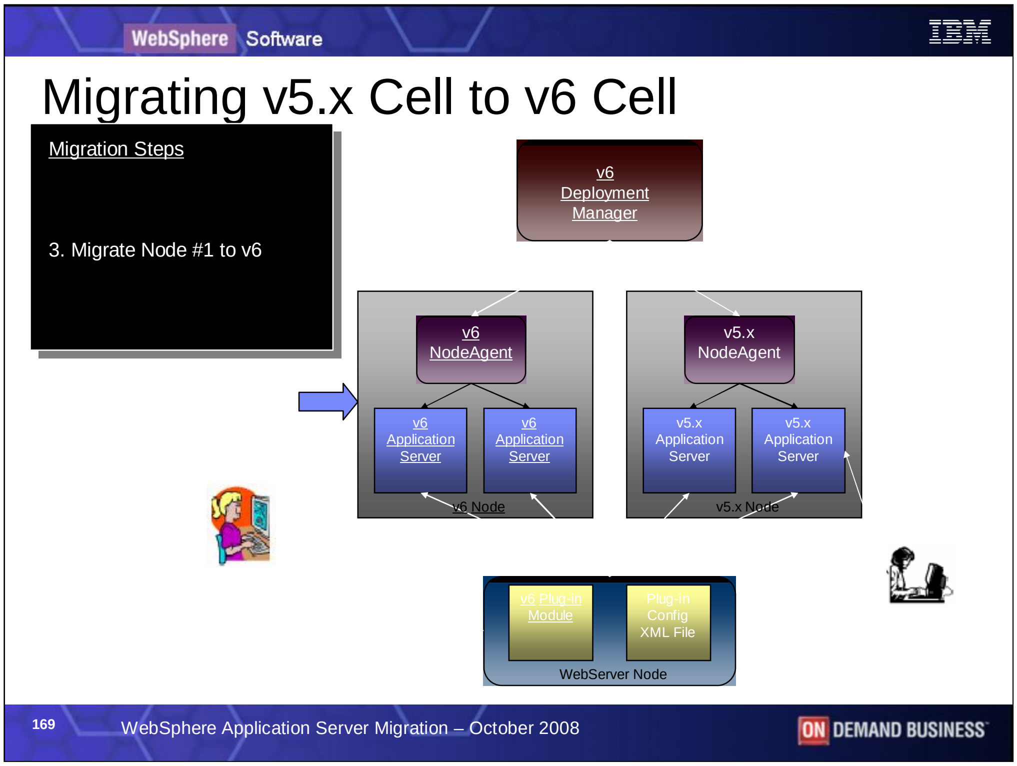
WebServer Node

168 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

Step#2: Install a WebSphere Application Server v6 compliant WebServer and migrate your WebServer settings. This step needs to be done before any v5 Federated nodes are migrated, this is because a v6 compliant WebServer can communicate or send requests to your v5 Federated nodes, but not vice versa.

Further details can be found in InfoCenter document:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tins_migWebserver.html



WebSphere Software

Migrating v5.x Cell to v6 Cell

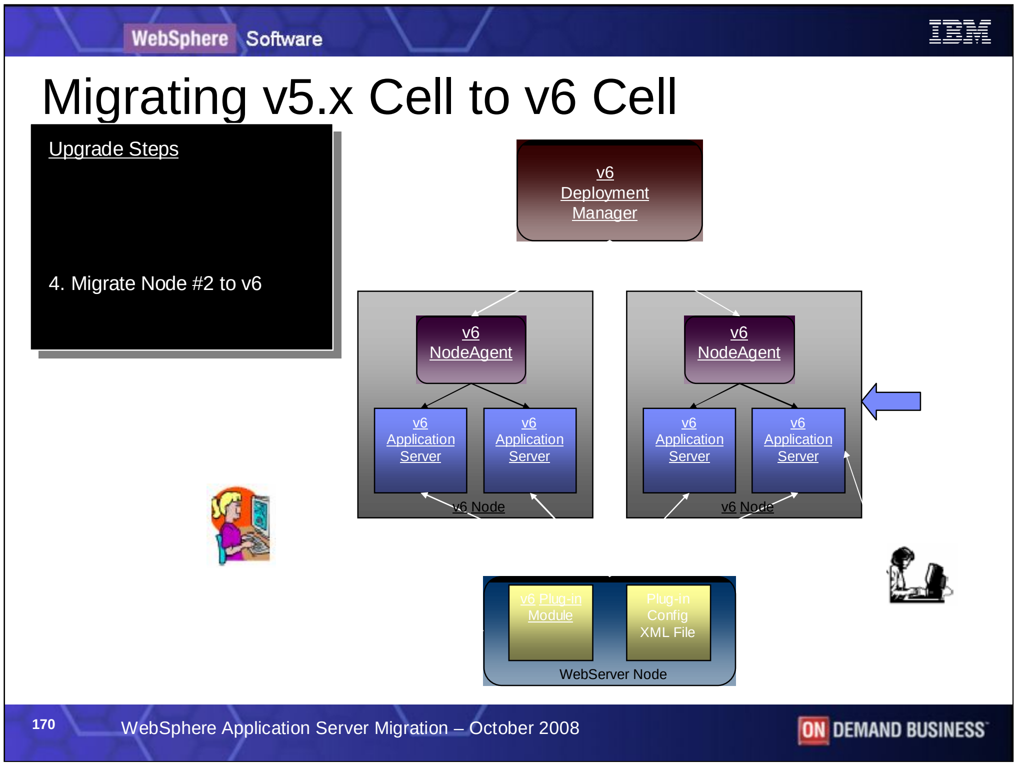
Migration Steps

3. Migrate Node #1 to v6

169 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

Step#3: At this point you have a v6 Deployment Manager managing your v5 Federated nodes and a v6 compliant WebServer routing application requests to your v5 Federated nodes. This step involves installing WebSphere Application Server v6 in a different directory(WebSphere Application Server does not support in-place install) and use the Migration tooling to migrate your v5 configuration data into your newly installed v6. Once you have completed this step you will be left with a v6 Deployment Manager managing a v5 Federated node, v6 Federated node and a v6 compliant WebServer routing application requests to the v5 and v6 Federated nodes. This is what is called a mix-node environment and WebSphere Application Server's support of this environment is crucial to the rolling migration(which is what you are performing). There are things to be aware and plan for when you begin your migration and thus entering a mix-node environment; For example you are not allowed to add additional v5 nodes to a mix-cluster(cluster that contains v5 and v6 nodes), see planning section for further details.



WebSphere Software IBM

Migrating v5.x Cell to v6 Cell

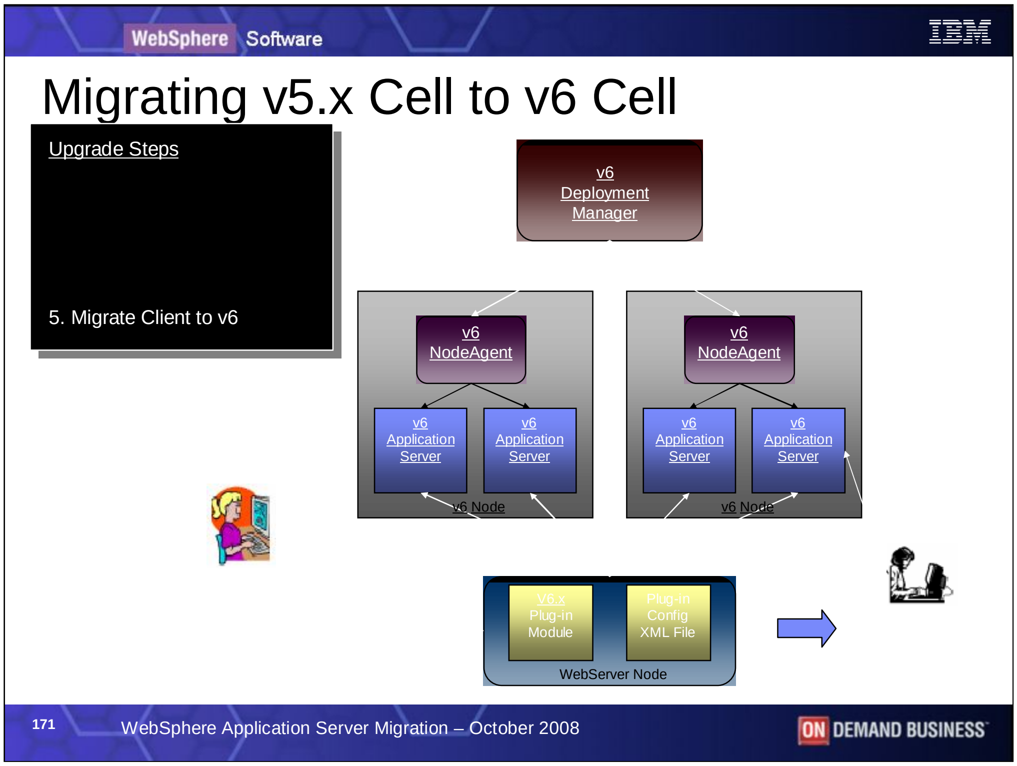
Upgrade Steps

4. Migrate Node #2 to v6

170 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

Step#4: See Step#3 in previous slide for description. At this point you are not in the mix-node environment anymore. Once you are this stage, all v6 features are available to you without restrictions.



