IBM WebSphere Application Server Network Deployment for IBM i, Version 8.5

Migrating, coexisting, and interoperating

Note

Before using this information, be sure to read the general information under “Notices” on page 123.

Compilation date: June 12, 2012

Contents

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

- To send comments on articles in the WebSphere Application Server Information Center
  1. Display the article in your Web browser and scroll to the end of the article.
  2. Click on the **Feedback** link at the bottom of the article, and a separate window containing an email form appears.
  3. Fill out the email form as instructed, and submit your feedback.
- To send comments on PDF books, you can email your comments to: **wasdoc@us.ibm.com**.

  Your comment should pertain to specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. Be sure to include the document name and number, the WebSphere Application Server version you are using, and, if applicable, the specific page, table, or figure number on which you are commenting.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer. When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about your comments.

Using this PDF

Links

Because the content within this PDF is designed for an online information center deliverable, you might experience broken links. You can expect the following link behavior within this PDF:

- Links to Web addresses beginning with http:// work.
- Links that refer to specific page numbers within the same PDF book work.
- The remaining links will *not* work. You receive an error message when you click them.

Print sections directly from the information center navigation

PDF books are provided as a convenience format for easy printing, reading, and offline use. The information center is the official delivery format for IBM WebSphere Application Server documentation. If you use the PDF books primarily for convenient printing, it is now easier to print various parts of the information center as needed, quickly and directly from the information center navigation tree.

To print a section of the information center navigation:

1. Hover your cursor over an entry in the information center navigation until the **Open Quick Menu** icon is displayed beside the entry.
2. Right-click the icon to display a menu for printing or searching your selected section of the navigation tree.
3. If you select **Print this topic and subtopics** from the menu, the selected section is launched in a separate browser window as one HTML file. The HTML file includes each of the topics in the section, with a table of contents at the top.
4. Print the HTML file.

For performance reasons, the number of topics you can print at one time is limited. You are notified if your selection contains too many topics. If the current limit is too restrictive, use the feedback link to suggest a preferable limit. The feedback link is available at the end of most information center pages.

Chapter 1. Migration, coexistence, and interoperability

The goal of migration is to reconstruct the environment of your previous version of WebSphere® Application Server in a Version 8.5 environment. Coexistence allows you to create a mixed-version environment that is not in conflict and allows the nodes of all versions to start and run at the same time. Coexistence also facilitates rollback and allows one or the other version to run at one time. Interoperating is exchanging data between two coexisting product installations or between products on different systems.

Overview of migration, coexistence, and interoperability

The goal of migration is to reconstruct your earlier version of WebSphere Application Server in a Version 8.5 environment. Coexistence allows you to create a mixed-version environment that is not in conflict and allows the nodes of all versions to start and run at the same time. Coexistence also facilitates rollback and allows one or the other version to run at one time. Interoperating is exchanging data between two coexisting product installations or between products on different systems.

Note: These tools are intended for use with the full WebSphere Application Server profile; they are not required or supported for use with the Liberty profile.

IBM i

Introduction

WebSphere Application Server Version 8.5 can coexist with Version 6.1 and above. Depending on the previous version of WebSphere Application Server, port conflicts might exist that must be resolved. Read “Coexistence support” on page 95 and Chapter 13, “Configuring port settings,” on page 99 for more information.

WebSphere Application Server Version 8.5 migration leverages the existing configuration and applications and changes them to be compatible with the WebSphere Application Server Version 8.5 environment. Existing application components and configuration settings are applied to the Version 8.5 environment during the migration process.

If you use an earlier version of WebSphere Application Server, the system administrator might have fine-tuned various application and server settings for your environment. It is important to have a strategy for migrating these settings with maximum efficiency.

You can perform incremental migration of your WebSphere Application Server Version 6.1 or above configuration by running the migration tools multiple times, each time specifying a different set of profiles.

Incremental migration of your WebSphere Application Server usually involves operating your system in a mixed cell release environment. Migration in this environment involves node migrations at various times and as such, may result in mixed cells running for extended periods of time until migration is complete.

A cell can contain nodes from different WebSphere Application Server versions. A WebSphere Application ServerVersion 8.5 mixed cell can contain nodes that support WebSphere Application Server Version 8.5 and Version 6.1 or above. A mixed cell environment can exist in two ways:

1. You perform incremental node migration of your existing system.
   a. You migrate the deployment manager to Version 8.5. The deployment manager has to be at the level of the highest node version. If you have nodes of the previous version, then this migration of the deployment manager creates a mixed cell at the highest version of WebSphere Application Server.
   b. Then when you migrate one node at a time to this new highest version, the cell becomes a cell at the highest version of WebSphere Application Server.

Note: This cell cannot be at a higher version than the deployment manager.

2. You migrate the deployment manager to Version 8.5 and then federate older version nodes to the new version deployment manager. This form of migration is supported for only Version 6.1 and above nodes.
   a. First, you migrate the deployment manager to Version 8.5. The deployment manager has to be at the level of the highest node version.
   b. You then can federate nodes from Version 6.1 and above to the new highest deployment manager version.

   gotcha: This method of incremental migration leaves your system in a mixed cell environment with nodes administered by a Version 8.5 deployment manager. Your migration planning should eventually include migrations of all nodes to the Version 8.5 level to ensure consistent administration of the nodes.

Existing functions continue to work in a mixed cell environment. You should be able to perform reasonable operations, such as run existing applications, perform management operations, such as addNode, create mixed cluster, configure the system, call Mbeans, and deploy applications. New function support in a mixed cell environment can be decided on a case by case basis - based on function, priority and available resources.

gotcha: **gotcha:** When running in a mixed-cell environment, clients might suddenly encounter a situation where the port information about the cluster members of the target cluster has become stale. This situation most commonly occurs when all of the cluster members have dynamic ports and are restarted during a time period when no requests are being sent. The client process in this state will eventually attempt to route to the node agent to receive the new port data for the cluster members, and then use that new port data to route back to the members of the cluster.

If any issues occur that prevent the client from communicating with the node agent, or that prevent the new port data being propagated between the cluster members and the node agent, request failures might occur on the client. In some cases, these failures are temporary. In other cases you need to restart one or more processes to resolve a failure.

To circumvent the client routing problems that might arise in these cases, you can configure static ports on the cluster members. With static ports, the port data does not change as a client process gets information about the cluster members. Even if the cluster members are restarted, or there are communication or data propagation issues between processes, the port data the client holds is still valid. This circumvention does not necessarily solve the underlying communication or data propagation issues, but removes the symptoms of unexpected or uneven client routing decisions.

Migration involves the following main steps:

1. Test your applications in a non-production WebSphere Application Server Version 8.5 environment, and make any changes to the applications that are necessary to ensure that they run in that environment.
2. Migrate those applications and your configuration to Version 8.5.

   You can complete this task by using the migration tools that are included with the product.

Use the migration tools to migrate applications and configuration information to the new version as described in Chapter 4, “Migrating product configurations,” on page 21. Read “Migrating product configurations with migration tools” on page 29 for more information.

Important reference articles for this migration include the following articles:

- manageprofiles command.
- Migrating servers from multi-broker replication domains to data replication domains
- Comparison of multi-broker versus data replication domains

- Migrating to Version 3 of the UDDI registry
- Migrating a complete gateway configuration

If you neither migrate nor coexist with an earlier version of WebSphere Application Server, you are choosing to ignore the previous installation and you can run only one version at a time because of conflicting default port assignments. It is possible for both versions to run at the same time without conflict if you use non-default ports in one version. To set up coexistence with WebSphere Application ServerVersion 6.1 or above, ensure that unique ports are selected during profile creation for the Version 8.5 installation. To set up coexistence with an existing installation of Version 8.5, select the **Install a new copy of the V8.5 Application Server product** radio button during installation.

You can resolve conflicting port assignments by specifying port assignments for coexistence during profile creation, by **wsadmin** scripting, or by using the **Servers > Application Servers > server1 > Ports** administrative console page to ensure that WebSphere Application Server Version 8.5 can run with an earlier version. Read the "Wsadmin tool" article in the information center for more information on **wsadmin** scripting.

Coexistence processing changes the following configuration files:

- `virtualhosts.xml`
- `serverindex.xml`

Read Port number settings in WebSphere(r) Application Server versions for more information.

Consider the following issues in a migration or coexistence scenario:

- Conflicting context roots when attempting to share the same Web server.

  Follow the procedure in Chapter 9, “Migrating web server configurations,” on page 85 to learn how to configure a web server for sharing between WebSphere Application Server versions.

Migrating and coexisting application servers

Migrating involves collecting the configuration information from a previous release of a WebSphere Application Server and merging it into a configuration for a new release. Coexisting involves running a new release of a WebSphere Application Server and an earlier release simultaneously on the same machine.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6. For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

The migration tools basically save the existing WebSphere configurations and user applications in a backup directory and then process the contents of this backup directory to migrate the configurations and your applications from previous WebSphere Application Server releases to the latest release.

If you have a previous version of WebSphere Application Server, you must decide whether to migrate the configuration and applications of the previous version to the new version.

Migration does not uninstall the previous version.

- For stand-alone application server migrations and for deployment-manager migrations in which you do not choose to disable the previous deployment manager during migration, the earlier release is still functional.

- For federated-node migrations and for deployment-manager migrations in which you do choose to disable the previous deployment manager during migration, the earlier release is disabled after migration completes successfully. You can re-enable the earlier version using the **migrationDisablementReversal.jacl** script.

If you run two different versions of the application server at the same time, the two versions are *coexisting*. For example, if Version 6.1 and 8.5 application servers are running on the same machine, they are coexisting.

IBM i To support coexistence, you must either use the -portBlock and -replacePorts options when you migrate a profile or you must resolve port conflicts manually so that the two releases do not attempt to use the same ports. Any ports bound when the first profile starts will prevent the second profile from starting because the port is in use. No port changes are required if only one release of the profile is active at any given time.

For help in troubleshooting problems when migrating, read Chapter 14, “Troubleshooting migration,” on page 109.

About this task

For information on migrating to Version 8.5, read Chapter 4, “Migrating product configurations,” on page 21. For more information on coexistence among releases, read “Coexistence support” on page 95.

Procedure

1. Update product prerequisites and corequisites to supported versions.

   Refer to the IBM® WebSphere Application Server supported hardware, software, and APIs site for current requirements.
2. Install the Version 8.5 product.

   IBM i Read the "Task overview: installing" article in the information center for more information.
3. Migrate your WebSphere Application Server Version 6.1 or above product configuration to Version 8.5.

   IBM i

   You have the choice between migrating your configuration automatically using the migration tools or manually.
   - Use the migration tools to automatically migrate your configuration.

     Read “Migrating product configurations with migration tools” on page 29 for more information.

     The following two WebSphere Application Server, Network Deployment migration scenarios are possible:
     - Automated migration with all-node upgrade

       In this scenario, you use the migration tools to migrate the deployment manager as well as all of its federated nodes.

       There are the following advantages and considerations with this approach:
       - Advantages
         - You copy the old configuration automatically.

           This includes all resource definitions, virtual host definitions, security settings, cluster definitions, and so forth.
         - You recreate the same exact Version 6.1 or above configuration in Version 8.5, including the node definitions, server definitions, and deployed applications by default.
         - You can enable support for script compatibility.

           Read “WASPostUpgrade command” on page 37 for more information.
       - Considerations

- You should have a good idea of how long it will take to migrate the configuration before you begin.
- You should migrate within a maintenance window.

– Automated migration with mixed-node utilization

This scenario involves the following activities:

- You use the migration tools to migrate the deployment manager only.
- You add Version 8.5 nodes.
- You move your applications to Version 8.5 as they are tested on Version 8.5.
- You remove Version 6.1 or above nodes from the cell when they are no longer needed.

There are the following advantages and considerations with this approach:

- Advantages
  - You copy the old configuration automatically.

    This includes all resource definitions, virtual host definitions, security settings, cluster definitions, and so forth.
  - You recreate the same exact Version 6.1 or above configuration in Version 8.5, including the node definitions, server definitions, and deployed applications by default.
  - You can have a mixed-node configuration.
  - You can enable support for script compatibility.

    Read “WASPostUpgrade command” on page 37 for more information.
  - You can move applications iteratively.
- Considerations
  - You should have a good idea of how long it will take to migrate the configuration before you begin.
  - You should migrate within a maintenance window.

- Manually migrate your configuration.

  Migrating your configuration manually involves the following activities:

  – You start with a clean slate and build up a new environment for Version 8.5.

  – Ideally, you would use an existing set of administration scripts to set up the complete Version 8.5 environment.

  – You move your applications to Version 8.5 as they are tested on Version 8.5.

  – You remove a Version 6.1 or above cell when it is no longer needed.

  Consider the following points related to manually migrating your configuration:

  – Advantages
    - You can reuse the scripts for maintenance, replication, and disaster recovery.
    - You can easily refactor the topology if you desire.

  – Considerations
    - A complete set of administration scripts is a significant investment.
    - You must address script incompatibilities and changes before you migrate.
    - You cannot have a mixed-node configuration.

4. Migrate web server plug-ins as described in Chapter 9, “Migrating web server configurations,” on page 85.
5. Optional: Set up multiple versions of WebSphere Application Server to coexist. IBM i

   No runtime conflicts can exist for multiple instances and versions of WebSphere Application Server if they are going to run at the same time on the same machine. Potential conflicts can occur with your port assignments. Read “Port number settings” on page 100 for more information.

Premigration considerations

Before you begin the process of migrating to WebSphere Application Server Version 8.5, there are some considerations of which you need to be aware.

IBM i

Considerations for AIX, HP-UX, IBM i, Linux, Solaris, and Windows operating systems

Consider the following information before you migrate Application Server:

- After you install WebSphere Application Server Version 8.5, you might want to build a complete WebSphere Application Server, Network Deployment cell configuration and verify that it works correctly before you attempt to migrate an existing cell or node.

  This process ensures that your system has all of the necessary prerequisites and supports the new level of WebSphere Application Server.
- Before you perform the migration, evaluate the items deprecated in WebSphere Application Server Version 8.5.

  For more information, read the "Deprecated and removed features" article in the information center.
- High-availability manager (HAM) and core-group functionality are included in WebSphere Application Server Version 6.1 and later.

  Read the "Core group migration considerations" article in the information center for core-group configuration and topology considerations that might impact your migration from Version 6.1 or above to Version 8.5.

  Note: In most cases the recommended number of servers in a core group should not exceed 50. You receive a warning message when the migration tools add a server that exceeds the recommended upper limit.
- Before you migrate to Java Standard Edition (SE) Development Kit (JDK) 6 from JDK 5 or JDK 1.4 , review your applications for necessary changes based on the Java specification. IBM® WebSphere® SDK Java Technology Edition Version 6.0 is installed by default with WebSphere Application Server Version 8.5. Optionally, you can use the Installation Manager to install IBM WebSphere SDK Java Technology Edition Version 7.0.

  Note: Do not enable SDK Java Technology Edition Version 7.0 unless all nodes are migrated to Version 8.5.

  Read “Migrating API and specifications” on page 10 for more information.
- When migrating a cell with multiple nodes, the applications must remain at the lowest JDK level until all nodes are migrated.
- The migration articles in this information center assume that WebSphere Application Server Version 8.5 is being installed in an environment where it must coexist with previous versions of WebSphere Application Server.

  Consider the following items when planning to enable coexistence:
  - Update prerequisites to the levels required by WebSphere Application Server Version 8.5.