WebSphere Software

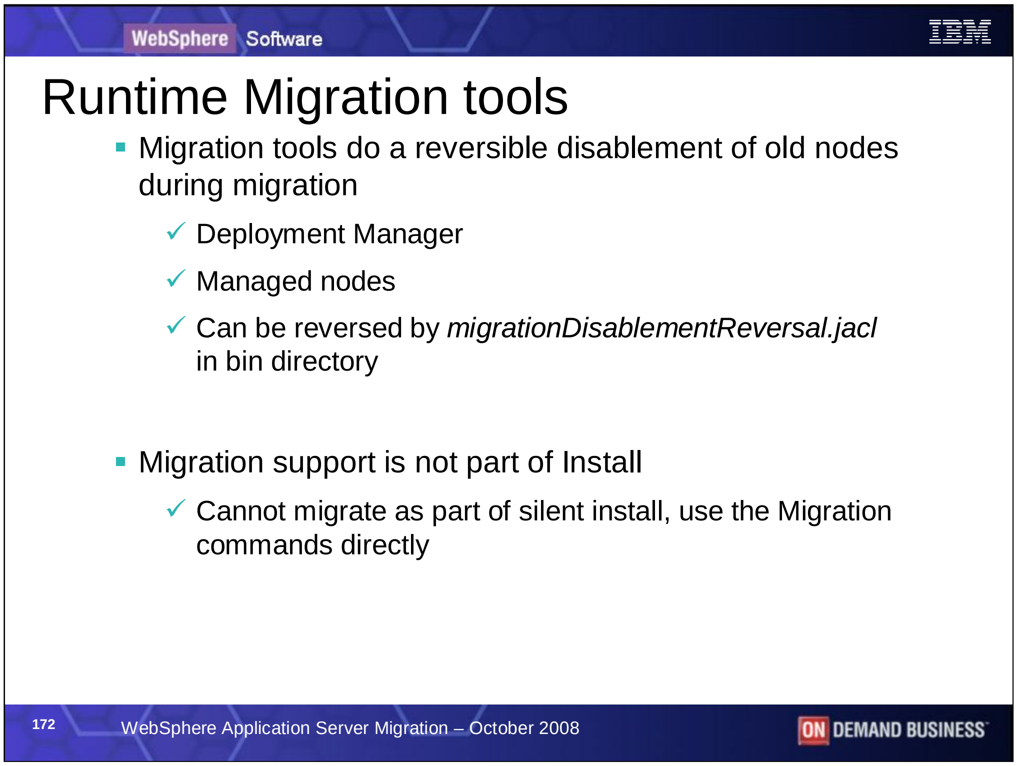
Migrating v5.x Cell to v6 Cell

Upgrade Steps

5. Migrate Client to v6

171 WebSphere Application Server Migration – October 2008

Step#5: Migrate any WebSphere Application Server client installations to v6.



WebSphere Software

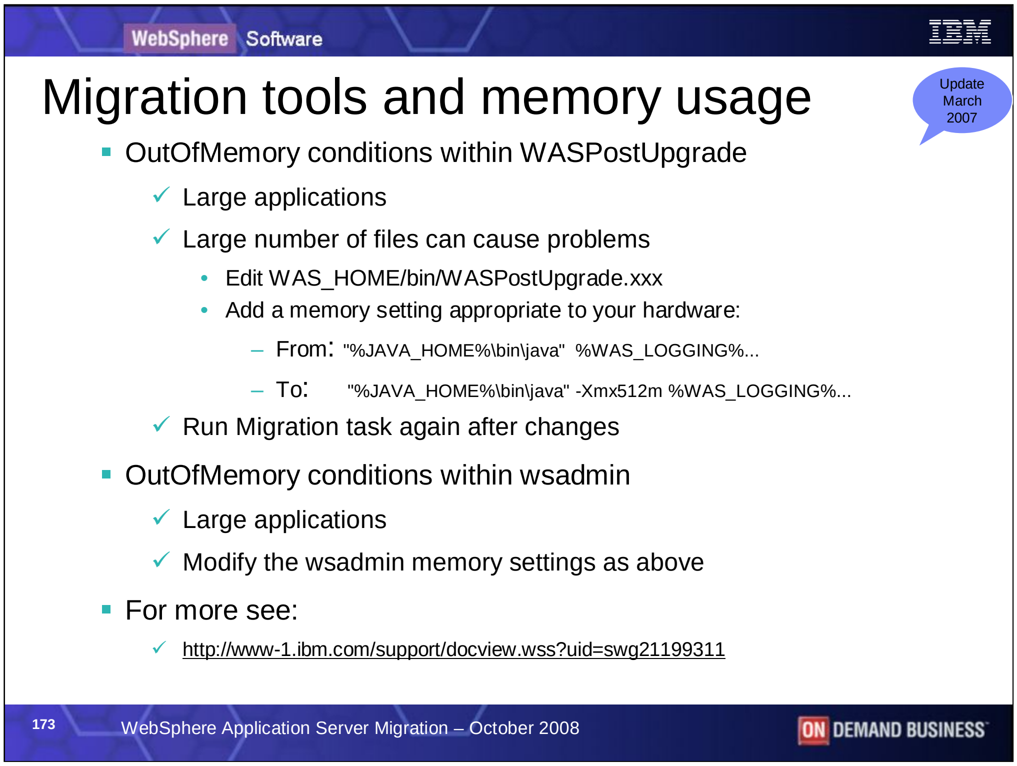
Runtime Migration tools

- Migration tools do a reversible disablement of old nodes during migration
  - ✓ Deployment Manager
  - ✓ Managed nodes
  - ✓ Can be reversed by *migrationDisablementReversal.jacl* in bin directory
- Migration support is not part of Install
  - ✓ Cannot migrate as part of silent install, use the Migration commands directly

During the migration of the Deployment Manager and managed nodes the Migration tooling will disable the prior version of the Deployment Manager or managed node for which you are migrating. The rationale for this disablement action in the case of the Deployment Manager is that it is not supported to have two Deployment Managers managing your cell, corruption is sure to result from this setup. The rationale for this disablement action in the case of the managed node is that it is not supported to have two physical installations representing the same node, again corruption is sure to result from this setup. If for some reason you wish to undo this disablement action you can run the “migrationDisablementReversal.jacl” script located in your prior installations “bin” directory.

An example usage of this script can be found in InfoCenter document: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tmig_rollbackdm.html

Support for the Migration tooling as part of Install is no longer available. You will need to run the commands directly.