    Previous versions of WebSphere Application Server continue to run at the higher prerequisite levels.
  - Review the ports that have been defined to ensure that the WebSphere Application Server Version 8.5 installation does not conflict.

    Read “Port number settings” on page 100 for default port information.

  Read Chapter 11, “Running multiple application server versions,” on page 95 for more information.
- Consider the following information if you are planning to have any mixed-release cells:

  - You can upgrade a portion of the nodes in a cell to WebSphere Application Server Version 8.5 while leaving others at the previous release level. This means that, for a period of time, you might be administering servers that are at the previous release level and servers that are running the newer release in the same cell.

    In this mixed-release environment, some restrictions on what you can do with servers at the previous release level might exist. For details, read the "Creating application servers" article in the information center.
- WebSphere Application Server Version 8.5 migration converts HTTP transports to channel-framework web container transport chains.

  For more information on WebSphere Application Server Version 8.5 transport support, read the following articles in the information center:
  - Configuring transport chains
  - HTTP transport channel settings
  - Transport chains
- The following restrictions apply to a deployment manager or job manager migration:
  - The Version 8.5 cell name must match the cell name in the Version 6.1 or above configuration.

    If you create a profile with a new cell name, the migration will fail.
  - Either one or the other of the following options must be true:
    - The Version 8.5 deployment manager or job manager node name must be the same as the Version 6.1 or above deployment manager or job manager node name.
    - The Version 8.5 deployment manager or job manager node name must be different from every node name in the Version 6.1 or above configuration.

    Otherwise, the migration fails with the following message:

    ```
    MIGR0488E: The deployment manager node name in the new configuration ({0})
    cannot be the same as a nodeagent node in the old configuration.
    ```
- When migrating a federated node, the WebSphere Application Server Version 8.5 cell name and node name must match their respective Version 6.1 or above names.
- If you create a profile that does not meet the migration requirements such as naming requirements, you can remove the previous profile and create a new one rather than uninstalling and reinstalling the WebSphere Application Server Version 8.5 product.
- If you migrate a node to WebSphere Application Server Version 8.5 and then discover that you need to revert back to Version 6.1 or above, read “Rolling back environments” on page 60.
- If you have a web services gateway running on a WebSphere Application Server Version 6.1 or above application server that is part of a WebSphere Application Server, Network Deployment cell and you want to migrate the cell from a Version 6.1 or above to a Version 8.5 deployment manager, you must first preserve the gateway configuration as described in the "Coexisting with previous gateway versions" article in the information center.
- If you have a web services gateway running on a WebSphere Application Server Version 6.1 or above application server that is part of a WebSphere Application Server, Network Deployment cell and you want to migrate the cell from a Version 6.1 or above to a Version 8.5 deployment manager, you must first preserve the gateway configuration as described in the information center.
- The migration tools create a migration backup directory containing a backup copy of the configuration from the previous version. The following guidelines might help you to determine the amount of file-system space that this directory might require:
  - If you are migrating from Version 6.1 or above, the space available for this directory must be at least the size of the configuration directory and applications from the previous profile.
- If you use the migration tools to create more than one Version 8.5 target profile on the same host or installation instance and you use the default port settings, there is a chance that the target profiles will share the same ports for some of the new Version 8.5 port definitions. This will cause startup problems if both of the migrated profiles are used.

If you are migrating two or more profiles that reside on the same host or installation instance, perform the following actions for each additional target profile:

1. Before using the migration tools, use the Profile Management tool or **manageprofiles** command to create the target profile and make sure that you select unique ports rather than using the default ports.
2. When you use the migration tools, select the target profile rather than letting the tools create it.

- The amount of storage that your system requires during migration to Version 8.5 depends on your environment as well as on the migration tool that you are using.
  - **WASPreUpgrade storage requirements**
    - **Location:** Backup directory specified as a parameter of the **WASPreUpgrade** command
    - **Amount:** For an estimate of your storage requirements when using this command, add the following amounts.
      - IBM i Size of the following items for the previous profile or instance that you are migrating:
        - *profile_root*/installableApps directory
        - *profile_root*/installedApps directory
        - *profile_root*/config directory
        - *profile_root*/properties directory
        - Shared libraries referenced in the libraries.xml configuration files
        - Resource adapter archive (RAR) files referenced in the resources.xml configuration files
      - If trace is enabled, which is the default, up to 200 MB (depending on the size and complexity of your configuration)

    For more information about this command, read "WASPreUpgrade command" on page 34.
  - **WASPostUpgrade storage requirements**
    - For an estimate of your storage requirements when using this command, add the following amounts.
      - **Location:** New configuration relative to the new *profile_root* directory

        IBM i Size of the following items for the previous profile or instance that you are migrating:
        - *profile_root*/installableApps directory
        - *profile_root*/installedApps directory
        - *profile_root*/config directory
        - *profile_root*/properties directory
        - Shared libraries referenced in the libraries.xml configuration files
        - RAR files referenced in the resources.xml configuration files
      - **Location:** Backup directory specified as a parameter of the **WASPreUpgrade** and **WASPostUpgrade** commands

        If trace is enabled, which is the default, up to 1 GB (depending on the size and complexity of your configuration)

    For more information about this command, read "WASPostUpgrade command" on page 37.
- If you use isolated data repositories—specifically, nonshared data repositories such as transaction logs for SIB and IBM Cloudscape or Apache Derby databases—and you migrate from a previous release, your existing databases and transaction logs are saved when the WASPreUpgrade tool is run. Any database changes that you make after the WASPreUpgrade tool is run will not be reflected in the migrated environment.
  - If you have mission-critical information that is stored in these local data repositories, you should safely shut down all servers that interact with those repositories before attempting migration. Those servers should remain offline until the migration has been successfully completed or rolled back.

- If you make multiple attempts at migration, either because of unexpected rollback or to apply fixes, rerun the WASPreUpgrade tool so that any changes to your isolated data repositories are reflected in the migrated environment.

  After the migration is complete or you have rolled back to the previous version, you can restart the servers that interact with these isolated data repositories.

- Before you migrate an Apache Derby database, ensure that any application servers hosting applications that are using the Apache Derby database are closed. Otherwise, the Apache Derby migration fails.
- You should be aware of the following rules related to migrating security domains:
  - If you migrate a deployment manager that has a security domain with a cell-level scope, the migration tools take the following actions:
    - Migration creates a domain in the new configuration called *PassThroughToGlobalSecurity* if it does not already exist.
    - Migration adds a cluster mapping to the new configuration for all clusters that existed in the old configuration.
      - Clusters that only existed in the Version 8.5 deployment-manager configuration before migration do not have their mappings to PassThroughToGlobalSecurity changed.
        - If mappings for the Version 8.5 clusters exist before migration, they still exist after migration.
        - If no mappings for the Version 8.5 clusters exist before migration, they still do not exist after migration.
      - If a cluster exists in both the previous configuration and the Version 8.5 configuration before migration, the cluster in the new configuration is added to the PassThroughToGlobalSecurity domain and behaves like the cluster in the previous release.
    - Migration adds a bus mapping for any busses that exist in a migrated Version 6.1.x configuration.

      Bus mappings are updated following the same rules as those for cluster mapping that are described above.
    - Administrative servers (deployment manager) are not added to the PassThroughToGlobalSecurity domain.
  - If you migrate a federated node that has a security domain with a cell-level scope, the migration tools take the following actions:
    - Migration creates a domain in the new configuration called *PassThroughToGlobalSecurity* if it does not already exist.
    - Migration adds a server-level mapping to the PassThroughToGlobalSecurity domain for all non-clustered servers in the old node's configuration.
      - Servers on the node that is being migrated that are part of a cluster do not receive entries in the PassThroughToGlobalSecurity domain because this was addressed through a cluster mapping during deployment-manager migration.

        If you have removed that mapping, migration maintains that behavior.
      - Administrative servers (node agents) are not added to the PassThroughToGlobalSecurity domain.

  Read the "Security domains in a mixed-version environment" section of the "Multiple security domains" article in the information center.
- If you updated your java.security file in the previous version of WebSphere Application Server, ensure that your updates are located in the migrated java.security file, which is located in `V85WAS_HOME/java/jre/lib/security/java.security`.
- The process for disabling credential prompting has changed.

  To disable credential prompting in Version 8.5, configure the ipc.client.props to disable credential prompting before migrating from Version 6.1 to Version 8.5.
- During migration, some of your application metadata might be reset to the default and cause the application to function differently from what you expect.

If you installed an application in your old environment with **Use Metadata From Binaries** set to true and during that installation or a future update of the application you made a change to the application's metadata (such as JNDI resource references or database entries for example), the change might be lost when you migrate.

When **Use Metadata From Binaries** is set to true, the administrative code only updates the metadata in the binary EAR file. This option is not supported in a mixed cell; therefore, it is automatically turned to false as part of migration. When this happens, the expanded metadata in the configuration directories take precedence over the values in the binary EAR file. This causes the values from the original EAR file installation to take precedence over any updates that you might have made.

Perform one of the following actions to resolve this issue:

- Before migrating, update your applications in the old environment and set **Use Metadata From Binaries** to false. Ensure that the applications are functioning correctly with this new setting, and then run the migration.
- After migrating, update your applications and correct the metadata as required to allow the applications to function appropriately.

- After you use the migration tools to migrate to WebSphere Application Server Version 8.5, you might need to perform some actions that are not done automatically by the migration tools.
  - Examine any Lightweight Third-Party Authentication (LTPA) security settings that you might have used in WebSphere Application Server Version 6.1 or above, and verify that Version 8.5 security is set appropriately.

    Read the "Lightweight Third Party Authentication" article in the information center for more information.
  - Check the `WASPostUpgrade.log` file in the `logs` directory for details about any JavaServer Pages (JSP) objects that the migration tools did not migrate.

    If Version 8.5 does not support a level for which JSP objects are configured, the migration tools recognize the objects in the output and log them.
  - Verify the results of the automatic Apache Derby database migration, and manually migrate any Apache Derby databases that are not automatically migrated by the tools.

    Read “Migrating IBM Cloudscape or Apache Derby databases” on page 56 for more information.
  - WebSphere Application Server Version 8.5 does not include the WebSphere Connect JDBC driver for SQL Server. The WebSphereConnectJDBCDriverConversion tool is provided to convert data sources from the WebSphere Connect JDBC driver to the DataDirect Connect JDBC driver or the Microsoft SQL Server JDBC driver.

    Read “Migrating from the WebSphere Connect JDBC driver” on page 58 for more information.

Migrating API and specifications

Migrating application programming interfaces (APIs) and specifications involves moving to the current Java component level as well as to other technologies that WebSphere Application ServerVersion 8.5 supports. If your existing applications currently support different specification levels than are supported by this version of the product, it is likely that you must update at least some aspects of the applications to comply with the new specifications.

In many cases, IBM provides additional features and customization options that extend the specification level even further. If your existing applications use IBM extensions from earlier product versions, it might be necessary for you to perform mandatory or optional migration to use the same kinds of extensions in Version 8.5.

WebSphere Application Server Version 8.5 supports Java SE Development Kit (JDK) 7.

Notes on the use of Java SE Development Kit 6:

- WebSphere Application Server began supporting Java SE Development Kit (JDK) 6 in Version 8.0.

  Read JSR 270: Java SE 6 Release Contents and Java SE 6 for more information about JDK 6.
- In general, existing Version 6.x application binaries that were developed using JDK 1.4 and 5 are highly compatible and typically do not require modifications to run. However, recompilation of the JDK 1.4 or 5 applications at the JDK 6 level might necessitate modifications of the source code to conform to incompatible changes that are present in JDK 6. As part of your migration planning, you should review the JDK compatibility restrictions that are documented at Java SE 6 Release Notes: Compatibility.
- A mixed cell containing Version 6.x and Version 7.0 nodes requires that all application binaries deployed on Version 6.x remain at the lowest JDK level associated with the Version 6.x nodes. Although you can successfully migrate Version 6.x applications to Version 8.5, this is only meant to be a temporary state as you transition to Version 8.5. After you begin migration to Version 8.5, plan to complete the migration of the entire cell, update your tooling to Version 8.5, and update your applications to conform to JDK 6 requirements. Complete this action before any further application changes. After you have completely migrated your cell to Version 8.5, upgrade your application binaries to the JDK 6 level the next time that you make application modifications that require recompiling. This action might require source code changes to your application to conform to the JDK 6 API changes as documented.

Note: The Java Virtual Machine Debug Interface (JVMDI) and the Java Virtual Machine Profiler Interface (JVMPI) were deprecated in JDK 5 and removed in JDK 6. Read Java SE 6 Release Deprecated API for more information.

Read the "Specifications and API documentation" article in the information center for a summary of the specifications and API documentation supported in current and prior product releases.

For more information on the items deprecated in WebSphere Application ServerVersion 8.5, read the "Deprecated, stabilized, and removed features" article in the information center.

Chapter 2. How do I migrate, coexist, and interoperate?

Use the documentation provided to answer your questions about migration, coexistence, and interoperability.

Follow these shortcuts to get started quickly with popular tasks. When you visit a task, look for the **IBM Suggests** feature at the bottom of the page. Use it to find available tutorials, demonstrations, presentations, articles, IBM Redbooks, support documents, and more.

IBM i "Overview of migration, coexistence, and interoperability" on page 1

"Configuration mapping during product-configuration migration" on page 22

IBM i "Migrating product configurations with migration tools" on page 29

IBM i Chapter 4, "Migrating product configurations," on page 21

IBM i Chapter 9, "Migrating web server configurations," on page 85

Review the software and hardware prerequisites

Chapter 11, "Running multiple application server versions," on page 95

Chapter 13, "Configuring port settings," on page 99

Chapter 12, "Interoperating multiple application server versions," on page 97

Chapter 14, "Troubleshooting migration," on page 109

Chapter 3. Migrating applications to the Liberty profile

This section provides information on how to migrate applications to the Liberty profile.

Procedure

Migrate data access applications to the Liberty profile.

Migrating data access applications to the Liberty profile

For data access applications, you need to make configuration changes when you migrate a data source from the WebSphere Application Server full profile to the Liberty profile.

Procedure

- "Configuration differences between the full profile and Liberty profile: dataSource and jdbcDriver elements."
- "Configuration differences between the full profile and Liberty profile: connectionManager element" on page 16.
- "Migrating a DB2 data source to the Liberty profile" on page 17.
- "Migrating a Derby embedded data source to the Liberty profile" on page 19.

Configuration differences between the full profile and Liberty profile: dataSource and jdbcDriver elements

This page identifies some differences in configuration between `dataSource` in the Liberty profile and data sources in the full profile.