Migration tools and memory usage

Update March 2007

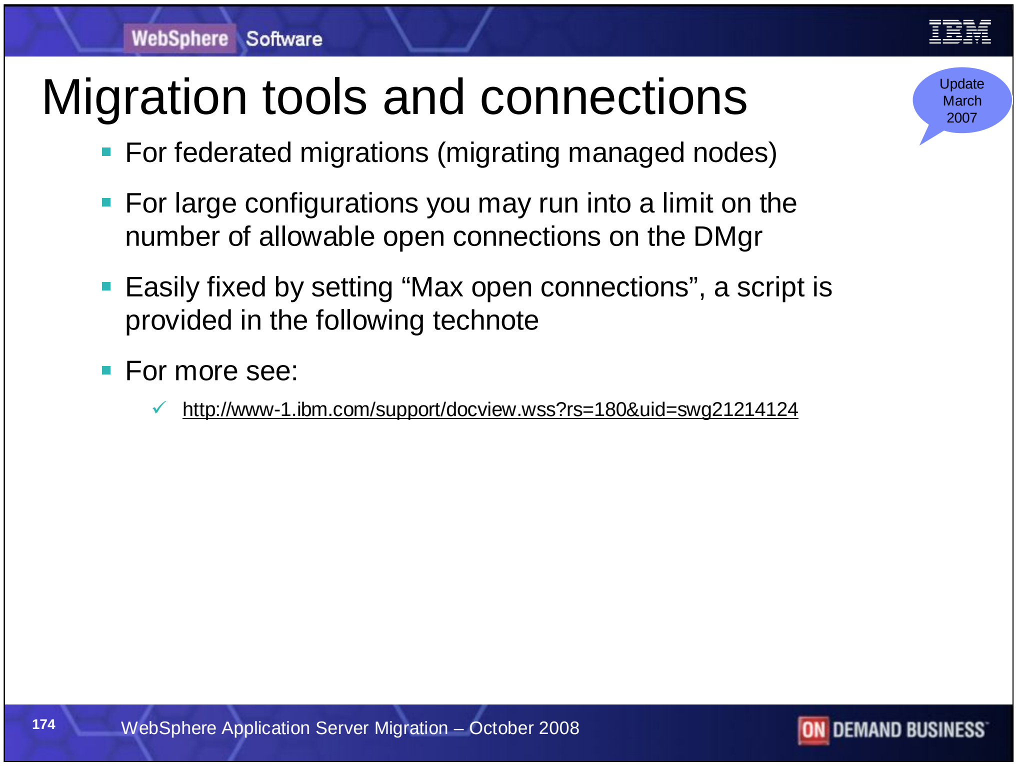
- OutOfMemory conditions within WASPostUpgrade
  - ✓ Large applications
  - ✓ Large number of files can cause problems
    - Edit WAS_HOME/bin/WASPostUpgrade.xxx
    - Add a memory setting appropriate to your hardware:
      - From: "%JAVA_HOME%\bin\java" %WAS_LOGGING%...
      - To: "%JAVA_HOME%\bin\java" -Xmx512m %WAS_LOGGING%...
  - ✓ Run Migration task again after changes
- OutOfMemory conditions within wsadmin
  - ✓ Large applications
  - ✓ Modify the wsadmin memory settings as above
- For more see:
  - ✓ http://www-1.ibm.com/support/docview.wss?uid=swg21199311

WASPostUpgrade.sh has the potential to throw an OutOfMemory error when migrating very large applications. This is due to fact that WASPostUpgrade.sh uses the default heap size which for the most part is an adequate setting for most customers, but with any setting it does not fit all customers. If you happen to be one of the unlucky ones that fall victim to this problem you have two choices to remedy the problem:

1) You can up the heap size in the WASPostUpgrade.sh script and rerun migration from scratch.

2) You can run the application installation portion of the migration manually via scripts that were saved in the backup directory during the migration operation.

A more detailed description and instructions can be found at the follow: http://www-1.ibm.com/support/docview.wss?uid=swg21199311



WebSphere Software

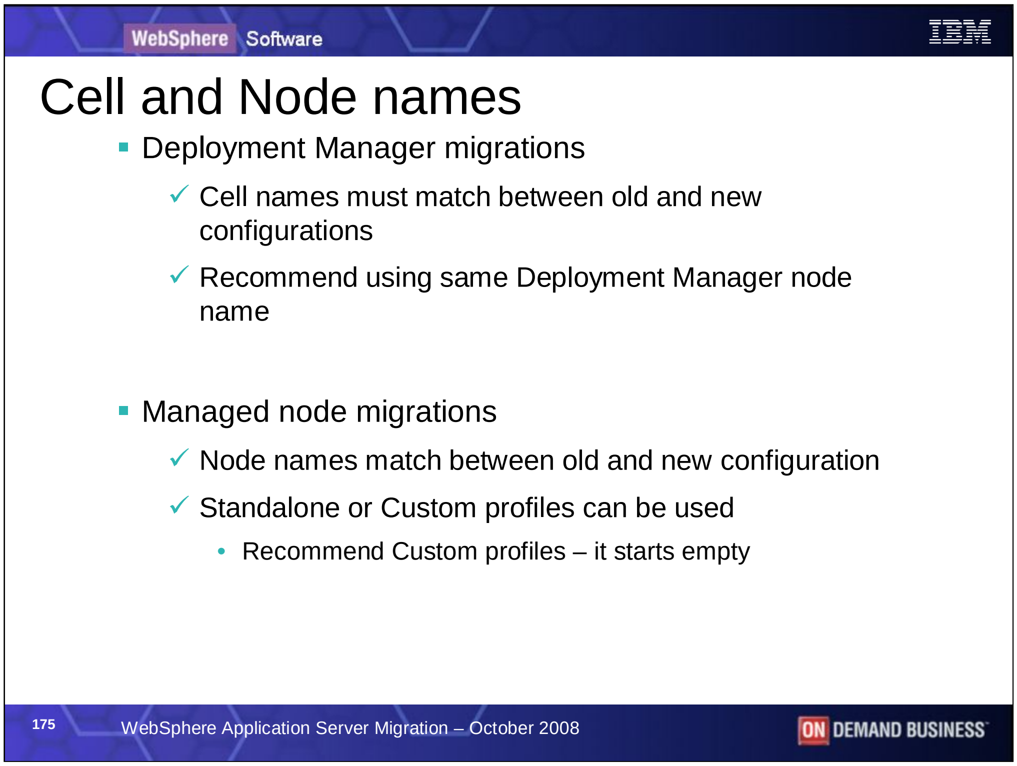
Migration tools and connections

Update March 2007

- For federated migrations (migrating managed nodes)
- For large configurations you may run into a limit on the number of allowable open connections on the DMgr
- Easily fixed by setting “Max open connections”, a script is provided in the following technote
- For more see:
  - ✓ http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21214124

174 WebSphere Application Server Migration – October 2008

For federated nodes with a large number of objects, like application servers this has been a problem. It is easily fixed with the setting as noted using the **modify_WC_adminhost.jacl** script that is included there. The settings can be rest after migration of all nodes is completed.

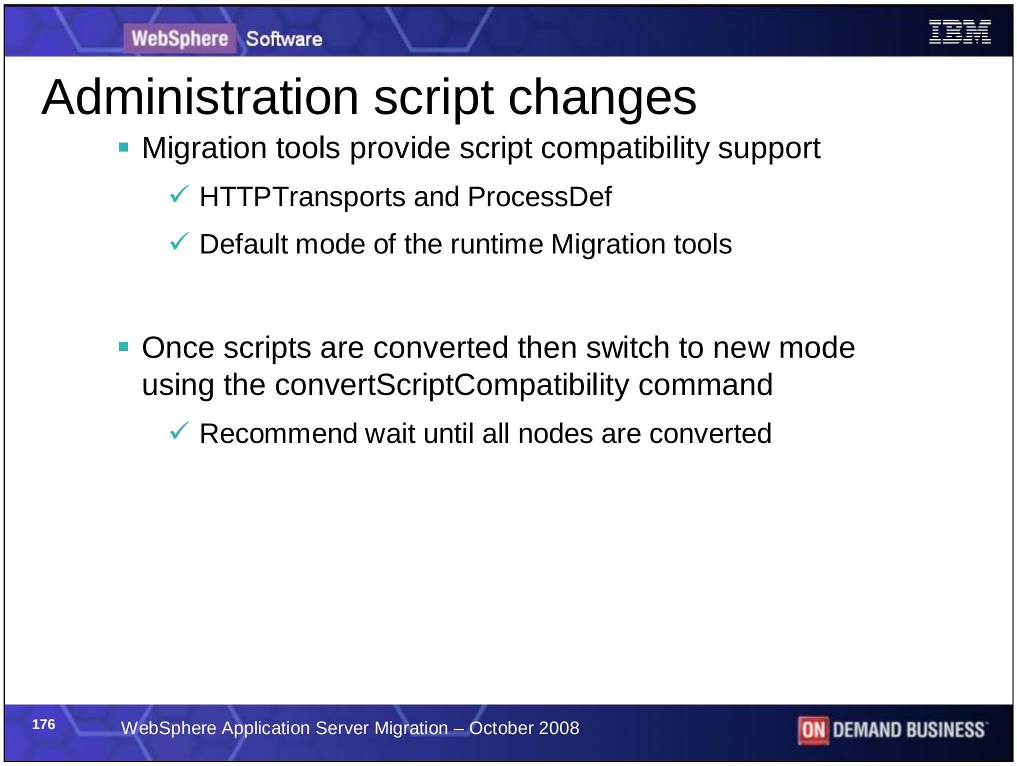


Cell and Node names

- Deployment Manager migrations
  - ✓ Cell names must match between old and new configurations
  - ✓ Recommend using same Deployment Manager node name
- Managed node migrations
  - ✓ Node names match between old and new configuration
  - ✓ Standalone or Custom profiles can be used
    - Recommend Custom profiles – it starts empty

Deployment Manager migration requires that your cell names between WebSphere Application Server v5 and v6 are the same prior to migration. It is also recommended that you keep your Deployment Manager node names the same as well. There are many internal references to these cell and node names as well any external references you may have, so by keeping with a consistent naming scheme you remove at least one avenue of potential errors during migration.