- Data source properties with different names
  - **ifxIFX_LOCK_MODE_WAIT**, which is **informixLockModeWait** in the full profile.
  - **supplementalJDBCTrace**, which is **supplementalTrace** in the full profile.
- Data source properties with different values
  - **beginTranForResultSetScrollingAPIs**, which is true by default in the Liberty profile
  - **beginTranForVendorAPIs**, which is true by default in the Liberty profile
  - **connectionSharing**, which is MatchOriginalRequest by default in the Liberty profile
  - **statementCacheSize**, which is 10 by default in the Liberty profile
- **connectionSharing** property of data source
  - The Liberty profile allows **connectionSharing** to be configured to either MatchOriginalRequest or MatchCurrentState. By default, it is **MatchOriginalRequest**.
  - The full profile allows **connectionSharing** to be configured in a finer grained manner, where individual connection properties can be matched based on the original connection request or current state of the connection. In the full profile, **connectionSharing** is a combination of bits representing which connection properties to match based on the current state of the connection. In the full profile, a value of 0 means match all properties based on the original connection request, and a value of -1 means to match all properties based on the current state of the connection. The default value for the full profile is 1, which means that the isolation level is matched based on the current state of the connection and all other properties are matched based on the original connection request.
- Time duration properties of data source

  Time duration properties can optionally be specified with units in the Liberty profile. For example,

```
<dataSource id="informix" jndiName="jdbc/informix" queryTimeout="5m"  ...>
        <properties.informix ifxIFX_LOCK_MODE_WAIT="120s" .../>
    </dataSource>
```

See Liberty profile: Configuration elements in the server.xml file for accepted time units and formats of `dataSource` element. Omitting the units in the Liberty profile is equivalent to the default units used in the full profile.

- Configuration for JDBC drivers
  - In the Liberty profile, you can take the same approach of configuring different `jdbcDriver` elements for XA capable and non-XA capable data source implementation classes, or you can use a single `jdbcDriver` element for both. Defining multiple jdbcDriver elements does not cause different class loaders to be used. In the Liberty profile, `jdbcDriver` elements always use the class loader of the shared library with which they are configured.
  - In the full profile, a JDBC provider is defined pointing to the JDBC driver JARs, compressed files and native files. Separate JDBC providers must be defined for XA capable and non-XA capable data source implementation classes.

  For some of the commonly used JDBC drivers, the Liberty profile infers the data source implementation class names based on the names the driver JARs. In these cases it is possible to omit the implementation class names. For example:

```
<jdbcDriver id="Derby" libraryRef="DerbyLib"/>
<library id="DerbyLib">
       <fileset dir="C:/Drivers/derby" includes="derby.jar" />
    </library>
```

  If you want to override the default implementation classes such as `javax.sql.DataSource`, `javax.sql.ConnectionPoolDataSource`, and `javax.sql.XADataSource`, you can use their optional properties.

  The following example overrides the default `javax.sql.XADataSource` and `javax.sql.ConnectionPoolDataSource` implementations that are selected by the Liberty profile:

```
<jdbcDriver id="Derby" libraryRef="DerbyLib"
      javax.sql.XADataSource="org.apache.derby.jdbc.EmbeddedXADataSource"
      javax.sql.ConnectionPoolDataSource="org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource"/>
   <library id="DerbyLib">
      <fileset dir="C:/Drivers/derby" includes="derby.jar" />
   </library>
```

  See Liberty profile: Configuration elements in the server.xml file for more information of `jdbcDriver`.

Configuration differences between the full profile and Liberty profile: connectionManager element

This page identifies some differences in configuration between `connectionManager` in the Liberty profile and connection pools in the full profile.

- Properties with different names
  - **`maxConnectionsPerThread`**, which is **`maxNumberofMCsAllowableInThread`** in the full profile.
  - **`maxIdleTime`**, which is **`unusedTimeout`** in the full profile.
  - **`maxPoolSize`**, which is **`maxConnections`** in the full profile.
  - **`minPoolSize`**, which is **`minConnections`** in the full profile.
- Time duration properties

  Time duration properties can optionally be specified with units in the Liberty profile. For example,

```
<connectionManager id="pool1" connectionTimeout="30s" reapTime="3m" maxIdleTime="30m"/>
```

  See Liberty profile: Configuration elements in the server.xml file for accepted time units and formats for the `connectionManager` element. If you do not specify time units in the Liberty profile, the same units are used as in the full profile.
- Differences in timeout values for immediate and never

  There are differences in the values used to represent immediate timeout and never (disabled) timeout.

  - The Liberty profile uses a value of 0 to represent immediate, whereas the full profile often uses -1 for immediate.
  - The Liberty profile uses a value of -1 to represent never (disabled), whereas the full profile often uses 0 for never (disabled).

  Specifically this applies to:
  - **agedTimeout**
  - **connectionTimeout**
  - **maxIdleTime**, which is **unusedTimeout** in the full profile
  - **reapTime**
- Purge Policy changes

  In the Liberty profile , there are three purge policy values: **EntirePool**, **FailingConnectionOnly**, and **ValidateAllConnections**.

  In the full profile, there are two purge policy values: **EntirePool** and **FailingConnectionOnly**, with a second property, defaultPretestOptimizationOverride, determining the behavior of **FailingConnectionOnly**.

  Purge policies in the Liberty profile, and their full profile equivalents, are as follows:
  - `purgePolicy="EntirePool"`, which is the same for both.
  - `purgePolicy="FailingConnectionOnly"`, which is equivalent to `purgePolicy="FailingConnectionOnly"` with `defaultPretestOptimizationOverride="false"` in the full profile.
  - `purgePolicy="ValidateAllConnections"`, which is equivalent to `purgePolicy="FailingConnectionOnly"` with `defaultPretestOptimizationOverride="true"` in the full profile.

Migrating a DB2 data source to the Liberty profile

You can easily migrate a DB2® data source to the Liberty profile.

About this task

Refer to the following code examples to see the configurations for a DB2 data source in the full profile and Liberty profile.

Example

In the full profile:

```
<resources.jdbc:JDBCProvider xmi:id="JDBCProvider_1321914412932"
            providerType="DB2 Using IBM JCC Driver" isolatedClassLoader="false"
            implementationClassName="com.ibm.db2.jcc.DB2ConnectionPoolDataSource" xa="false">
   <classpath>${DB2_JCC_DRIVER_PATH}/db2jcc4.jar</classpath>
   <classpath>${DB2_JCC_DRIVER_PATH}/db2jcc_license_cu.jar</classpath>
   <classpath>${DB2_JCC_DRIVER_PATH}/db2jcc_license_cisuz.jar</classpath>
   <factories xmi:type="resources.jdbc:DataSource" xmi:id="DataSource_1321914498985"
            name="DefaultDB2Datasource" jndiName="jdbc/DefaultDB2Datasource"
            providerType="DB2 Using IBM JCC Driver" authMechanismPreference="BASIC_PASSWORD"
            authDataAlias="IBM-9NE5C7ONIG4Node01/dbuser2" relationalResourceAdapter="builtin_rra"
            statementCacheSize="10"
            datasourceHelperClassname="com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper">
      <propertySet xmi:id="J2EEResourcePropertySet_1321914499000">
         <resourceProperties xmi:id="J2EEResourceProperty_1321914499000" name="databaseName"
            type="java.lang.String" value="TESTDB" required="true" ignore="false"
            confidential="false" supportsDynamicUpdates="false"/>
         <resourceProperties xmi:id="J2EEResourceProperty_1321914499001" name="driverType"
            type="java.lang.Integer" value="4" required="true" ignore="false"
            confidential="false" supportsDynamicUpdates="false"/>
```

```
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499002" name="serverName"
           type="java.lang.String" value="localhost" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499003" name="portNumber"
           type="java.lang.Integer" value="50000" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499010" name="currentLockTimeout"
           type="java.lang.Integer" value="10" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499013" name="currentSchema"
           type="java.lang.String" value="DBUSER2" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499015" name="cursorSensitivity"
           type="java.lang.Integer" value="0" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499016" name="deferPrepares"
           type="java.lang.Boolean" value="true" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499027" name="loginTimeout"
           type="java.lang.Integer" value="0" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499032" name="resultSetHoldability"
           type="java.lang.Integer" value="1" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499034"
           name="retrieveMessagesFromServerOnGetMessage" type="java.lang.Boolean" value="true"
           required="false" ignore="false" confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499041" name="traceLevel"
           type="java.lang.Integer" value="-1" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499052"
           name="beginTranForResultSetScrollingAPIs" type="java.lang.Boolean" value="false"
           required="false" ignore="false" confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499053" name="beginTranForVendorAPIs"
           type="java.lang.Boolean" value="false" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499054" name="connectionSharing"
           type="java.lang.Integer" value="-1" required="false" ignore="false"
           confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499060"
           name="nonTransactionalDataSource" type="java.lang.Boolean" value="false" required="false"
           ignore="false" confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499063"
           name="syncQueryTimeoutWithTransactionTimeout" type="java.lang.Boolean" value="false"
           required="false" ignore="false" confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499069"
           name="webSphereDefaultIsolationLevel" type="java.lang.Integer" value="2" required="false"
           ignore="false" confidential="false" supportsDynamicUpdates="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1321914499070"
           name="webSphereDefaultQueryTimeout" type="java.lang.Integer" value="10" required="false"
           ignore="false" confidential="false" supportsDynamicUpdates="false"/>
      </propertySet>
      <connectionPool xmi:id="ConnectionPool_1321914499012" connectionTimeout="180"
            maxConnections="10" minConnections="1" reapTime="180" unusedTimeout="1800"
            agedTimeout="7200" purgePolicy="EntirePool" />
      <mapping xmi:id="MappingModule_1321914681786" mappingConfigAlias=""
            authDataAlias="IBM-9NE5C7ONIG4Node01/dbuser2"/>
   </factories>
</resources.jdbc:JDBCProvider>
<systemLoginConfig xmi:id="JAASConfiguration_2">
   <authDataEntries xmi:id="auth1" alias="IBM-9NE5C7ONIG4Node01/dbuser2"
            userId="dbuser2" password="{xor}LDcfLTo7Oz0=" />
</systemLoginConfig>
```

In the Liberty profile, the equivalent configuration is:

```
<variable name="DB2_JCC_DRIVER_PATH" value="C:/Drivers/DB2" />
<library id="db2Lib">
   <fileset dir="${DB2_JCC_DRIVER_PATH}" includes="db2jcc4.jar
            db2jcc_license_cu.jar db2jcc_license_cisuz.jar" />
</library>
<dataSource id="DefaultDB2Datasource" jndiName="jdbc/DefaultDB2Datasource"
    statementCacheSize="10"
    beginTranForResultSetScrollingAPIs="false"
    beginTranForVendorAPIs="false"
    connectionSharing="MatchCurrentState"
    transactional="false"
    syncQueryTimeoutWithTransactionTimeout="false"
    isolationLevel="TRANSACTION_READ_COMMITTED"
    queryTimeout="10"
   >
   <jdbcDriver libraryRef="db2Lib"
            javax.sql.ConnectionPoolDataSource="com.ibm.db2.jcc.DB2ConnectionPoolDataSource"/>
   <properties.db2.jcc
    databaseName="TESTDB"
    driverType="4"
    serverName="localhost"
    portNumber="50000"
    currentLockTimeout="10"
    currentSchema="DBUSER2"
    cursorSensitivity="0"
    deferPrepares="true"
    loginTimeout="0"
    resultSetHoldability="1"
    retrieveMessagesFromServerOnGetMessage="true"
    traceLevel="-1"
    user="dbuser2"
    password="{xor}LDcfLTo7Oz0="
   />
   <connectionManager connectionTimeout="180" maxPoolSize="10" minPoolSize="1" reapTime="180"
            maxIdleTime="1800" agedTimeout="7200" purgePolicy="EntirePool"/>
</dataSource>
```

Migrating a Derby embedded data source to the Liberty profile

You can easily migrate a Derby Embedded data source to the Liberty profile.

About this task

Refer to the following code examples to see the configurations for a Derby Embedded data source in the full profile and Liberty profile.

Example

In the full profile:

```
<resources.jdbc:JDBCProvider xmi:id="JDBCProvider_1183122153343"
          providerType="Derby JDBC Provider"
          implementationClassName="org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource"
          xa="false">
   <classpath>${DERBY_JDBC_DRIVER_PATH}/derby.jar</classpath>
   <factories xmi:type="resources.jdbc:DataSource" xmi:id="DataSource_1183122153625"
          name="DefaultDerbyDatasource" jndiName="jdbc/DefaultDatasource"
          providerType="Derby JDBC Provider" authMechanismPreference="BASIC_PASSWORD"
          relationalResourceAdapter="builtin_rra" statementCacheSize="10"
          datasourceHelperClassname="com.ibm.websphere.rsadapter.DerbyDataStoreHelper">
      <propertySet xmi:id="J2EEResourcePropertySet_1183122153625">
```

```
        <resourceProperties xmi:id="J2EEResourceProperty_1183122153625" name="databaseName"
         type="java.lang.String" value="C:/myDerby/DefaultDB" required="true"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1183122153626" name="shutdownDatabase"
         type="java.lang.String" value="false" required="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1183122153629" name="connectionAttributes"
         type="java.lang.String" value="upgrade=true" required="false"/>
        <resourceProperties xmi:id="J2EEResourceProperty_1183122153630" name="createDatabase"
         type="java.lang.String" value="create" required="false"/>
      </propertySet>
      <connectionPool xmi:id="ConnectionPool_1183122153625" connectionTimeout="180"
          maxConnections="10" minConnections="1" reapTime="180" unusedTimeout="1800"
          agedTimeout="7200" purgePolicy="EntirePool"/>
   </factories>
</resources.jdbc:JDBCProvider>
```

In the Liberty profile, the equivalent configuration is:

```
<variable name="DERBY_JDBC_DRIVER_PATH" value="C:/Drivers/derby" />
<library id="derbyLib">
   <fileset dir="${DERBY_JDBC_DRIVER_PATH}" includes="derby.jar" />
</library>
<dataSource id="DefaultDerbyDatasource" jndiName="jdbc/DefaultDerbyDatasource"
       statementCacheSize="10">
   <jdbcDriver libraryRef="derbyLib"
       javax.sql.ConnectionPoolDataSource="org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource"/>
   <properties.derby.embedded
       databaseName="C:/myDerby/DefaultDB"
       shutdownDatabase="false"
       connectionAttributes="upgrade=true"
       createDatabase="create"
   />
   <connectionManager connectionTimeout="180" maxPoolSize="10" minPoolSize="1" reapTime="180"
       maxIdleTime="1800" agedTimeout="7200" purgePolicy="EntirePool" />
</dataSource>
```

Chapter 4. Migrating product configurations

Use the WebSphere Application Server Version 8.5 migration tools to migrate your product configurations. These migration tools support migration from Version 6.1 or above.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6. For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

The following configuration upgrades of WebSphere Application Server versions and offerings are directly supported. IBM i

Table 1. Directly Supported Configuration Upgrades. The table lists supported configuration upgrades of WebSphere Application Server versions and offerings.

Migration Source (Version 6.1 or above)	WebSphere Application Server, Network Deployment Version 8.5 Target	
	Stand-alone and Custom Profiles	Deployment-Manager Management Profile
WebSphere Application Server (base) stand-alone application server	Supported	Not supported
WebSphere Application Server, Network Deployment stand-alone application server	Supported	Not supported
WebSphere Application Server, Network Deployment federated application server	Supported	Not supported
WebSphere Application Server, Network Deployment deployment manager	Not supported	Supported
WebSphere Application Server, Express stand-alone application server	Supported	Not supported

IBM i You can migrate your product configurations using the WebSphere Application Server Version 8.5 command-line migration tools.

Before using the migration tools, consult the IBM WebSphere Application Server supported hardware, software, and APIs website to understand what fixes you must apply to earlier versions. Applying fixes to an earlier version might also apply fixes to files that have a role in the migration. Apply any fixes to ensure the most effective migration of configurations and applications.

Note: The deployment manager maintains the master configuration data for all of the nodes that it manages. This configuration data is updated through the configuration manager. When the configuration manager detects that the updates to the configuration data were not made against the latest saved copy, it reject the updates and creates an exception. To avoid this situation, following these best practices:

- Migrate each node independently. For example, let the first node complete the migration process before starting the process for the second node, and so on.
- Ensure that the administrative console for the deployment manager is not running when the migration process is in progress.

If you need to migrate federated nodes concurrently, use the following steps to minimize the potential failures:

Procedure

1. Run the **backupConfig** command for the deployment manager and each federated node before you begin. For more information, see the documentation about the **backupConfig** command.
2. Stop the nodeagent process before you migrate the nodes.
3. Stagger the start of the migration process for each node by 3 to 5 minutes.
4. Run the **restoreConfig** command on that node and rerun the migration process if a failure occurs. For more information, see the documentation about the **restoreConfig** command.

Configuration mapping during product-configuration migration

Various configurations are mapped during product-configuration migration.

IBM i Migration always involves migrating a single profile to another single profile on the same iSeries® server. The migration tools map objects and attributes existing in the version or WebSphere Application Server from which you are migrating to the corresponding objects and attributes in the Version 8.5 environment.

Many migration scenarios are possible. The migration tools map objects and attributes existing in the version from which you are migrating to the corresponding objects and attributes in the Version 8.5 environment.

Bootstrap port

The migration tools carry the old release value into the Version 8.5 environment.

IBM i If a value for the -portBlock parameter is specified during the call to **WASPostUpgrade**, however, a new port value is given to each application server that is migrated to Version 8.5.

Command-line parameters

The migration tools convert appropriate command-line parameters to Java Virtual Machine (JVM) settings in the server process definition. Most settings are mapped directly. Some settings are not migrated because their roles in theWebSphere Application Server Version 8.5 configuration do not exist, have different meanings, or have different scopes.

For information on how to change the process-definition settings, read the "Process definition settings" article in the information center. For information on how to change the JVM settings, read the "Java virtual machine settings" article in the information center.

Generic server

In Version 6.1 and later, a generic server has its own type, called GENERIC_SERVER. Migration will perform this conversion, but migration cannot accurately migrate the external resources that the generic server references. After migration has completed migrating the generic server settings, you might need to perform additional tasks. If the old resource that the generic server was managing is located under the old WebSphere Application Server installation, perform the following tasks:

1. Copy any related files to the new installation.
2. Run any setup required to put the external application back into a valid and working state.

   It is best that you reinstall the resource into the new WebSphere Application Server directory. Whatever you choose to do, the final step is to reset the reference to the new location of the application.

If the old resource that the generic server was managing is not installed under the oldWebSphere Application Server installation, nothing further is required.

Migration of a Version 6.1 or above node to a Version 8.5 node

You can migrate a WebSphere Application Server Version 6.1 or above node that belongs to a cell without removing the node from the cell.

Migrate the deployment manager first, before migrating any base nodes in the cell.

Important: Use the same cell name when migrating WebSphere Application Server, Network Deployment from Version 6.1 or above to Version 8.5. If you use a different cell name, federated nodes cannot successfully migrate to the WebSphere Application Server, Network Deployment Version 8.5 cell.

Migrating a base WebSphere Application Server node that is within a cell to Version 8.5 also migrates the node agent to Version 8.5. A cell can have some Version 8.5 nodes and other nodes that are at Version 6.1 or above levels. Read “Coexistence support” on page 95 for information on restrictions on using mixed-release cells.

Policy files

WebSphere Application Server Version 8.5 migrates all the policy files that are installed with Version 6.1 or above by merging settings into the Version 8.5 policy files with the following characteristics:

- Any comments located in the Version 8.5 policy files will be preserved. Any comments contained in the Version 6.1 or above policy files will not be included in the Version 8.5 file.
- Migration will not attempt to merge permissions or grants; it is strictly an add-type migration. If the permission or grant is not located in the Version 8.5 file, the migration will bring it over.
- Security is a critical component; thus, the migration makes any additions at the end of the original `.policy` files right after the comment `MIGR0372I: Migrated grant permissions follow`. This is done to help administrators verify any policy-file changes that the migration has made.

IBM i Properties and classes directories

Migration copies files from prior version directories into the WebSphere Application Server Version 8.5 configuration.

Property files

WebSphere Application Server Version 8.5 migrates all the property files that are installed with Version 6.1 or above by merging settings into the Version 8.5 property files.

Resource adapter archives (RARs) referenced by J2C resources

RARs that are referenced by J2C resources are migrated if those RARs are in the old WebSphere Application Server installation. In this case, the RARs are copied over to the corresponding location in the new WebSphere Application Server installation. Relational Resource Adapter RARs will not be migrated.

Migrating cluster-level resources:

WebSphere Application Server Version 6.0 introduced the concept of cluster-level resources. These are configured in `resource`*`xxx`*`.xml` files under the cluster directories. For example:

```
<resources.j2c:J2CResourceAdapter xmi:id="J2CResourceAdapter_1112808424172"
  name="ims" archivePath="${WAS_INSTALL_ROOT}\installedConnectors\x2.rar">
  ...
</resources.j2c:J2CResourceAdapter>
```

If you have a cluster-level resource, this resource must be in the same location on each cluster member (node). Using the above example, therefore, each cluster member must have the RAR file installed at location `${WAS_INSTALL_ROOT}\`

`installedConnectors\x2.rar`. `${WAS_INSTALL_ROOT}` is resolved on each cluster member to get the exact location.

In the migration of a deployment manager, the tools migrate the cluster files on the deployment manager, including the `resource`*`xxx`*`.xml` files.

In the migration of a federated node, the tools process each J2C adapter.

RAR files in a Version 6.1 to Version 8.5 migration:

Migration from Version 6.1 to Version 8.5 copies files such as RAR files from WAS_INSTALL_ROOT to WAS_INSTALL_ROOT and from USER_INSTALL_ROOT to USER_INSTALL_ROOT.

If you have a RAR file in the WAS_INSTALL_ROOT for Version 6.1, for example, the migration tools do not automatically copy the file from WAS_INSTALL_ROOT to USER_INSTALL_ROOT. This maintains the integrity of the cluster-level J2C resources. If you hardcoded a path to a RAR file (`archivePath="C:/WAS/installedConnectors/x2.rar"` for example) in Version 6.1, however, the Version 8.5 migration tools cannot change the archivePath attribute to reflect this because that would break all of the other cluster members that have not been migrated.

Samples

No migration of samples from previous versions is available. There are equivalent WebSphere Application ServerVersion 8.5 samples that you can install.

Security

Java 2 security is enabled by default when you enable security in WebSphere Application Server Version 8.5. Java 2 security requires you to grant security permissions explicitly.

IBM i There are several techniques that you can use to define different levels of Java 2 security in Version 8.5. One is to create a `was.policy` file as part of the application to enable all security permissions. The migration tools call the **`wsadmin`** command to add an existing `was.policy` file in the Version 8.5 `properties` directory to enterprise applications as they are being migrated.

When migrating to WebSphere Application ServerVersion 8.5, your choice of whether or not to migrate to support script compatibility results in one of two different outcomes.

- If you choose to migrate to support script compatibility, your security configuration is brought over to Version 8.5 without any changes.

  This is the default.
- If you choose not to migrate to support script compatibility, the security configuration is converted to the default configuration for WebSphere Application Server Version 8.5. The default security configuration for Version 6.1 and later acts almost the same as in the previous versions, but there are some changes.

  For example, existing keyfiles and trustfiles are moved out of the SSLConfig repertoire and new keystore and truststore objects are created.

For more information on migrating your security configurations to Version 8.5, read the "Migrating, coexisting, and interoperating – Security considerations" article in the information center.

Stdin, stdout, stderr, passivation, and working directories

IBM i The location for these directories is typically the `logs` directory under the WebSphere Application Server profile directory. For WebSphere Application Server Version 8.5, the default location for the `stdin`, `stdout`, and `stderr` files is the `logs` directory located under the WebSphere Application Server profile directory—for example, the logs directory for the default profile is `/QIBM/UserData/WebSphere/AppServer/V80/Base/profiles/default/logs`.

The migration tools attempt to migrate existing passivation and working directories. Otherwise, appropriate Version 8.5 defaults are used.

Note: In a coexistence scenario, using common directories between versions can create problems.

IBM i Transport ports

The migration tools migrate all ports. You must resolve any port conflicts before you can run servers at the same time.

Note: If ports are already defined in a configuration being migrated, the migration tools fix the port conflicts in the Version 8.5 configuration and log the changes for your verification.

If you specify the -portBlock parameter in the **WASPostUpgrade** command, a new value is assigned to each transport that is migrated.

If you specify true for the -replacePorts parameter in the **WASPostUpgrade** command, all port values from the old configuration are used in the new configuration. If you specify false for the -replacePorts parameter, the default port definitions in the new profile are not replaced with the values from the old configuration during migration.

For more information on the **WASPostUpgrade** command, read “WASPostUpgrade command” on page 37.

For further information on transport chains and channels, read the 'Transport chains' article in the information center.

You must manually add virtual host alias entries for each port. For more information, read the "Configuring virtual hosts" article in the information center.

Web modules

The specification level of the Java Platform, Enterprise Edition (Java EE) implemented in WebSphere Application Server Version 6.1 required behavior changes in the web container for setting the content type. If a default servlet writer does not set the content type, not only does the web container no longer default to it but the web container returns the call as "null." This situation might cause some browsers to display resulting web container tags incorrectly. To prevent this problem from occurring, migration sets the autoResponseEncoding IBM extension to "true" for web modules as it migrates enterprise applications.

JVM system properties

If you migrate a Version 6.1 configuration that has feature packs installed, the migration tools might add one or two JVM system properties for each Java server in your configuration, including your administrative servers. Web servers are not affected. The properties are set to indicate to the JVM that the configuration should use a Java annotation scan policy other than the Version 8.5 default scan policy.

- If you migrate a Version 6.1 profile that has the Feature Pack for EJB 3.0 installed, the migration tools add the following system property to the JVM definitions for all Java servers defined on that node:

  ```
  com.ibm.websphere.ejb.UseEJB61FEPScanPolicy = true
  ```

- If you migrate a Version 6.1 profile that has the Feature Pack for Web Services installed, the migration tools add the following system property to the JVM definitions for all Java servers defined on that node:

```
com.ibm.websphere.webservices.UseWSFEP61ScanPolicy = true
```

- If you migrate a Version 6.1 profile that has both the Feature Pack for EJB 3.0 and the Feature Pack for Web Services installed, the migration tools add both of the system properties to the JVM definitions for all Java servers defined on that node:

  ```
  com.ibm.websphere.ejb.UseEJB61FEPScanPolicy = true
  com.ibm.websphere.webservices.UseWSFEP61ScanPolicy = true
  ```

  A WebSphere Application Server, Network Deployment configuration requires that the deployment manager profile be augmented with all of the feature packs used in the cell. This means that the deployment-manager profile can potentially have both feature packs installed even if none of its federated nodes have both installed.

If these properties are set, the following two changes take place in the default Version 8.5 behavior:

- Application installation generates classes based on the annotation scan policy associated with the settings for those two properties.

  This means that you can potentially use the following four annotation scan policies:

  – Version 8 default behavior
  – Feature Pack for EJB 3.0 behavior
  – Feature Pack for Web Services behavior
  – Net behavior from having both the Feature Pack for EJB 3.0 and the Feature Pack for Web Services installed

- The servers use the generated annotation classes based on the properties set, resulting in four potential behaviors.

You can change the scan policy behavior by adding or removing the custom JVM system properties from your `server.xml` files.

Important: To evoke the correct installApp behavior, the `server.xml` file for the deployment manager must retain any property specified for any node in the cell.

After changing the properties, you must reinstall or update your applications and then resynchronize the cell to implement the change.

Preparing for product-configuration migration

You must prepare your system for migrating to WebSphere Application Server Version 8.5.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6.

For help, read Chapter 14, “Troubleshooting migration,” on page 109.

Procedure

1. Install WebSphere Application Server Version 8.5.
2. Familiarize yourself with the tools and features of WebSphere Application Server Version 8.5.
3. Verify that you have the minimum prerequisites required for migration.
   For more information, read “Checking for the product-configuration migration prerequisites” on page 27.
4. Evaluate the items deprecated in WebSphere Application Server Version 8.
   For more information, read the "Deprecated and removed features" article in the information center.
5. Evaluate the changes to API specification levels to determine which applications you need to migrate.
   To plan your application migration requirements, use “Migrating API and specifications” on page 10.

6. Evaluate the changes to configuration settings.

   For more information, read “Configuration mapping during product-configuration migration” on page 22.

Checking for the product-configuration migration prerequisites

Before you migrate your old version of WebSphere Application Server to Version 8.5, you must make sure that you meet certain requirements.

Before you begin

Procedure

- Make sure that you have the minimum version of source WebSphere Application Server that is required. If you are migrating from WebSphere Application Server 6.x, you must be at Version 6.0 or higher.

  To determine the current level of WebSphere Application Server installed on your system, perform these steps:

  Version 6.x:

  1. Enter the following command on the command line:

     ```
     WRKLNK '/QIBM/ProdData/WebSphere/AppServer/V6/product_dir/properties/version/WAS.product'
     ```

     where *product_dir* is:

     – `Base` for WebSphere Application Server (base) and WebSphere Application Server, Express Version 6.x

     – `ND` for WebSphere Application Server, Network Deployment Version 6.x
  2. Specify option 5 (Display) next to the product file to view the contents. The number within the <version> tags shows the current version that you have installed.

  If you do not meet the minimum version, obtain the latest group PTF. Read WebSphere Application Server PTFs for iSeries for information on the correct group PTF for your operating-system release level and WebSphere Application Server Version 6.x product.
- Make sure that WebSphere Application Server Version 8.5 is installed.

  WebSphere Application Server, Network Deployment needs to be installed if you are migrating to WebSphere Application Server, Network Deployment.
- Make sure that your user profile has *ALLOBJ authority.

  When you call the **WASPreUpgrade** and **WASPostUpgrade** migration tools, your user profile must have *ALLOBJ authority.

Migrating profiles using the migration wizard

Use the migration tools to migrate WebSphere Application Server Version 6.x, 7.x, or 8.x profiles to Version 8.5.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6.

For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

Read WASPreUpgrade command and WASPostUpgrade command for descriptions of the parameters related to the information that you need to collect before you begin this procedure. The Migration wizard prompts you for this information during the migration.

Before using the Migration wizard, install WebSphere Application Server Version 8.5.

Use the Profile Management tool or the manageprofiles command to create a valid new target Version 8.5 profile if one does not already exist, or you can create a target profile using the Migration wizard.

The WebSphere Application Server - Express product creates a valid Version 8.5 profile for a standalone application server during installation.

Tip: Before migrating a WebSphere Application Server Version 6.x, 7.x, or 8.0 application server profile, use the **backupConfig** command or your own preferred backup utility to back up your existing configuration. Note the name and location of your backup configuration. You can also choose to let the Configuration Migration Tool run the **backupConfig** command. You can run the **backupConfig** command on the source profile and the intended target profile if the Configuration Migration Tool is not creating the profile. For more information, see **backupConfig** command.

About this task

The wizard is the graphical user interface to the primary Version 8.5 command-line migration tools, which are the WASPreUpgrade command and the WASPostUpgrade command. After gathering all of the information that is required for the migration, use the wizard to migrate a WebSphere Application Server Version 6.x, 7.x, or Version 8.0 profile to a Version 8.5 profile.