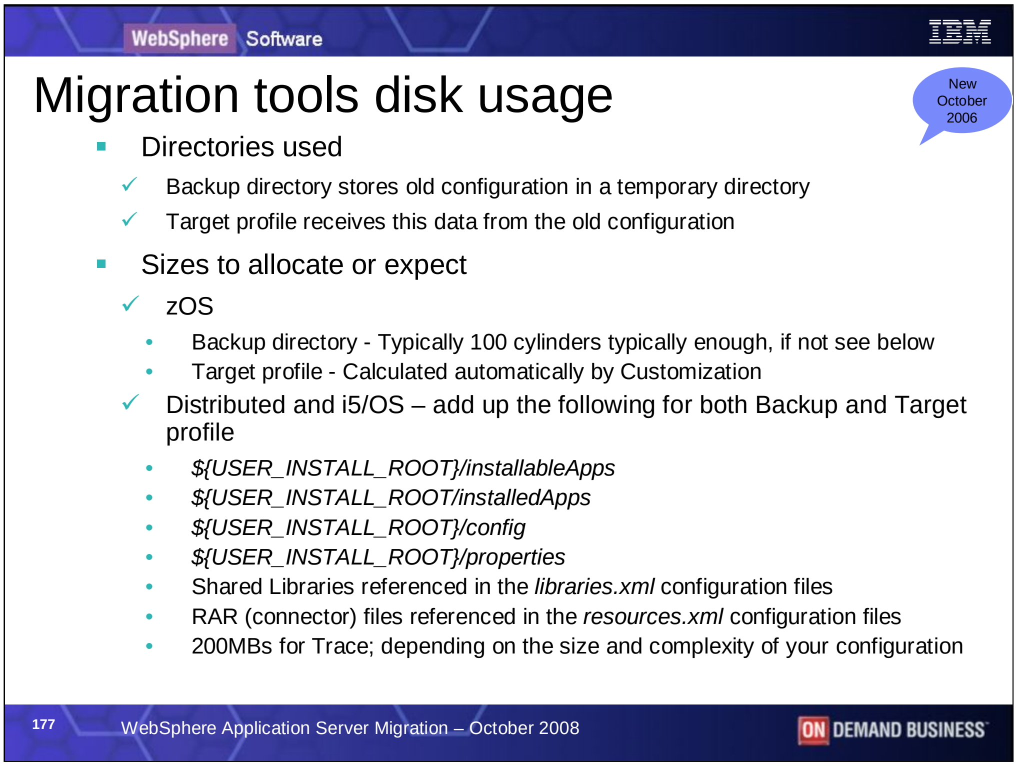
Managed node migration requires that your node names between WebSphere Application Server v5 and v6 are the same prior to migration. It is also recommended that you start with a “custom” profile prior to migration. This allows for a clean slate for which the Migration tooling will migrate into and thus removing a potential avenue of errors.



Administration script changes

- Migration tools provide script compatibility support
  - ✓ HTTPTransports and ProcessDef
  - ✓ Default mode of the runtime Migration tools
- Once scripts are converted then switch to new mode using the convertScriptCompatibility command
  - ✓ Recommend wait until all nodes are converted

Wherever possible WebSphere is committed to maintaining your existing scripts. This support extends into the Migration tooling(WASPostUpgrade.sh). The Migration tooling offers support to migrate while maintaining compatibility with existing scripts. This compatibility is accomplished by bringing over the internal configuration data as is and letting the runtime convert on the fly. The Migration tooling by default will maintain compatibility, but if you do not care or do not have a need for compatibility it is recommended that you run WASPostUpgrade.sh with the –scriptCompatibility flag set to “false”.



WebSphere Software

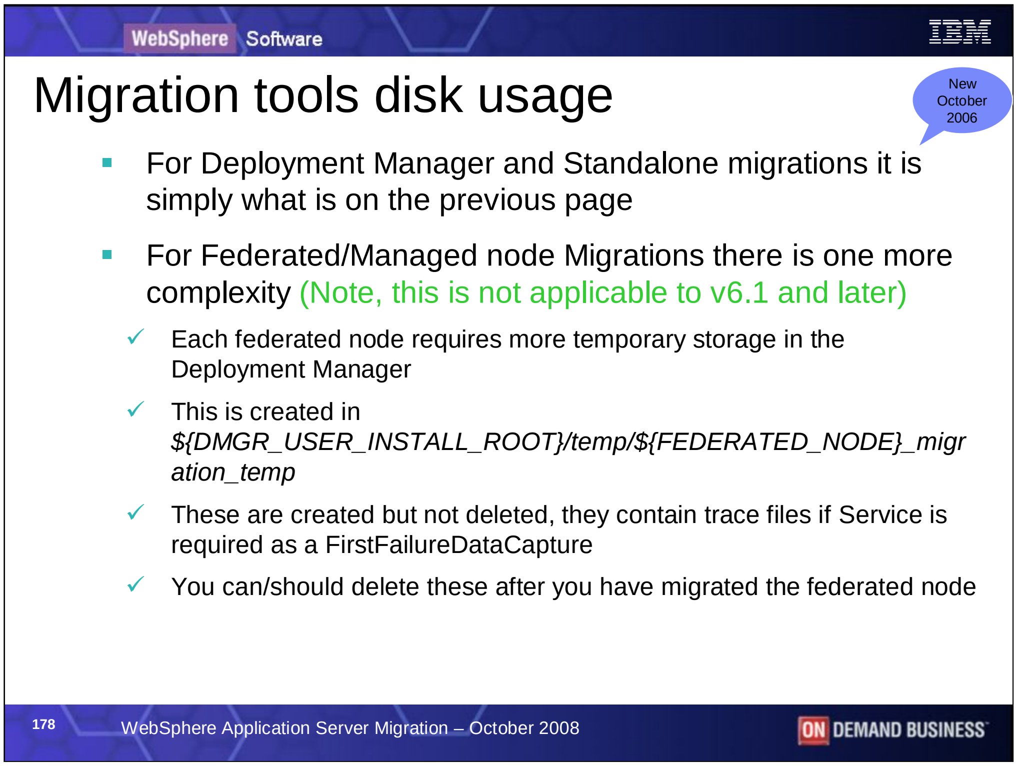
Migration tools disk usage

New October 2006

- Directories used
  - ✓ Backup directory stores old configuration in a temporary directory
  - ✓ Target profile receives this data from the old configuration
- Sizes to allocate or expect
  - ✓ zOS
    - Backup directory - Typically 100 cylinders typically enough, if not see below
    - Target profile - Calculated automatically by Customization
  - ✓ Distributed and i5/OS – add up the following for both Backup and Target profile
    - *${USER_INSTALL_ROOT}/installableApps*
    - *${USER_INSTALL_ROOT/installedApps*
    - *${USER_INSTALL_ROOT}/config*
    - *${USER_INSTALL_ROOT}/properties*
    - Shared Libraries referenced in the *libraries.xml* configuration files
    - RAR (connector) files referenced in the *resources.xml* configuration files
    - 200MBs for Trace; depending on the size and complexity of your configuration

Migration tooling is broken up into two steps. Step#1(WASPreUpgrade.sh) is run against your prior WebSphere Application Server installation and the relevant(relevant is based on the contents of the configuration data) content is backed up into a user specified backup directory. Depending on the size and complexity of your configuration the backup directory can become quite large. It is best to prepare for this situation by estimating your storage needs. It may be next to impossible to diagnose a storage limitation problem(when a storage limit is reached all tracing is halted and thus the error is not included in the trace files for diagnosing the problem), so proper planning is a must. Unfortunately there is no one size fits all answer that we can give you, it really is dependent on your configuration, number of application installed and the number of external references(ex: .rar files, shared libraries).

The following InfoCenter document provides instructions for estimating your requirements:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/cmig_pre.html



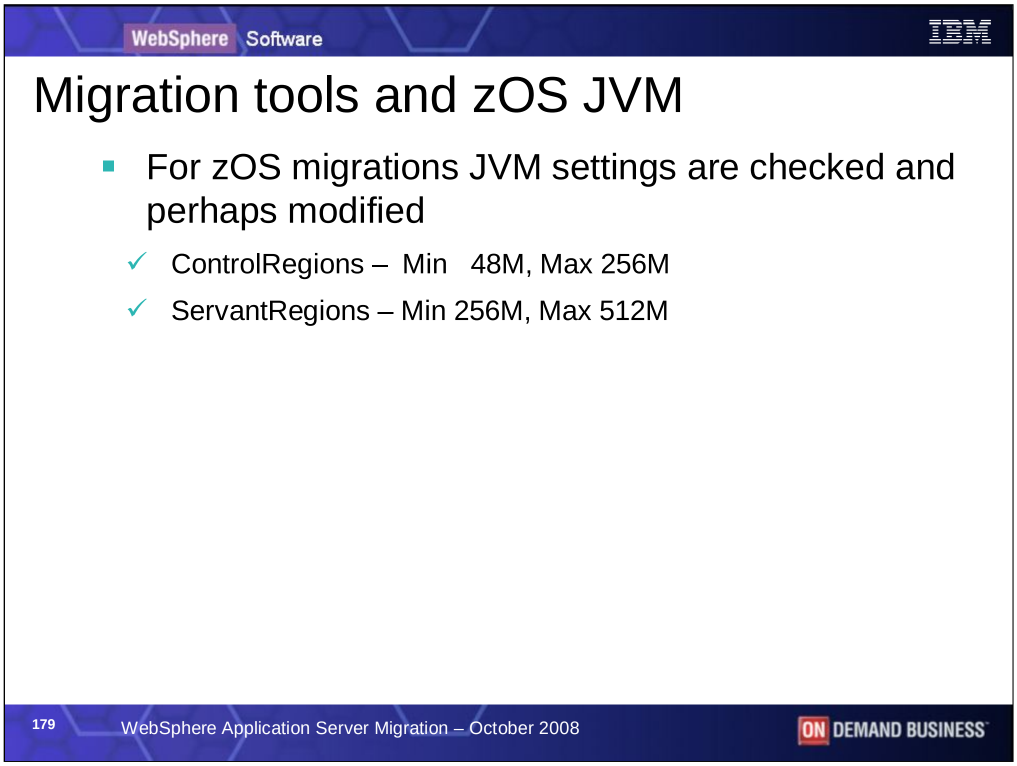
Migration tools disk usage

New October 2006

- For Deployment Manager and Standalone migrations it is simply what is on the previous page
- For Federated/Managed node Migrations there is one more complexity (Note, this is not applicable to v6.1 and later)
  - ✓ Each federated node requires more temporary storage in the Deployment Manager
  - ✓ This is created in *${DMGR_USER_INSTALL_ROOT}/temp/${FEDERATED_NODE}_migration_temp*
  - ✓ These are created but not deleted, they contain trace files if Service is required as a FirstFailureDataCapture
  - ✓ You can/should delete these after you have migrated the federated node

Federated migration is unique in that it is a two stage process. Half of the Federated migration takes place on the Federated node and the other half takes place on the Deployment Manager node. This two stage process requires a temporary work area located on the Deployment Manager node under the temp directory:
${DMGR_USER_INSTALL_ROOT}/temp/${FEDERATED_NODE}_migration_temp

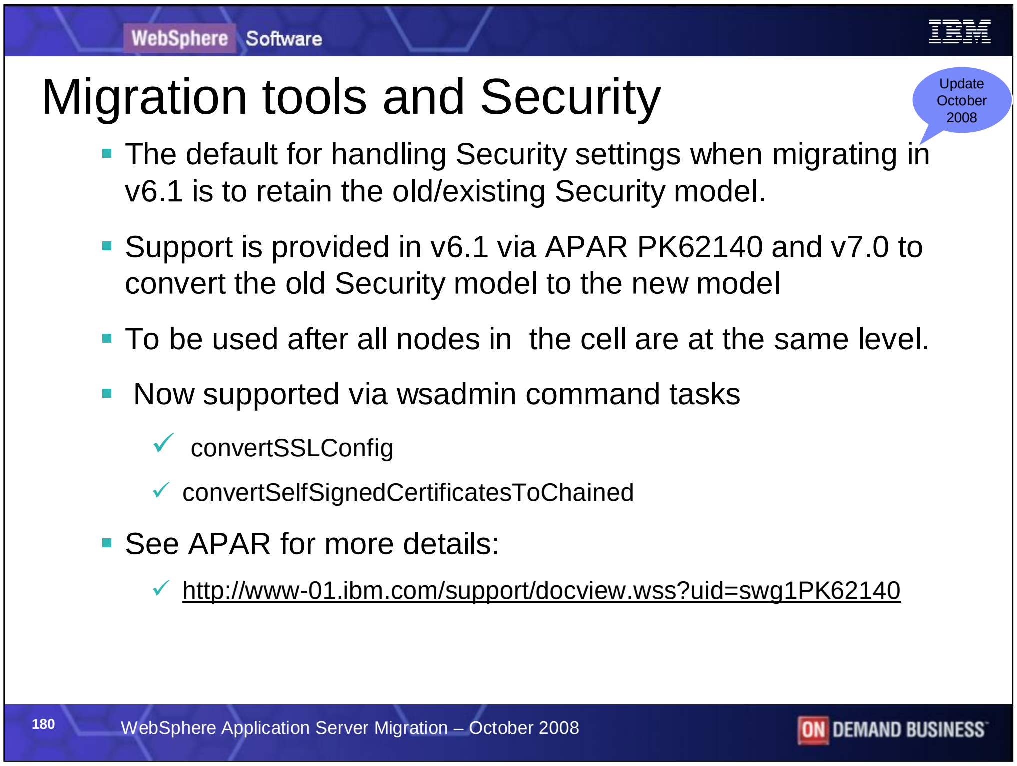
A rough estimate of the storage requirements of stage one (occurs on Federated node) can be derived in the same fashion as described in the previous slide. In stage two tracing is forced on and cannot be disabled. Trace files can grow quite large (In the ballpark of 200MBs for a single run). Federated migration will create this directory, but will not delete this directory; because by the time the Migration process knows enough to know that it succeeded it no longer has access to this temp directory. This may cause storage issues if you run multiple Federated migrations on different nodes without cleaning up the temp directory (Each node will get its own temporary work area located on the Deployment Manager node).



Migration tools and zOS JVM

- For zOS migrations JVM settings are checked and perhaps modified
  - ✓ ControlRegions – Min 48M, Max 256M
  - ✓ ServantRegions – Min 256M, Max 512M

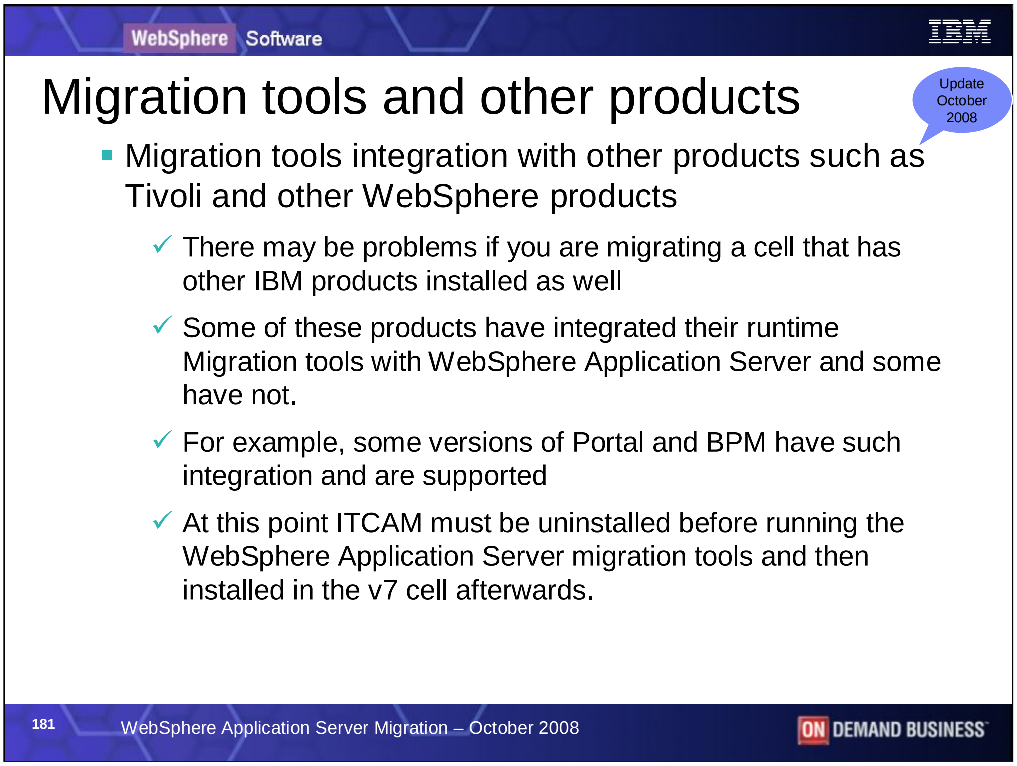
The new java runtimes minimum requirements are met. Existing JVM settings from the old configuration may not be optimal, These settings can be changed by fine tuning but for starting points these are the recommended settings.