For help, read Chapter 14, “Troubleshooting migration,” on page 109.

Procedure

1. Start the migration wizard. Perform one of the following options to access and start the Migration wizard:
   - Run one of the following commands:
2. Define a migration source. On the Migration Sources panel, click **New**.
3. Select the installations that you want to migrate. View the list of valid installations which you can migrate to Version 8.5. You can select installations from this list, or you can specify an installation which has not been detected. Click **Next**.
4. Create a backup of the valid source profiles managed by the installation you selected. This step is optional. If you wish to make a backup of the profile, select the profile check box and specify the location for the profile backup.
   a. Select a valid source profile from the drop down list.
   b. Optionally choose to run backupConfig on the selected source profile.
   c. Click **Next**.

   Note: You can use the restoreConfig script to apply this backup and restore the profile.
5. Specify the location for the migration output. The migration output includes log files. You can also optionally enable tracing to the appropriate level and select the location for the trace file. The trace file destination can be a different location from the Migration output location. Click **Next**.
6. Create the target profile or use an existing profile from the target installation. If you are using an existing profile, specify the name of the profile and click **Next**.

   Note: Only the names of those profiles which match the type of the source profile being migrated will be displayed in the list. If you are using an existing profile, optionally choose to run backupConfig on the target profile by selecting the checkbox and specifying the location for the profile backup.
7. Provide the required information for the profile creation process. If you are creating a new profile, the information required depends on the profile type that you are migrating. Click **Next**.
   a. If migrating a federated node, verify the Deployment Manager is running and open for connections.

8. Select a method to use for migrating the applications. If you are migrating applications, specify the application installation directory. Click **Next**.
9. Specify whether to use the port values from the source or target profile. Click **Next**.
10. Select additional migration options. Select the appropriate settings. To proceed to the summary page, click **Next**.
11. Review and verify the migration data.

    Note: You can generate a text file of the commands. You can use this text file to create a script that uses the commands.
12. Start the migration. Click **Next** to start the migration process. The migration process consists of 2-6 sub-processes, depending on the choices you made in previous steps. Each sub-process has a tab where its progress can be monitored. Controls are in place to allow the process to flow continuously upon each success, or the user can control when to stop or continue the migration process as a whole. The **Stop** button brings up a confirmation dialog box. You can review any of the sub-process tabs at any time during the migration. If the process requires user input, you are automatically moved to the appropriate tab.
13. Review the results page. After all sub-processed have completed, the results page is displayed. To return to the list of Migration sources, click **Finish**.

What to do next

You can now start the migrated standalone application server in the WebSphere Application Server Version 8.5 environment or begin a new migration scenario.

Migrating product configurations with migration tools

Migration support consists of tools that are included with WebSphere Application Server. These tools primarily provide support for saving the configuration and applications from a previous version of the product into a migration-specific backup directory and then importing that configuration into the latest version of the application server.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6. For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

Important: Use the migration tools for the version of WebSphere Application Server that you are installing. The tools change over time. If you use migration tools from an earlier release of WebSphere Application Server, you are likely to encounter a problem with the migration.

The migration scripts are located in the *app_server_root*/bin directory after installation.

Procedure

Select the appropriate migration tools to migrate your product configurations.

IBM i **WASPreUpgrade command**

You use the **WASPreUpgrade** command to save the applications and configuration data from a previous installation of WebSphere Application Server to a backup directory.

The **WASPostUpgrade** command restores the configuration data from the directory to the new installation.

Read “WASPreUpgrade command” on page 34 for more information.

IBM i **WASPostUpgrade command**

You use the **WASPostUpgrade** command to restore the configuration data from a previous release.

The **WASPostUpgrade** command reads the data from the backup directory where the **WASPreUpgrade** command stored the data.

Read “WASPostUpgrade command” on page 37 for more information.

clientUpgrade tool

You can use the clientUpgrade tool to upgrade the client application to a new release level.

IBM i Read “clientUpgrade script” for more information.

convertScriptCompatibility tool

Administrators use the **convertScriptCompatibility** tool to convert their configuration from a mode that supports backward compatibility of Version 6.1 or above administration scripts to a mode that is fully Version 8.5.

Read “convertScriptCompatibility command” on page 31 for more information.

convertSelfSignedCertificatesToChained task

Chained certificates are the default certificate type in WebSphere Application Server Version 8.5. Administrators can use the **convertSelfSignedCertificatesToChained** task with the **wsadmin** tool to convert self-signed certificates to chained certificates.

Read the "SSLMigrationCommands command group for the AdminTask object" article for more information.

Tip: For help, read Chapter 14, “Troubleshooting migration,” on page 109.

What to do next

Use the selected tools to migrate your product configuration.

clientUpgrade script

The clientUpgrade script migrates application client modules and their resources in an enterprise archive (EAR) file so that these application clients can run in WebSphere Application Server Version 8.5. The script converts an EAR file that you want to migrate and then overwrites the original EAR file with the converted EAR file.

The following title provides information about the clientUpgrade script.

Attention: This command was deprecated in Version 6.1.

Type	Description
Product	The clientUpgrade script is available in the WebSphere Application Server (WebSphere Application Server, Express and WebSphere Application Server (base)) product only.
Authority	To run this script, your user profile must have *ALLOBJ authority.

Syntax	The syntax of the clientUpgrade script is:
	`clientUpgrade` *`EAR_file`* `[ -clientJAR` *`client_JAR_file`* `]`
	`[ -logFileLocation` *`logFileLocation`* `]`
	`[ -traceString` *`trace_spec`* `[ -traceFile` *`file_name`* `] ]`

Type	Description
Parameters	The parameters of the clientUpgrade script are: • *EAR_file* -- This is a required parameter. The value EAR_file specifies the fully-qualified path of the EAR file that contains the application client modules that you want to migrate. • -clientJAR -- This is an optional parameter. The value client_JAR_file specifies a JAR file that you want to migrate. The script overwrites the original EAR file with a new EAR file that contains only the specified JAR files. If you do not specify this parameter, the clientUpgrade script migrates all client JAR files in the EAR file. • -logFileLocation -- Use this optional parameter to specify an alternate location to store the log output. • -traceString -- This is an optional parameter. The value trace_spec specifies the trace information that you want to collect. To gather all trace information, specify "*=all=enabled" (including the double quotation marks (")). By default, the script does not gather trace information. If you specify this parameter, you must also specify the -traceFile parameter. • -traceFile -- This is an optional parameter. The value file_name The value file_name specifies the name of the output file for trace information. If you specify the -traceString parameter but do not specify the -traceFile parameter, the script does not generate a trace file.
Logging	The clientUpgrade script displays status while it runs. It also saves more extensive logging information to the `clientupgrade.log` file. This file is located in the `/QIBM/UserData/WebSphere/AppServer/V85/edition/profiles/default/logs` directory (for a default installation using the default profile) or in the location specified by the -logFileLocation parameter.

These examples demonstrate correct syntax. In this example, the `My51Application.ear` file is migrated from WebSphere Application Server Version 5.1, The script overwrites the original EAR file with a new file that you can deploy in your WebSphere Application Server Version 8.5 profile.

```
clientUpgrade /My51Application/My51Application.ear
```

In this example, only the `myJarFile.jar` client JAR file is migrated. The script overwrites `My51Application.ear` with an EAR file that contains `myJarFile.jar`. You can deploy the new EAR file in your WebSphere Application Server profile.

```
clientUpgrade /My51Application/My51Application.ear -clientJAR myJarFile.jar
```

convertScriptCompatibility command

The **convertScriptCompatibility** command is used by administrators to convert their configurations from a mode that supports backward compatibility of WebSphere Application Server Version 6.x or Version 7.0.x administration scripts to a mode that is fully in the Version 8.0 configuration model.

The scope of the configuration changes depend on the type of profile that is being processed.

- For stand-alone configurations, the default is to convert all servers owned by the node in that configuration.

  Use the -serverName parameter for more granular control.
- For WebSphere Application Server, Network Deployment configurations, the default behavior is to convert all nodes and all servers owned by those nodes.

  Use the -nodeName and -serverName parameters for more granular control.

Nodes are checked to verify that they are at a WebSphere Application Server Version 8.0 level before they are processed in order to support mixed-node configurations. Client environments are not processed.

The following conversions take place with this tool:

- IBM i `processDef` to `processDefs`

  WCCM objects of type `processDef` from WebSphere Application Server Version 6.x are converted to use `processDefs` as defined in the Version 8.0 `server.xml` model. The existing `processDef` object remains in the configuration and is ignored by the runtime.

- transports to channels

  Existing transport entries in the configuration from WebSphere Application Server Version 6.x are mapped to channel support. This affects `server.xml` and `serverindex.xml` files. The values of the transport settings are used to create new channel entries.
- SSL configuration

  WebSphere Application Server Version 8.0 contains enhancements to SSL configuration that result in refactoring the existing SSL configuration model. Both the old and the new model are supported. The default is to map to the WebSphere Application Server Version 6.x or Version 7.x SSL configuration model.
- `bootstrapAddress` to `bootstrapAddresses`

  Each single bootstrap address configuration is converted to a new bootstrap address list configuration containing that single bootstrap address.
- `ObjectRequestBroker` from not using the server thread pool to using it

  For example, `<ObjectRequestBroker useServerThreadPool="false"...>` is changed to `<ObjectRequestBroker useServerThreadPool="true">`.

Location

The **convertScriptCompatibility** command is located in the following directory.

- IBM i *app_server_root*/bin

Syntax

The syntax is as follows:

IBM i

```
convertScriptCompatibility [-help]

convertScriptCompatibility [-profileName profile_name]
                     [-backupConfig true | false]
                     [-nodeName node_name  [-serverName server_name]]
                     [-traceString trace_spec  [-traceFile file_name]]
```

Parameters

Supported arguments include the following parameters:

-help
This displays help for this command

-backupConfig
This is an optional parameter that is used to back up the existing configuration of the current profile. The default is true—that is, to use the **backupConfig** command to save a copy of the current configuration into the *profile_name*/temp directory.

Use the **restoreConfig** command to restore that configuration as required.

Read the "restoreConfig command" article in the information center for more information.

IBM i **-profileName**
This is an optional parameter that is used to specify the profile configuration in the Version 8.0 environment. If this is not specified, the default profile is used. If the default profile has not been set or cannot be found, the system returns an error.

-nodeName
This is an optional parameter that is used to specify a particular node name be processed rather than every node in the configuration. If this is not specified, all nodes in the configuration are converted.

-serverName
This is an optional parameter that is used to specify a particular server name to be processed rather than every server in the configuration. It can be used on all profile types and can be used in conjunction with the -nodeName parameter when processing WebSphere Application Server, Network Deployment configurations. If this parameter is not specified, all servers in the configuration are converted. If it is used in conjunction with the -nodeName parameter, all processing is limited to the specified node name.

-traceString
This is an optional parameter. The value *trace_spec* specifies the trace information that you want to collect. To gather all trace information, specify `"*=all=enabled"` (with quotation marks). The default is to not gather trace information. If you specify this parameter, you must also specify the -traceFile parameter.

-traceFile
This is an optional parameter. The value *file_name* specifies the name of the output file for trace information. If you specify the -traceString parameter but do not specify the -traceFile parameter, the command does not generate a trace file.

Usage

Stand-alone application server profile

Example scenario

1. IBM i Run the **WASPostUpgrade** command and specify `-scriptCompatibility=true` or do not specify a value for the -scriptCompatibility parameter (which has a default value of true).
2. Follow these steps to convert all servers under this stand-alone profile:
   a. IBM i Start a Qshell session.
   ```
   STRQSH
   ```
   b. IBM i Change to the Version 8.0 *app_server_root*`/bin` directory.
   c. Run the following command:
   IBM i
   ```
   convertScriptCompatibility -profileName profile_name
   ```

Deployment manager with federated nodes

Example scenario 1

1. IBM i Run the **WASPostUpgrade** command against the deployment manager as well as all of its federated nodes and specify `-scriptCompatibility=true` or do not specify a value for the -scriptCompatibility parameter (which has a default value of true).
2. Follow these steps to convert all nodes and servers in this cell.

   Important: The following steps should be taken against the deployment-manager management profile. If you run the **convertScriptCompatibility** command against a federated profile, the changes will be removed the next time the deployment manager synchronizes with the federated node.

   a. IBM i Start a Qshell session.
   ```
   STRQSH
   ```
   b. IBM i Change to the Version 8.0 *app_server_root*`/bin` directory.
   c. Run the following command:
   IBM i
   ```
   convertScriptCompatibility -profileName dmgr_profile_name
   ```
3. Perform a full resynchronization of the deployment manager's configuration with each federated node to produce a consistent configuration.

Example scenario 2

1. IBM i Run the **WASPostUpgrade** command against the deployment manager and specify `-scriptCompatibility=false`.
2. IBM i Run the **WASPostUpgrade** command against the deployment manager's federated nodes and specify `-scriptCompatibility=true` or do not specify a value for the -scriptCompatibility parameter (which has a default value of true).
3. Follow these steps to convert all non-converted nodes and servers in the cell.

   Important: The following steps should be taken against the deployment-manager management profile. If you run the **convertScriptCompatibility** command against a federated profile, the changes will be removed the next time the deployment manager synchronizes with the federated node.

   a. IBM i Start a Qshell session.

   ```
   STRQSH
   ```

   b. IBM i Change to the Version 8.0 *app_server_root*`/bin` directory.

   c. For each federated node that has not been converted, run the following command:

   IBM i

   ```
   convertScriptCompatibility -profileName dmgr_profile_name –nodeName ${non_converted_nodename}
   ```

4. Perform a full resynchronization of the deployment manager's configuration with each federated node to produce a consistent configuration.

For more information about where to run this command, read the "Using command line tools" article in the information center.

WASPreUpgrade command

The **WASPreUpgrade** command for WebSphere Application Server Version 8.5 saves the configuration of a previously installed version of WebSphere Application Server into a migration-specific backup directory.

Location

The command file is located in and must be run from the Version 8.5 *app_server_root*`/bin` directory.

IBM i

Authority

To run this command script, your user profile must have *ALLOBJ authority.

Syntax

IBM i

```
WASPreUpgrade backupDirectory
              currentWebSphereDirectory
              [-traceString trace_spec [-traceFile file_name ]]
              [-machineChange true | false]
              [-workspaceRoot profile1=user_workspace_folder_name_1;profile2=user_workspace_folder_name_2]
              [-username < user name >]
              [-password < password >]
              [-javaoption < -Xms...m > -javaoption < -Xmx...m > ]
```

Parameters

The command has the following parameters:

backupDirectory
This is a required parameter and must be the first parameter that you specify. The value *backupDirectory* specifies the name of the directory where the command script stores the saved configuration.

Note: The `WAS_INSTALL` and `USER_INSTALL` root directories are invalid directories for the location of the WebSphere Application Server backup directory.

This is also the directory from which the **WASPostUpgrade** command reads the configuration.

If the directory does not exist, the **WASPreUpgrade** command script creates it.

currentWebSphereDirectory
This is a required parameter and must be the second parameter that you specify. This can be any edition of WebSphere Application Server Version 6.1 for which migration is supported.

IBM i The value *currentWebSphereDirectory* specifies the name of the instance or profile root directory for the current WebSphere Application Server instance Version 6.1 profile that you want to migrate.