Migration tools and Security

Update October 2008

- The default for handling Security settings when migrating in v6.1 is to retain the old/existing Security model.
- Support is provided in v6.1 via APAR PK62140 and v7.0 to convert the old Security model to the new model
- To be used after all nodes in the cell are at the same level.
- Now supported via wsadmin command tasks
  - ✓ convertSSLConfig
  - ✓ convertSelfSignedCertificatesToChained
- See APAR for more details:
  - ✓ http://www-01.ibm.com/support/docview.wss?uid=swg1PK62140



Migration tools and other products

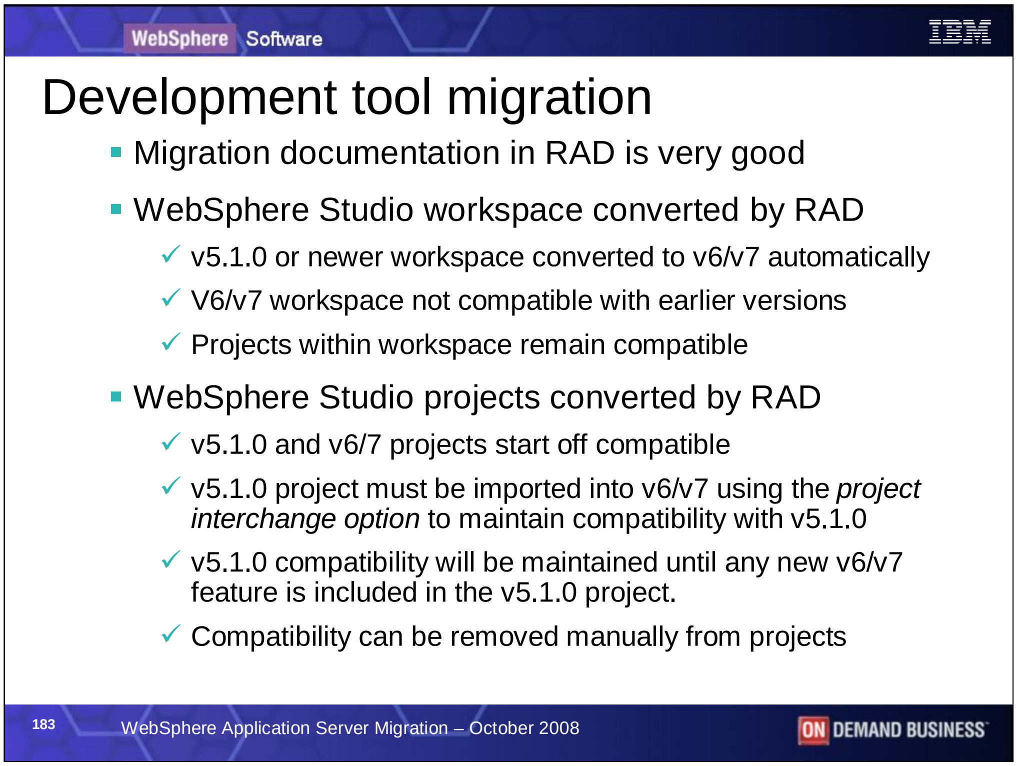
Update October 2008

- Migration tools integration with other products such as Tivoli and other WebSphere products
  - ✓ There may be problems if you are migrating a cell that has other IBM products installed as well
  - ✓ Some of these products have integrated their runtime Migration tools with WebSphere Application Server and some have not.
  - ✓ For example, some versions of Portal and BPM have such integration and are supported
  - ✓ At this point ITCAM must be uninstalled before running the WebSphere Application Server migration tools and then installed in the v7 cell afterwards.

WebSphere Software

RAD/RSA migration support

October 2008

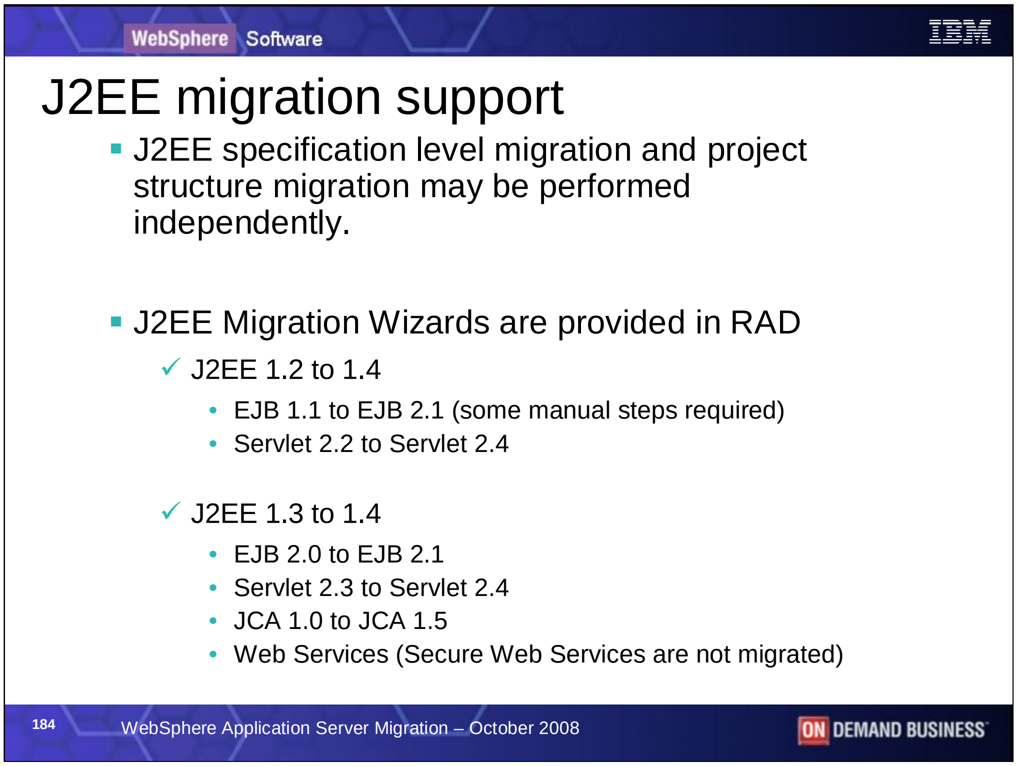


Development tool migration

- Migration documentation in RAD is very good
- WebSphere Studio workspace converted by RAD
  - ✓ v5.1.0 or newer workspace converted to v6/v7 automatically
  - ✓ V6/v7 workspace not compatible with earlier versions
  - ✓ Projects within workspace remain compatible
- WebSphere Studio projects converted by RAD
  - ✓ v5.1.0 and v6/7 projects start off compatible
  - ✓ v5.1.0 project must be imported into v6/v7 using the *project interchange option* to maintain compatibility with v5.1.0
  - ✓ v5.1.0 compatibility will be maintained until any new v6/v7 feature is included in the v5.1.0 project.
  - ✓ Compatibility can be removed manually from projects

A nice feature of RAD is that it will convert your v5.1.x workspaces automatically to RAD compliant workspaces. Something to keep in mind is that v6 workspaces are not compatible with earlier versions. This information should be part of your migration planning(ex: you do not want to open your workspace in RAD if you still have development need to open the workspace in v5.1). This may sound like a major set back, given the question “How do we do parallel development?”. To accommodate this, RAD will maintain project compatibility with v5.1.x. This project compatibility feature stays into effect until the project makes use of any new RAD features at which point compatibility with v5.1 will be broken. Again this should be part of your migration planning(ex: It must be communicated to development to not use any new features of RAD until such time that all of development is working on RAD and off v5.1).

More information on this is described in this slide can be found in the following Redbook in Chapter 2:
http://www.redbooks.ibm.com/abstracts/sg246369.html



J2EE migration support

- J2EE specification level migration and project structure migration may be performed independently.

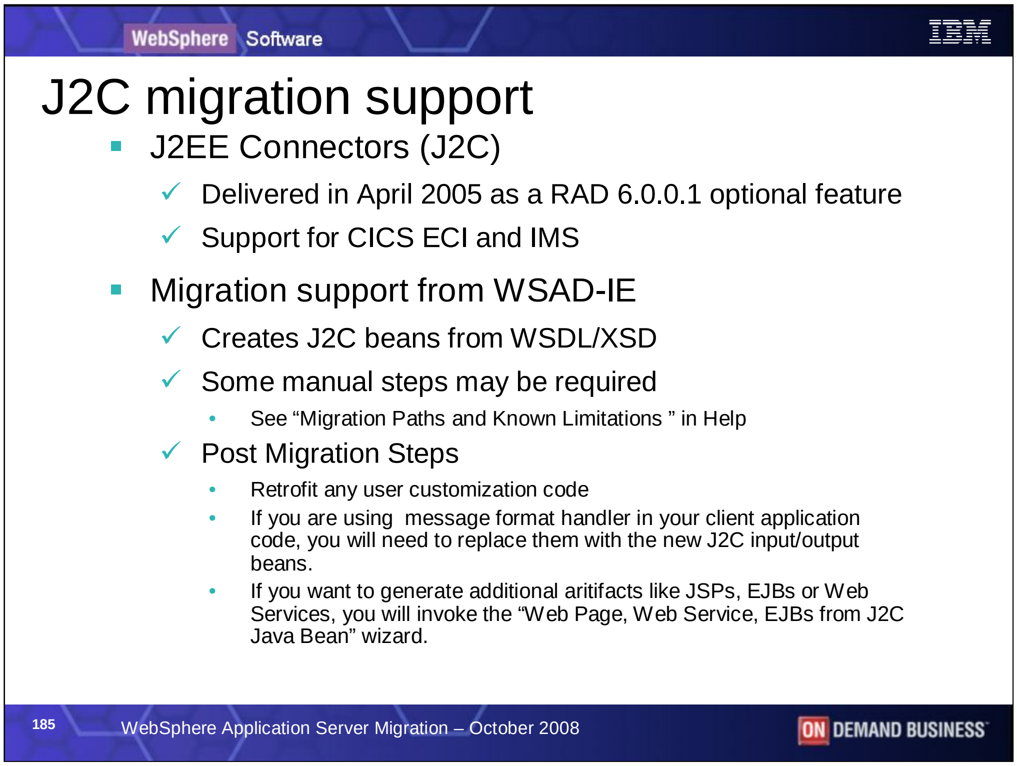
- J2EE Migration Wizards are provided in RAD
  - ✓ J2EE 1.2 to 1.4
    - EJB 1.1 to EJB 2.1 (some manual steps required)
    - Servlet 2.2 to Servlet 2.4
  - ✓ J2EE 1.3 to 1.4
    - EJB 2.0 to EJB 2.1
    - Servlet 2.3 to Servlet 2.4
    - JCA 1.0 to JCA 1.5
    - Web Services (Secure Web Services are not migrated)

Project structure and J2EE specification level are not related, thus you can change the project structure(ex: move from v5.1 to RAD) without changing J2EE versions and vice versa.

RAD includes a few J2EE Migration Wizards that can help you with your J2EE application migration. These tools includes support for EJB migration, JSP and Servlet migration, JCA migration, Web Services migration. The J2EE Migration Wizards do quite a good job in migrating your application artifacts. In some circumstances the Wizards may not have enough information to make the migration happen in which case it will either document it in the code or log file. This documentation can be used by you to finish the job.

It should be noted, that we recommend that you wait on performing application migration until after you have completed your environment migration if possible, it is understood that this is not always possible. If problems do happen during your WebSphere Application Server v6.x migration it helps to reduce the number of variables when determining what the issue is.

More information on these J2EE Migration Wizards can be found in Redbook:
http://www.redbooks.ibm.com/abstracts/sg246449.html



WebSphere Software

IBM

J2C migration support

- J2EE Connectors (J2C)
  - ✓ Delivered in April 2005 as a RAD 6.0.0.1 optional feature
  - ✓ Support for CICS ECI and IMS
- Migration support from WSAD-IE
  - ✓ Creates J2C beans from WSDL/XSD
  - ✓ Some manual steps may be required
    - See “Migration Paths and Known Limitations ” in Help
  - ✓ Post Migration Steps
    - Retrofit any user customization code
    - If you are using message format handler in your client application code, you will need to replace them with the new J2C input/output beans.
    - If you want to generate additional aritifacts like JSPs, EJBs or Web Services, you will invoke the “Web Page, Web Service, EJBs from J2C Java Bean” wizard.

185 WebSphere Application Server Migration – October 2008

ON DEMAND BUSINESS

RAD v6.0.0.1 introduced J2EE Connectors(J2C) in April 2005 as an optional feature which includes support for CICS ECI and IMS. Support has been added to RAD as a J2C Migration Wizard which will help you migrate your WSAD-IE applications to J2C applications.

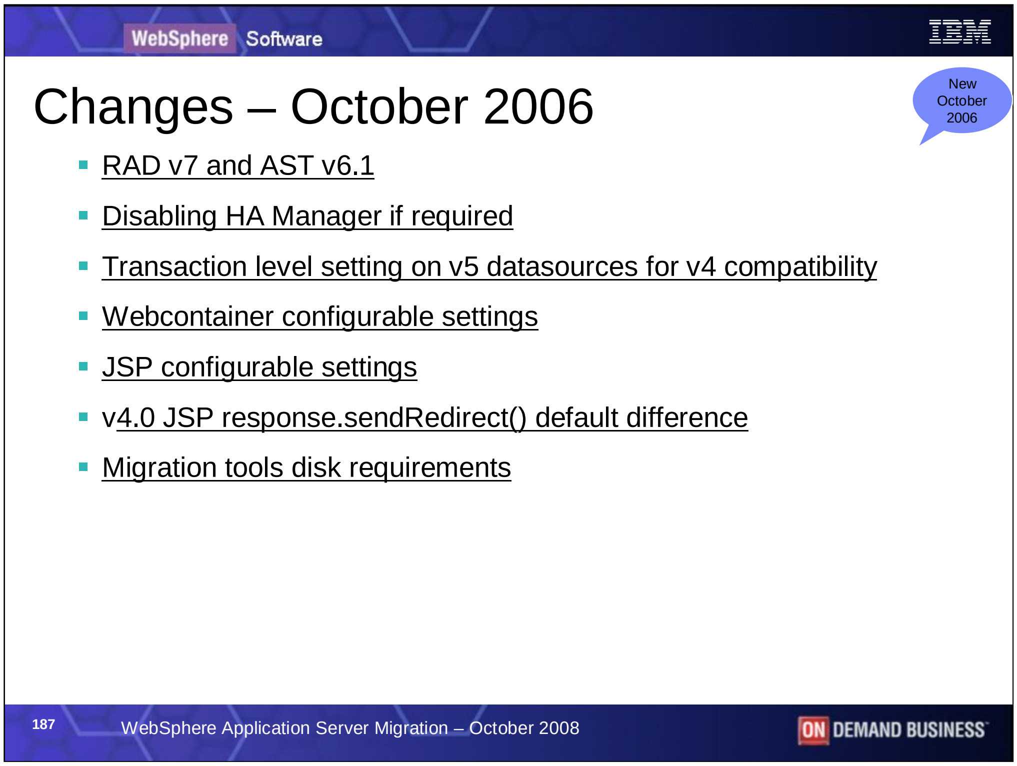
The J2C Migration wizard enables you to migrate the functionality of your WebSphere Studio Application Developer Integration Edition projects into the Rational Application Developer environment in a few simple steps. The migration tool extracts the information from within the WSDL files and uses this information to create a J2C application.