- In Version 6.1, the profile root may be a unique value chosen during profile creation but the following directories are the defaults:
  - **For Version 6.1 Express or base:** /QIBM/UserData/WebSphere/AppServer/V61/Base/profiles/*profile*
  - **For Version 6.1 WebSphere Application Server, Network Deployment:** /QIBM/UserData/WebSphere/AppServer/V61/ND/profiles/*profile*

-traceString
This is an optional parameter. The value *trace_spec* specifies the trace information that you want to collect.

To gather all trace information, specify `"*=all=enabled"` (with quotation marks).

If you do not specify the -traceString or -traceFile parameter, the command creates a trace file by default and places it in the *backupDirectory*/`logs` directory.

-traceFile
This is an optional parameter. The value *file_name* specifies the name of the output file for trace information.

If you do not specify the -traceString or -traceFile parameter, the command creates a trace file by default and places it in the *backupDirectory*/`logs` directory.

-machineChange
This is an optional parameter used for a migration involving cross operating-system and machine boundaries. If specified as true, this parameter provides support for changing physical hardware when migrating by backing up items that are stored outside the WebSphere Application Server installation or profile folder hierarchy. If specified as false, only files stored under the WebSphere Application Server installation folder or profile folders are copied to the backup directory during migration.

The default is false.

When this value is false, migration assumes that the new and old WebSphere Application Server installations are on the same physical machine with shared access to the file system. Therefore, any files located outside the WebSphere directories are communal and can be shared. Migration does not copy files outside the WebSphere Application Server tree into the backup directory when -machineChange is false. If you select -machineChange=false, you must run the **WASPostUpgrade** command on the same physical hardware.

If you intend to run the **WASPostUpgrade** command on a different machine or file system, you should run the **WASPreUpgrade** command with -machineChange=true. If you select -machineChange=true, migration creates an additional subdirectory (`/migrated/`) in the migration backup directory that

contains any files referenced by the WebSphere Application Server configuration that reside outside the product or profile directories. When you run the **WASPostUpgrade** command, these files are returned to their original paths on the new machine.

Performance considerations:

If you migrate with Service Integration Bus (SIB) busses configured with file-system file-store repositories, you might require additional space in your migration heap and migration backup directory. Each bus has three file-store values—a log, a tempspace, and a repository. These three files vary in size, but they can be as much as 100-500 MB each. When migration is running, it backs up any file stores that are in the WebSphere Application Server tree during the pre-upgrade process. There needs to be sufficient space on the file system to permit this. If file stores exist at the destination location already during the post-upgrade process, migration backs up the file stores in memory to support rollback.

If you run the **WASPreUpgrade** command with -machineChange=true, resulting in a backup directory that contains shared file-store objects, you might find that the post-upgrade process suffers from out-of-memory exceptions because the default maximum heap is too small to contain the file-store backups in support of rollback. To resolve this issue, perform one of the following three tasks:

- If the file stores at the system location are valid, delete the copies from the backup directory before running the **WASPostUpgrade** command.

  By deleting the entire `/migrated/` subdirectory from the migration backup directory before running the **WASPostUpgrade** command, you essentially convert your pre-upgrade backup from -machineChange=true to -machineChange=false.
- If the copies of the file stores in the backup directory are valid, delete the versions at the destination location.

  This changes the rollback support so that the destination files do not exist and will not occupy space in memory during the migration.
- If you require rollback support and you need both the files in the backup directory as well as the files on the file system, increase your maximum heap size for the post-upgrade process to some value great enough to support all of the SIB files that conflict.

-workspaceRoot
This is an optional parameter. The value *user_workspace_folder_name_x* specifies the location of the administrative console customized "My tasks" settings for one or more profiles.

-username
This is an optional parameter. The value *user name* specifies the administrative user name of the current WebSphere Application Server installation.

This is a required parameter if the following conditions are true:

- You are migrating a deployment manager.
- Administrative or global security is enabled in the source installation.
- The WebSphere Application Server installation you are migrating from is Version 8.0 or above.

-password
This is an optional parameter. The value *password* specifies the administrative password of the current WebSphere Application Server installation.

This is a required parameter if the following conditions are true:

- You are migrating a deployment manager.
- Administrative or global security is enabled in the source installation.
- The WebSphere Application Server installation you are migrating from is Version 8.0 or above.

-javaoption

This is an optional parameter. Use this parameter to specify memory sizes for the Java heap used by the **WASPostUpgrade** command.

The value "-Xms...m" is the parameter specified to indicate the starting heap size. Replace the "..." with the size in Megabytes that you intended to use. For example, if the starting heap size is to be 128 MB, specify the parameter as: *-javaoption -Xms128m*

The value "-Xmx...m" is the parameter specified to indicate the maximum heap size. Replace the "..." with the size in Megabytes that you intend to use. For example, if the maximum heap size is to be 1024 MB, specify the parameter as: *-javaoption -Xmx1024m*

Logging

The **WASPreUpgrade** tool displays status to the screen while it runs. The tool also saves a more extensive set of logging information in the `WASPreUpgrade.`*`time_stamp`*`.log` file written to the *backupDirectory* directory, where *backupDirectory* is the value specified for the backupDirectory parameter. You can view the `WASPreUpgrade.`*`time_stamp`*`.log` file with a text editor.

Migrated resources

WASPreUpgrade saves all of your resources, but it does not migrate entities in your `classes` directory.

Migration saves the following files in the *backupDirectory* directory.

- `config`
- `properties`

WASPostUpgrade command

The **WASPostUpgrade** command for WebSphere Application Server retrieves the saved configuration that was created by the **WASPreUpgrade** command from the *backupDirectory* that you specified. The **WASPostUpgrade** script for WebSphere Application Server reads the configuration from this directory to migrate to WebSphere Application Server Version 8.5 and adds all migrated applications into the *app_server_root*`/installedApps` directory for the Version 8.5 installation.

Location

The command file is located in and must be run from the *app_server_root*`/bin` directory.

IBM i

Authority

To run this command script, your user profile must have *ALLOBJ authority.

Syntax

IBM i

```
WASPostUpgrade backupDirectory
                    [-username userID]
                    [-password password]
                    [-profileName profile_name]
                    [-scriptCompatibility true | false]
```

```
                    [-portBlock port_starting_number]
                    [-backupConfig true | false]
                    [-replacePorts true | false]
                    [-includeApps true | false | script]
                    [-keepDmgrEnabled true | false]
[-requestTimeout seconds]
[-javaoption -Xms...m -javaoption -Xmx...m]
                    [[-appInstallDirectory user_specified_directory] | [-keepAppDirectory true | false]]
                    [-traceString trace_spec [-traceFile file_name]]
```

Use these parameters when migrating a registered application server with security enabled for both the target and source administrative agent: IBM i

```
WASPostUpgrade backupDirectory
                    [-oldAdminAgentProfilePath path to old admin agent]
[-oldAdminAgentSoapPort soap port of old admin agent]
[-oldAdminAgentHostname hostname of old admin agent, defaults to localhost ]
[-oldAdminAgentUsername login username for old admin agent, if admin security is enabled ]
[-oldAdminAgentPassword login password for old admin agent, if admin security is enabled ]
[-newAdminAgentProfilePath path to new admin agent ]
[-newAdminAgentSoapPort soap port of new admin agent ]
[-newAdminAgentHostname hostname of new admin agent, defaults to localhost ]
[-newAdminAgentUsername login username for new admin agent, if admin security is enabled ]
[-newAdminAgentPassword login password for new admin agent, if admin security is enabled ]
```

Parameters

The command has the following parameters:

backupDirectory
This is a required parameter. The value *backupDirectory* specifies the name of the directory in which the **WASPreUpgrade** tool stores the saved configuration and files and from which the **WASPostUpgrade** tool reads the configuration and files.

-username
This is an optional parameter. The value *userID* specifies the administrative user name of the current WebSphere Application Server Version 6.x or 7.x installation.

This is a required parameter if the following conditions are true:

- You are migrating a deployment manager or a federated node.
- Administrative or global security is enabled in the source installation.
- The administrative or global security user ID is not defined in the `security.xml` file.

-password
This is an optional parameter. The value *password* specifies the password for the administrative user name of the current WebSphere Application Server Version 6.x or 7.x installation.

This is a required parameter if the following conditions are true:

- You are migrating a deployment manager or a federated node.
- Administrative or global security is enabled in the source installation.
- The administrative or global security password is not defined in the `security.xml` file.

-profileName
This is an optional parameter for migrating to specific profiles in WebSphere Application Server Version 8.5. The value *profile_name* specifies the name of the Version 8.5 profile to which the script migrates your configuration. You must have already created this profile before calling the **WASPostUpgrade** command.

If the -profileName parameter is not specified, the default profile is used. If no default profile is found, the system reports an error.

gotcha: If you do not specify the specific profile name on -profileName, then whatever is the designated "default" profile will be migrated. You might have to migrate each profile in the backup taken on pre-migration, using the WASPostUpgrade post-migration command specifying the -oldProfile and -profileName parameters for each and every profile that the client wants in the new environment. If the old profile contains installed applications (installedApps) in addition to the sample application and system applications, then the migration process automatically migrates those applications.

Note: When migrating a stand-alone application server from Version 8.5, you can choose a stand-alone application server node that has already been registered with an administrative agent as the target of the migration.

-scriptCompatibility

This is an optional parameter used to specify whether or not migration should create the following Version 6.x or 7.x configuration definitions:

- Transport
- ProcessDef
- Version 6.x SSL

instead of the following Version 8.5 configuration definitions:

- Channels
- ProcessDefs
- Version 8.5 SSL

The default value is true.

Specify true for this parameter in order to minimize impacts to existing administration scripts. If you have existing wsadmin scripts or programs that use third-party configuration APIs to create or modify the Version 6.x or 7.x configuration definitions, for example, you might want to specify true for this option during migration.

If you want to use a cell that contains Version 6.x or 7.x nodes, you must specify true for this variable.

Note: This is meant to provide a temporary transition until all of the nodes in the environment are at the Version 8.5 level. When they are all at the Version 8.5 level, you should perform the following actions:

1. Modify your administration scripts to use all of the Version 8.5 settings.
2. Use the **convertScriptCompatability** command to convert your configurations to match all of the Version 8.5 settings.

-backupConfig

This is an optional parameter used to specify whether the existing WebSphere Application Server Version 8.5 configuration is saved before any changes are made by the **WASPostUpgrade** tool. The default is true—that is, to use the **backupConfig** command to save a copy of the current configuration into the *profile_name*/temp directory.

Use the **restoreConfig** command to restore that configuration as required. Read the "restoreConfig command" article in the information center for more information.

-portBlock

This is an optional parameter. The *port_starting_number* value specifies the starting value of a block of consecutive port numbers to assign when creating new ports.

By default, this parameter is not set.

If a value is specified for this parameter, any new ports that are assigned are set based on this value. Every time a new port value is required, the port is created based on this value and the seed value is incremented for the next usage. No duplicate ports are assigned.

-replacePorts

This optional parameter is used to specify how to map port values.

- True

  Use all port values from the old configuration in the new configuration. All ports from the old configuration supersede the settings for the same ports in the new configuration.

  – If the -portBlock parameter is not set, a conflicting port is renumbered incrementally from its original value until a nonconflicting port is identified.

  – If the -portBlock parameter is set, a conflicting port is renumbered incrementally from the port block number until a nonconflicting port is identified.

  This value is the default.

- False

  Do not replace the default port definitions in the new profile with the values from the old configuration during migration. All ports values from the new configuration supersede the settings for the same ports in the old configuration

  – If the -portBlock parameter is not set, a conflicting port is renumbered incrementally from its original value until a nonconflicting port is identified.

  – If the -portBlock parameter is set, a conflicting port is renumbered incrementally from the port block number until a nonconflicting port is identified.

-includeApps

You can include business level applications, assets, and composition units as part of the migration. You can optionally migrate these items using the -IncludeApps parameter on the **WASPostUpgrade** command. This is an optional parameter that can be specified in the following ways:

- True

  Include user enterprise applications, business level applications, assets, and composition units as part of the migration.

  This value is the default.

- False

  Do nothing with user enterprise applications, business level applications, assets, and composition units during **WASPostUpgrade** processing.

- Script

  – Enterprise applications

    Prepare user enterprise applications for installation in the WebSphere Application Server Version 8.5 `installableApps` directory without installing them during **WASPostUpgrade** processing.

    Scripts that can be used to install these applications are generated and saved in the *backupDirectory* directory. You can then run these files at any point and in any combination after the **WASPostUpgrade** command has completed. You can also reorganize and combine these files to enhance the efficiency of application installation.

  – Business level applications, assets, and composition units

    The install_all_BLAs.jy script is generated and placed in the backup directory. This script can migrate all business level applications, assets, and composition units located in the backup directory to your target profile. The WASPostUpgradeBLAHelper.bat/.sh script, located in the `<WAS_PROFILE_ROOT>/bin` directory, is used to migrate the business level applications, assets and composition units in the install_all_BLAs.txt file.

    Note: To migrate business level applications, assets, and composition units, you must first create their dependencies.

WebSphere Application Server system applications migrate regardless of the value set by this parameter.

-keepDmgrEnabled

This is an optional parameter used to specify whether to disable the existing WebSphere Application Server Version 6.x or 7.x deployment manager. The default is false.

If this parameter is specified as true, you can use the existing Version 6.x or 7.x deployment manager while the migration is completed. It is only valid when you are migrating a deployment manager; it is ignored in all other migrations.

Caution: Use this parameter with care.

- The reason that WebSphere Application Server Version 6.x or 7.x deployment manager configurations normally are stopped and disabled is to prevent multiple deployment managers from managing the same nodes. You must stop the Version 6.x or 7.x deployment manager before you start using the Version 8.5 deployment manager. The most likely error conditions that occur if this is not done are port conflicts when the second instance of the deployment manager is started.
- Specifying true for this parameter means that any configuration changes made in the old configuration during migration might not be migrated.

-keepAppDirectory

This is an optional parameter used to specify whether to install all applications to the same directories in which they are currently located. The default is false.

If this parameter is specified as true, each individual application retains its location.

If you specify this parameter, you cannot specify the -appInstallDirectory parameter.

Restrictions: If this parameter is specified as true, the location is shared by the existing WebSphere Application Server Version 6.x or 7.x installation and the Version 8.5 installation. If you keep the migrated applications in the same locations as those of the previous version, the following restrictions apply:

- The WebSphere Application Server Version 8.5 mixed-node support limitations must be followed. This means that the following support cannot be used when evoking the **wsadmin** command:
  - Precompile JSP
  - Use Binary Configuration
  - Deploy EJB
- You risk losing the migrated applications unintentionally if you later delete applications from these locations when administering (uninstalling for example) your Version 6.x or 7.x installation.

-appInstallDirectory

This is an optional parameter that is used to pass the directory name to use when installing all applications during migration. The default of *profile_name*\installedApps is used if this parameter is not specified.

If you specify this parameter, you cannot specify the -keepAppDirectory parameter.

Quotes must be used around the directory name if one or more spaces are in the name.