The following link provides more details on this topic:
http://publib.boulder.ibm.com/infocenter/radhelp/v6r0m1/topic/com.ibm.etools.j2c.doc/topics/tmigprojj2cmigrationtool.html

WebSphere Software

Change History

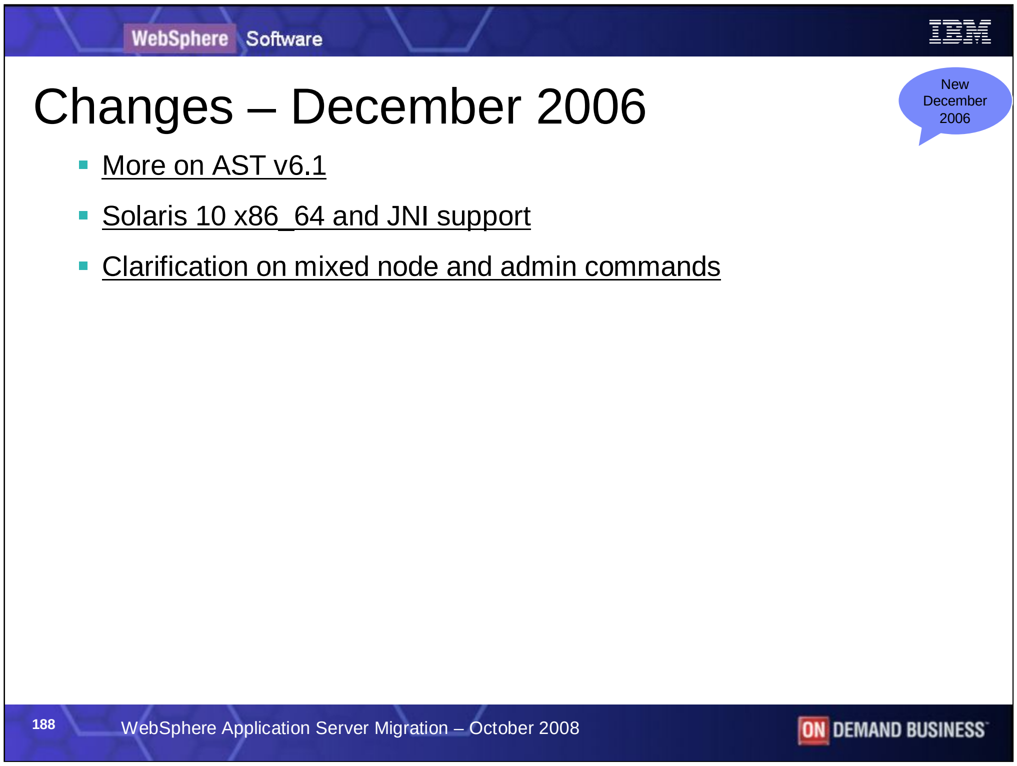
October 2008



Changes – October 2006

New
October
2006

- RAD v7 and AST v6.1
- Disabling HA Manager if required
- Transaction level setting on v5 datasources for v4 compatibility
- Webcontainer configurable settings
- JSP configurable settings
- v4.0 JSP response.sendRedirect() default difference
- Migration tools disk requirements



Changes – December 2006

New
December
2006

- More on AST v6.1
- Solaris 10 x86_64 and JNI support
- Clarification on mixed node and admin commands

Changes – March 2007

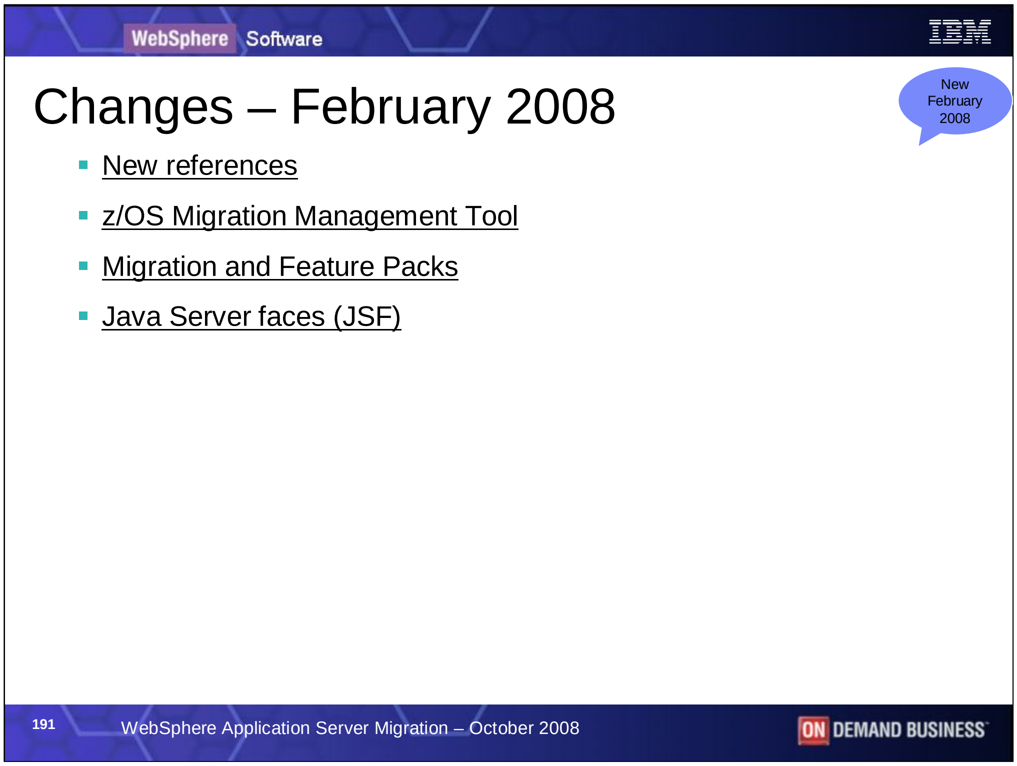
New March 2007

- Online availability
- Silent Install response files have changed
- Some scripts may break in v6.1 if you reference jars directly
- New Security model may affect scripts
- Migration tools and memory usage
- Migration tools and open connections

Changes – November 2007

New
November
2007

- JSPs using Java 5 specific language features
- JNDI Direct lookup undeprecation



Changes – February 2008

New
February
2008

- New references
- z/OS Migration Management Tool
- Migration and Feature Packs
- Java Server faces (JSF)

Changes – May 2008

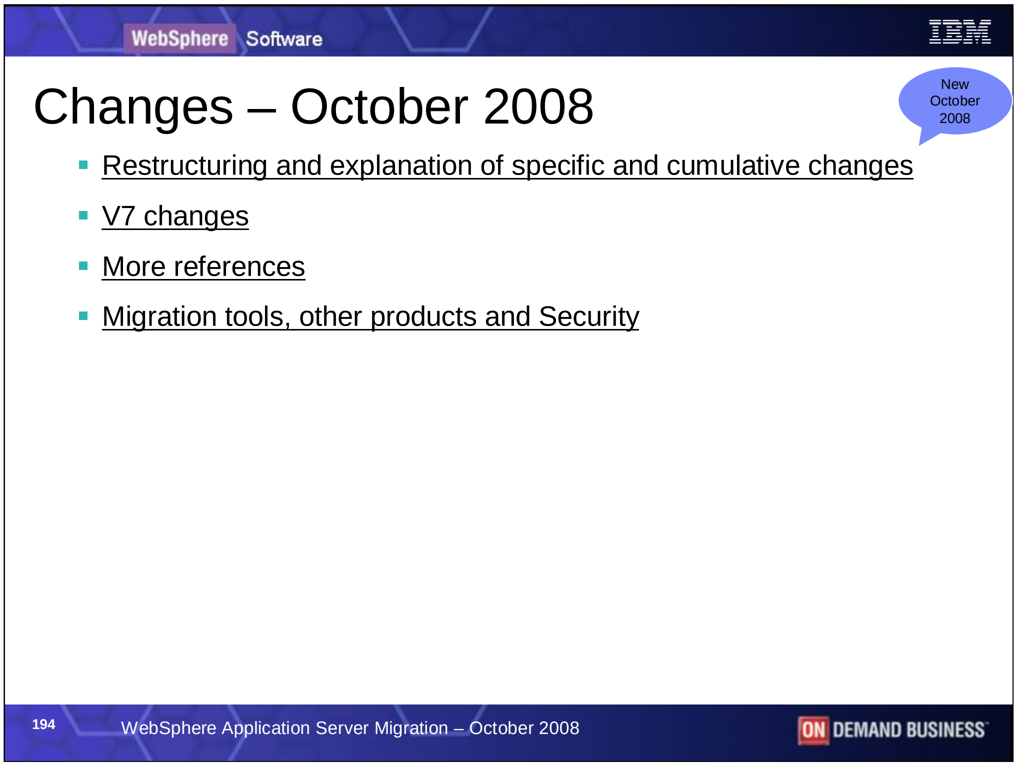
New May 2008

- More Support statement updates
- Expanded explanations of JSP and Servlet changes in J2EE 1.4
- Client access to datasources
- Auto-encoding change starting in v5.0
- Restriction on starting threads in containers

Changes – August 2008

New
August
2008

- Additional explanation for using migration tool options for large topologies
- Web Services incompatibility in v6.0 and later



Changes – October 2008

New
October
2008

- Restructuring and explanation of specific and cumulative changes
- V7 changes
- More references
- Migration tools, other products and Security