If you use this parameter, the migration tools investigate the node-level variables for the node being migrated both in the backup directory (variables for the old release) and in the destination profile (variables from the new release). If the path is part of any of the following variables in either of these releases, the tools contract the path information to use the related variable:

- APP_INSTALL_ROOT
- USER_INSTALL_ROOT
- WAS_INSTALL_ROOT

When the contraction takes place, you receive the following warning message that tells you that the tools changed your specified value and what that contracted value is:

```
MIGR0341W: Application install directory has been updated to {0}.
```

For example:

```
MIGR0341W: Application install directory has been updated to ${USER_INSTALL_ROOT}\customAppDirectory.
```

or

```
MIGR0341W: Application install directory has been updated to ${APP_INSTALL_ROOT}\
cellName\customAppDirectory\.
```

-traceString
This is an optional parameter. The value *trace_spec* specifies the trace information that you want to collect.

To gather all trace information, specify `"*=all=enabled"` (with quotation marks).

If you do not specify the -traceString or -traceFile parameter, the command creates a trace file by default and places it in the *backupDirectory*`/logs` directory.

-traceFile
This is an optional parameter. The value *file_name* specifies the name of the output file for trace information.

If you do not specify the -traceString or -traceFile parameter, the command creates a trace file by default and places it in the *backupDirectory*`/logs` directory.

-requestTimeout
This is an optional parameter. The value *seconds* refers to the number of seconds that migration will wait before failing attempted wsadmin connections.

This value is also used as the timeout parameter during application migration.

-oldAdminAgentProfilePath
This is an optional parameter. The value *path to old admin agent* refers to the file system path of the profile directory for the original administrative agent.

This parameter is only required if the application server being migrated is managed by an administrative agent.

-oldAdminAgentSoapPort
This is an optional parameter. The value *soap port of old admin agent* refers to the SOAP port used by the original administrative agent for administrative connections.

This parameter is required only if the application server being migrated is managed by an administrative agent.

-oldAdminAgentHostname
This is an optional parameter. The value *hostname of old admin agent* refers to the hostname location of the original administrative agent. If the parameter is not specified, the value is set by default to "localhost".

This parameter is required only if the application server being migrated is managed by an administrative agent.

-oldAdminAgentUsername
This is an optional parameter. The value *login username for old admin agent* refers to the username for the original administrative agent.

This parameter is required only if the application server being migrated is managed by an administrative agent that has administrative security enabled.

-newAdminAgentProfilePath
This is an optional parameter. The value *path to new admin agent* refers to the file system path of the profile directory for the newly migrated Administrative Agent.

This parameter is required only if the application server being migrated is managed by an administrative agent.

-newAdminAgentSoapPort
This is an optional parameter. The value *soap port of old admin agent* refers to the SOAP port used by the newly migrated Administrative Agent for administrative connections.

This parameter is required only if the application server being migrated is managed by an administrative agent.

-newAdminAgentHostname
This is an optional parameter. The value *hostname of old admin agent* refers to the hostname location of the new Administrative Agent. If the parameter is not specified, the value is set by default to "localhost".

This parameter is required only if the application server being migrated is managed by an administrative agent.

-newAdminAgentUsername
This is an optional parameter. The value *login username for old admin agent* refers to the username for the new Administrative Agent.

This parameter is required only if the application server being migrated is managed by an administrative agent that has administrative security enabled.

-newAdminAgentPassword
This is an optional parameter. The value *login password for old admin agent* refers to the username for the new Administrative Agent.

This parameter is required only if the application server being migrated is managed by an administrative agent that has administrative security enabled.

-javaoption < -Xms...m > -javaoption < -Xmx...m >
This is an optional parameter. Use this parameter to specify memory sizes for the Java heap used by WASPostUpgrade.

The value "-Xms...m" specifies the starting heap size. Replace the "..." with the size in Megabytes that you need. For example, if the starting heap size is to be 128 MB, specify the parameter as: -javaoption -Xms128m

The value "-Xmx...m" specifies the maximum heap size. Replace the "..." with the size in Megabytes that you need. For example, if the maximum heap size is to be 1024 MB, specify the parameter as: -javaoption -Xmx1024m

Security considerations

The target system must have security disabled before migration. If you migrate from a source configuration that has security enabled, the **WASPostUpgrade** command automatically enables security for the Version 8.5 target configuration during the migration.

java.security file

During WASPostUpgrade, a copy of the java.security file is made in the target WAS installation before migrating the file. The presence of this copy, named java.security.premigration, indicates to future migrations that the file has already been migrated. Also, the copy gives you a chance to reference the default Version 8.0 settings and determine if you want to make changes to the migrated java.security file.

The following properties are migrated in the java.security file, all other properties and comments are not migrated:

```
?security.provider.* - prepend the source provider list before the target provider list and renumber the target provider li
?networkaddress.cache.ttl
?networkaddress.cache.negative.ttl
?ocsp.enable
?ocsp.responderURL
?ocsp.responderCertSubjectName
?ocsp.responderCertIssuerName
?ocsp.responderCertSerialNumber
```

The following situations might cause the migration of the java.security file to fail. If you must use any of the following settings, then manually migrate the java.security file.

- IBM i If the source profile is configured to use classic JVM, the java.security file will not be migrated. This is because Version 8.0 only supports the j9 JVM and the contents of the java.security file are not compatible between classic and j9 JVMs.
- If the **WASPreUpgrade** command is run with the machineChange=true option, the source and target operating systems will be checked before migrating the java.security file. If the source operating system iss HP or Sun, and the target operating system is not the same, the file will not be migrated. If the target operating system is HP or Sun, and the source operating system is not the same, the file will not be migrated. This is because the contents of the java.security file are not compatible between the different operating systems.
- If the copy of the java.security file to java.security.premigration cannot be made, the file will not be migrated.
- If the java.security file is not writeable, the file will not be migrated.
- If the java.security file is not found in the backup directory, the file will not be migrated. If you use an old backup directory, or if something there is a problem with the old installation, then this problem might occur.

Migrating profiles

After you have migrated your applications, you need to migrate your instance configurations to profiles.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6.

For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

Read “Checking for the product-configuration migration prerequisites” on page 27 to see how to determine the currently installed product level of WebSphere Application Server.

Procedure

Select the appropriate option to obtain instructions on how to migrate from your old version of WebSphere Application Server to a new or default Version 8.5 profile.

- “Migrating to a Version 8.5 stand-alone application server profile” on page 45

  This article contains instructions for migrating a WebSphere Application Server Version 6.x profile to a Version 8.5 stand-alone application server profile.
- “Migrating to a Version 8.5 WebSphere Application Server, Network Deployment cell” on page 47

  This article contains instructions for migrating a WebSphere Application Server, Network Deployment Version 6.x to a Version 8.5 WebSphere Application Server, Express cell.

- “Migrating a large WebSphere Application Server, Network Deployment configuration with a large number of applications” on page 53

  This article contains suggestions for migrating a large WebSphere Application Server Version 6.x WebSphere Application Server, Network Deployment configuration with a large number of applications.

Tip: For help, read Chapter 14, “Troubleshooting migration,” on page 109.

Migrating to a Version 8.5 stand-alone application server profile

Use the migration tools to migrate from WebSphere Application Server Version 6.x to a new Version 8.5 stand-alone application server profile.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6.

For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

For help, read Chapter 14, “Troubleshooting migration,” on page 109.

Before following these instructions, perform the actions in “Preparing for product-configuration migration” on page 26.

Tip: Before migrating a WebSphere Application Server Version 6.x stand-alone application server profile, use the **backupConfig** command or your own preferred backup utility to back up your existing configuration if you want to be able to restore it to its previous state after migration. Read the "backupConfig command" article in the information center for more information. Make sure that you note the exact name and location of this backed-up configuration.

Procedure

1. Create a WebSphere Application Server Version 8.5 profile to receive the Version 6.x configuration.
   a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b. Run the **dspwasinst** script to obtain the node name and server name for the Version 6.x instance or profile that is to be migrated.

      Use the following parameters:

      ```
      app_server_root/bin/dspwasinst
       -instance 6x_profile_name
      ```

      where

      - *app_server_root* is the location of the Version 6.x installation that contains the instance or profile to be migrated
      - *6x_profile_name* is the name of the Version 6.x instance or profile that is to be migrated

      The name of the Version 6.x node is listed in the **Node** section, and the name of the server is listed in the **Information for server** section.

   c. Run the **manageprofiles** script.

      Use the following parameters:

      ```
      app_server_root/bin/manageprofiles
       -create
       -profileName 80_profile_name
      ```

```
-startingPort starting_port_number
-templatePath app_server_root/profileTemplates/default
-serverName 6x_application_server_name
-nodeName 6x_node_name
```

where

- *app_server_root* is the location where Version 8.5 is installed
- *80_profile_name* is the name of your Version 8.5 profile

  This parameter must be identical to the Version 6.x instance or profile that is to be migrated.
- *starting_port_number* is the first of a block of 13 consecutive ports
- *6x_application_server_name* is the name of the Version 6.x application server obtained in the previous step
- *6x_node_name* is the Version 6.x node name obtained in the previous step

  The source and target node names must be identical when migrating to Version 8.5.

For details on the syntax and parameters of the **manageprofiles** command, read the "manageprofiles command" article in the information center.

2. Save the WebSphere Application Server Version 6.x configuration.
   a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b. Run the **WASPreUpgrade** script.

      Use the following parameters:

      ```
      app_server_root/bin/WASPreUpgrade
       backup_directory_name
       profile_root
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *backup_directory_name* (required parameter) is the fully qualified path to the integrated file system directory where the **WASPreUpgrade** migration tool stores the saved configuration and files

        The directory is created if it does not already exist. It is also the directory where the **WASPreUpgrade** migration tool writes a log file called `WASPreUpgrade.log` that chronicles the steps taken by the **WASPreUpgrade** command.
      - *profile_root* (required parameter) is the path to the Version 6.x instance or profile that is to be migrated

      For a full explanation of the **WASPreUpgrade** command and its parameters, read “WASPreUpgrade command” on page 34.
3. Restore the WebSphere Application Server Version 6.x configuration into a Version 8.5 profile.
   a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b. Run the **WASPostUpgrade** script.

      Use the following parameters:

      ```
      app_server_root/bin/WASPostUpgrade
       backup_directory_name
       -profileName 80_profile_name
       [-portBlock port_starting_number]
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed

- *backup_directory_name* is the required name of the directory in which the **WASPreUpgrade** tool stored the saved configuration and files and from which the **WASPostUpgrade** tool reads the configuration and files
- *80_profile_name* is the name of the Version 8.5 profile to which the script migrates your configuration
- *port_starting_number* specifies the first of a block of 10 to 15 consecutive port numbers that are not in use on the iSeries server where the migration is being performed

  It is recommended that you always specify the -portBlock parameter if you do not want your profile's ports to conflict with the default profile's ports.

  Note: When migrating a stand-alone application server from Version 6.x to Version 8.5, you can choose a stand-alone application server node that has already been registered with an administrative agent as the target of the migration.

For a full explanation of the **WASPostUpgrade** command and its parameters, read "WASPostUpgrade command" on page 37.

4. Start the WebSphere Application Server Version 8.5 profile that receives the Version 6.x configuration.
   a. Start the QWAS85 subsystem if it is not already started.

      Enter the following command from a command line:

      ```
      STRSBS QWAS85/QWAS85
      ```

   b. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   c. Run the **startServer** script.

      Use the following parameters:

      ```
      app_server_root/bin/startServer
       -profileName 80_profile_name
       6x_server_name
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *80_profile_name* is the name of the Version 8.5 profile created in an earlier step
      - *6x_server_name* is the name of the Version 6.x application server that was migrated

Migrating to a Version 8.5 WebSphere Application Server, Network Deployment cell

Use the migration tools to migrate a WebSphere Application Server, Network Deployment Version 6.x cell to a Version 8.5 WebSphere Application Server, Network Deployment cell.

Before you begin

Read "Overview of migration, coexistence, and interoperability" on page 1 and "Premigration considerations" on page 6.

For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

Read "Checking for the product-configuration migration prerequisites" on page 27 to determine the currently installed product level of WebSphere Application Server.

For help, read Chapter 14, "Troubleshooting migration," on page 109.

Before following these instructions, perform the actions in “Preparing for product-configuration migration” on page 26.

Migration of a WebSphere Application Server, Network Deployment Version 6.x deployment manager and associated federated nodes can be performed in stages.

1. Migrate the WebSphere Application Server Version 6.x deployment manager to a Version 8.5 deployment manager.

   After you migrate the Version 6.x deployment manager to a Version 8.5 deployment manager, you are no longer able to use the Version 6.x deployment manager. You are only able to use the Version 8.5 deployment manager.

   The Version 6.x nodes can run in a Version 8.5 WebSphere Application Server, Network Deployment cell.

   Tip: Before migrating a WebSphere Application Server Version 6.x deployment manager, use the **backupConfig** command or your own preferred backup utility to back up your existing configuration if you want to be able to restore it to its previous state after migration. Read the "backupConfig command" article in the information center for more information. Make sure that you note the exact name and location of this backed-up configuration.
2. Migrate each WebSphere Application Server Version 6.x node to Version 8.5.

   After you migrate the Version 6.x node to a Version 8.5 node, you are no longer able to use the Version 6.x node. You are only able to use the Version 8.5 node.

   Tip: When migrating a WebSphere Application Server Version 6.x federated node, you must perform the following actions if you want to be able to roll it back to its previous state after migration:
   a. Back up your existing configuration using the **backupConfig** command or your own preferred backup utility.
      - Run the **backupConfig** command or your own preferred utility to back up the Version 8.5 deployment manager configuration.

        Important: Make sure that you note the exact name and location of this backed-up configuration.

        Read the "backupConfig command" article in the information center for more information.
      - Run the **backupConfig** command or your own preferred utility to back up the Version 6.x federated node configuration.

        Important: Make sure that you note the exact name and location of this backed-up configuration.

        Read the "backupConfig command" article in the information center for more information.

   b. Migrate the federated node.

   If necessary, you can now roll back the federated node that you just migrated. Read “Rolling back a federated node” on page 63 for more information.

Procedure

1. Create a WebSphere Application Server Version 8.5 deployment-manager management profile to receive the Version 6.x deployment manager configuration.

   This step can be skipped if you are migrating to the Version 8.5 default profile.

   a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b. Run the **dspwasinst** script to display information about the Version 6.x deployment manager profile.

      Use the following parameters:

```
app_server_root/bin/dspwasinst
 -instance 6.x_profile_name
```

where

- *app_server_root* is the location of the Version 6.x installation that contains the deployment manager to be migrated
- *6.x_profile_name* is the name of the Version 6.x deployment manager profile that is to be migrated

The name of the Version 6.x node is listed in the **Node** section, and the name of the cell is listed in the **Cell** section.

Also make note of the **Name service port** setting in the **Additional ports** section. This setting will be used as the starting point when you create the new Version 8.5 deployment-manager management profile in the next step.

c. Run the **manageprofiles** command.

Use the following parameters:

```
app_server_root/bin/manageprofiles
 -create
 -profileName 80ND_profile_name
 -startingPort starting_port_number
 -templatePath app_server_root/profileTemplates/dmgr
 -cellName  6x_cell_name
 -nodeName  6x_node_name
```

where

- *app_server_root* is the location where Version 8.5 is installed
- *80ND_profile_name* is the name of your Version 8.5 deployment-manager management profile

  This parameter must be identical to the Version 6.x instance or profile that is to be migrated.
- *starting_port_number* is the first of a block of 10 consecutive ports
- *6x_cell_name* is the name of the Version 6.x cell obtained in the previous step

  The source and target cell names must be identical when migrating to Version 8.5.
- *6x_node_name* is the Version 6.x node name obtained in the previous step

  The source and target node names must be identical when migrating to Version 8.5.

For details on the syntax and parameters of the **manageprofiles** command, read the "manageprofiles command" article in the information center.

2. Save the WebSphere Application Server Version 6.x deployment manager configuration.

a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

Enter the following command from a command line:

```
STRQSH
```

b. Run the **WASPreUpgrade** script.

Use the following parameters:

```
app_server_root/bin/WASPreUpgrade
 backup_directory_name
 old_profile_root
```

where

- *app_server_root* is the location where Version 8.5 is installed
- *backup_directory_name* (required parameter) is the fully qualified path to the integrated file system directory where the **WASPreUpgrade** migration tool stores the saved configuration and files

  The directory is created if it does not already exist. Additionally, the tool writes a log file called `WASPreUpgrade.log` that chronicles the steps taken by the **WASPreUpgrade** command.

- *old_profile_root* (required parameter) is the path to the Version 6.x instance or profile to be migrated

For a full explanation of the **WASPreUpgrade** command and its parameters, read "WASPreUpgrade command" on page 34.

3. Restore the WebSphere Application Server Version 6.x deployment manager configuration into the Version 8.5 deployment-manager management profile.
   a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b. Run the **WASPostUpgrade** script.

      Use the following parameters:

      ```
      app_server_root/bin/WASPostUpgrade
       backup_directory_name
       -profileName 80ND_profile_name
       -replacePorts true
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *backup_directory_name* (required parameter) is the fully qualified path to the integrated file system directory that the **WASPreUpgrade** migration tool previously used to save the Version 6.x deployment manager configuration
      - *80ND_profile_name* (required parameter) is the name of the Version 8.5 deployment-manager management profile to which the script migrates your configuration
      - `-replacePorts true` replaces all virtual host alias port and transport settings in the Version 8.5 profile with the settings from the Version 6.x instance or profile during migration

      For a full explanation of the **WASPostUpgrade** command and its parameters, read "WASPostUpgrade command" on page 37.

4. Start the WebSphere Application Server Version 8.5 deployment-manager management profile.
   a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b. Verify that the Version 6.x deployment manager, node agent, and federated nodes are stopped for the Version 6.x deployment manager that was migrated.
   c. If the QWAS85 subsystem has not been started, start the default profile.

      Enter the following command from a command line:

      ```
      STRSBS QWAS85/QWAS85
      ```

   d. Start the Version 8.5 deployment manager using the **startManager** script.

      Use the following parameters:

      ```
      app_server_root/bin/startManager
       -profileName 80ND_profile_name
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *80ND_profile_name* is the name of the Version 8.5 deployment-manager management profile

   e. Start the Version 6.x node agent and federated nodes.
      1) Start the Version 6.x node agent using the **startNode** script.

         Use the following parameters:

         ```
         app_server_root/bin/startNode
          -instance 6.x_profile_name
         ```

where

- *app_server_root* is the location of the Version 6.x installation that contains the federated node
- *6.x_profile_name* is the name of the Version 6.x instance or profile for the federated node

2) Start the Version 6.x federated node using the **startServer** script.

Use the following parameters:

```
app_server_root/bin/startServer
 -instance 6.x_profile_name
 6.x_application_server_name
```

where

- *app_server_root* is the location of the Version 6.x installation that contains the federated node
- *6.x_profile_name* is the name of the Version 6.x instance or profile for the federated node
- *6.x_application_server_name* is the name of the Version 6.x application server

5. Migrate the WebSphere Application Server Version 6.x federated nodes to Version 8.5.
   a. Verify that the Version 6.x node agent and federated node are stopped for the federated profile that is to be migrated to Version 8.5.
   b. Verify that the Version 8.5 deployment manager is running.
   c. Complete the following instructions for each Version 6.x federated node that is to be migrated to Version 8.5.
      1) Create a WebSphere Application Server Version 8.5 profile to receive the Version 6.x configuration.
         a) Start the Qshell environment so that you can run WebSphere Application Server scripts.

            Enter the following command from a command line:

            ```
            STRQSH
            ```

         b) Run the **dspwasinst** script to obtain the node name and server name for the Version 6.x instance or profile that is to be migrated.

            Use the following parameters:

            ```
            app_server_root/bin/dspwasinst
             -instance 6x_profile_name
            ```

            where

            - *app_server_root* is the location of the Version 6.x installation that contains the instance or profile to be migrated
            - *6x_profile_name* is the name of the Version 6.x instance or profile that is to be migrated

            The name of the Version 6.x node is listed in the **Node** section, and the name of the server is listed in the **Information for server** section.

         c) Run the **manageprofiles** script.

            Use the following parameters:

            ```
            app_server_root/bin/manageprofiles
             -create
             -profileName 80_profile_name
             -startingPort starting_port_number
             -templatePath app_server_root/profileTemplates/default
             -serverName 6x_application_server_name
             -nodeName 6x_node_name
            ```

            where

            - *app_server_root* is the location where Version 8.5 is installed
            - *80_profile_name* is the name of your Version 8.5 profile

This parameter must be identical to the Version 6.x instance or profile that is to be migrated.

- *starting_port_number* is the first of a block of 13 consecutive ports
- *6x_application_server_name* is the name of the Version 6.x application server obtained in the previous step
- *6x_node_name* is the Version 6.x node name obtained in the previous step

  The source and target node names must be identical when migrating to Version 8.5.

For details on the syntax and parameters of the manageprofiles command, read the "manageprofiles command" article in the information center.

Tip: If you make any cell-level changes to the new Version 8.5 node before migration, such as changes to virtual-host information, these changes will be lost during migration. Therefore, you should wait until after the node has been migrated before making any such changes. Otherwise, you will have to manually remake all of the changes, such as any changes to the virtual-host and host-alias information, to the new cell after migration using the administrative console running on the deployment manager. This tip is reflected in message MIGR0444W.

Skip the instruction that tells you to start the Version 8.5 profile that receives the WebSphere Application Server Version 6.x configuration.

Specify `-replacePorts true` when you run the **WASPostUpgrade** script. This allows the Version 8.5 federated node to use the same virtual host ports and transport ports as the Version 6.x federated node.

2) Save the WebSphere Application Server Version 6.x configuration.
   a) Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b) Run the **WASPreUpgrade** script.

      Use the following parameters:

      ```
      app_server_root/bin/WASPreUpgrade
       backup_directory_name
       profile_root
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *backup_directory_name* (required parameter) is the fully qualified path to the integrated file system directory where the **WASPreUpgrade** migration tool stores the saved configuration and files

        The directory is created if it does not already exist. It is also the directory where the **WASPreUpgrade** migration tool writes a log file called `WASPreUpgrade.log` that chronicles the steps taken by the **WASPreUpgrade** command.
      - *profile_root* (required parameter) is the path to the Version 6.x instance or profile that is to be migrated

      For a full explanation of the **WASPreUpgrade** command and its parameters, read “WASPreUpgrade command” on page 34.
3) Restore the WebSphere Application Server Version 6.x configuration into a Version 8.5 profile.
   a) Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b) Run the **WASPostUpgrade** script.

Use the following parameters:

```
app_server_root/bin/WASPostUpgrade
 backup_directory_name
 -profileName 80_profile_name
 [-portBlock port_starting_number]
```

where

- *app_server_root* is the location where Version 8.5 is installed
- *backup_directory_name* is the required name of the directory in which the **WASPreUpgrade** tool stored the saved configuration and files and from which the **WASPostUpgrade** tool reads the configuration and files
- *80_profile_name* is the name of the Version 8.5 profile to which the script migrates your configuration
- *port_starting_number* specifies the first of a block of 10 to 15 consecutive port numbers that are not in use on the iSeries server where the migration is being performed

  It is recommended that you always specify the -portBlock parameter if you do not want your profile's ports to conflict with the default profile's ports.

For a full explanation of the **WASPostUpgrade** command and its parameters, read "WASPostUpgrade command" on page 37.

4) Start the WebSphere Application Server Version 8.5 node agent and federated nodes.
   a) Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b) Start the Version 8.5 node agent using the **startNode** script.

      Use the following parameters:

      ```
      app_server_root/bin/startNode
       -profileName 8.0_profile_name
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *8.0_profile_name* is the name of the Version 8.5 profile for the federated node

   c) Start the Version 8.5 federated node using the **startServer** script.

      Use the following parameters:

      ```
      app_server_root/bin/startServer
       -profileName 8.0_profile_name
       8.0_application_server_name
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *8.0_profile_name* is the name of the Version 8.5 profile for the federated node
      - *8.0_application_server_name* is the name of the Version 8.5 application server

Migrating a large WebSphere Application Server, Network Deployment configuration with a large number of applications

If you have an existing WebSphere Application Server Version 6.x WebSphere Application Server, Network Deployment configuration with a significant number of large applications and you must meet a specific maintenance window for migration, you might have some difficulty if you use the standard migration scenario. In this case, you might want to copy the resources in the configuration tree from a Version 6.x deployment manager configuration to a Version 8.5 deployment-manager management profile but defer adding applications to the Version 8.5 profile so that you can continue managing the environment using the Version 6.x deployment manager.

Before you begin

Tip: To avoid possible connection-timeout problems, modify the connection-timeout value before running the **WASPostUpgrade** command to migrate the federated nodes in a cell containing many small applications, a few large applications, or one very large application. If you use a SOAP connector, for example, perform the following actions:

1. Go to the following location in the Version 8.5 directory for the profile to which you are migrating your federated node:

   profile_root/properties
2. Open the soap.client.props file in that directory and find the value for the com.ibm.SOAP.requestTimeout property. This is the timeout value in seconds. The default value is 180 seconds.
3. Change the value of com.ibm.SOAP.requestTimeout to make it large enough to migrate your configuration. For example, the following entry would give you a timeout value of a half of an hour:

   com.ibm.SOAP.requestTimeout=1800

   Note: Select the smallest timeout value that will meet your needs. Be prepared to wait for at least three times the timeout that you select—once to download files to the backup directory, once to upload the migrated files to the deployment manager, and once to synchronize the deployment manager with the migrated node agent.
4. Go to the following location in the backup directory that was created by the **WASPreUpgrade** command:

   backupDirectory/profiles/*profile_name*/properties
5. Open the soap.client.props file in that directory and find the value for the com.ibm.SOAP.requestTimeout property:
6. Change the value of com.ibm.SOAP.requestTimeout to the same value that you used in the Version 8.5 file.

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6. For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

About this task

You can use this strategy to satisfy your specific maintenance-window requirement by building the full WebSphere Application Server Version 8.5 WebSphere Application Server, Network Deployment configuration in the background while the existing topology is still running and being managed.

For help in troubleshooting problems when migrating, read Chapter 14, “Troubleshooting migration,” on page 109.

Procedure

1. Make sure that the WebSphere Application Server Version 6.x deployment manager is running and managing the existing environment, and make sure that no Version 8.5 deployment manager is running.

   This is important in order to prevent two different deployment managers from trying to manage the same environment.
2. IBM i Start the Qshell environment so that you can run WebSphere Application Server scripts.

   Enter the following command from a command line:

   STRQSH
3. Run the **WASPreUpgrade** command. IBM i

Use the following parameters:

```
app_server_root/bin/WASPreUpgrade
 backup_directory_name
 old_profile_root
```

where

- *app_server_root* is the location where Version 8.5 is installed
- *backup_directory_name* (required parameter) is the fully qualified path to the integrated file system directory where the **WASPreUpgrade** migration tool stores the saved configuration and files

  The directory is created if it does not already exist. Additionally, the tool writes a log file called `WASPreUpgrade.log` that chronicles the steps taken by the **WASPreUpgrade** command.
- *old_profile_root* (required parameter) is the path to the Version 6.x instance or profile to be migrated

For a full explanation of the **WASPreUpgrade** command and its parameters, read "WASPreUpgrade command" on page 34.

4. Run the **WASPostUpgrade** command. IBM i

   Use the following parameters:

   ```
   app_server_root/bin/WASPostUpgrade
    backup_directory_name
    -profileName 80ND_profile_name
    -includeApps script
    -keepDmgrEnabled true
   ```

   where

   - *app_server_root* is the location where Version 8.5 is installed
   - *backup_directory_name* (required parameter) is the fully qualified path to the integrated file system directory that the **WASPreUpgrade** migration tool previously used to save the Version 6.x deployment manager configuration
   - *80ND_profile_name* (required parameter) is the name of the Version 8.5 deployment-manager management profile to which the script migrates your configuration

   For a full explanation of the **WASPostUpgrade** command and its parameters, read "WASPostUpgrade command" on page 37.

   At this point, you can exit the maintenance window and still manage the environment using the WebSphere Application Server Version 6.x deployment manager.
5. Customize the administration files.
   a. Go to the migration backup directory location that contains the generated administration files.
   b. Combine and tailor the administration files as needed.

      This might include grouping applications together in some administration files or specifying the `installedApplications` directory using the installed.ear.destination parameter .
6. IBM i Start the Qshell environment so that you can run WebSphere Application Server scripts.

   Enter the following command from a command line:

   ```
   STRQSH
   ```
7. Run the **wsadmin** command to install the applications.
   - Install the applications in the Version 8.5 configuration during either normal operations or in applicable maintenance windows.
   - Specify `-conntype NONE`. For example:

     ```
     wsadmin -f application_script -conntype NONE
     ```

   After all applications have been installed, you are ready to start using the WebSphere Application Server Version 8.5 deployment manager.
8. Stop the WebSphere Application Server Version 6.x deployment manager.

This is important in order to prevent two different deployment managers from trying to manage the same environment.

You can do this in a number of ways. One easy way is to rename the `serverindex.xml` file in the node directory of the Version 6.x deployment manager to something else.

9. Start the WebSphere Application Server Version 8.5 deployment manager. IBM i
   a. Start the Qshell environment so that you can run WebSphere Application Server scripts.

      Enter the following command from a command line:

      ```
      STRQSH
      ```

   b. If the QWAS85 subsystem has not been started, start the default profile.

      Enter the following command from a command line:

      ```
      STRSBS QWAS85/QWAS85
      ```

   c. Start the Version 8.5 deployment manager using the **startManager** script.

      Use the following parameters:

      ```
      app_server_root/bin/startManager
       -profileName 80ND_profile_name
      ```

      where

      - *app_server_root* is the location where Version 8.5 is installed
      - *80ND_profile_name* is the name of the Version 8.5 deployment-manager management profile

Results

At this point, the WebSphere Application Server Version 8.5 deployment manager should be running and the normal application synchronization should occur.

You can follow either of the following procedures:

- Migrate the entire cell before installing the applications.
- Perform the following actions:
  1. Install the applications and leave the cell in a mixed state.
  2. When you are ready, modify the connection-timeout values (as described in the tip at the beginning of this article) before running the **WASPostUpgrade** command to migrate the federated nodes.

Migrating IBM Cloudscape or Apache Derby databases

The migration tools migrate any IBM Cloudscape database instances to Apache Derby instances in the new configuration, and they copy any Apache Derby instances that are stored in the previous release's WebSphere Application Server configuration tree to the new release's configuration tree. After you use the migration tools, you should verify the results of the database migration and manually migrate any Cloudscape database instances or copy any Derby database instances that are not automatically migrated or copied by the tools.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6. For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

Tips:

- Before you run the migration tools, ensure that any application servers hosting applications that are using a Cloudscape or a Derby database are closed.

  Otherwise, the database migration will fail.

- Before you run the migration tools, ensure that the *debug migration trace* is active.

  By default, this trace function is enabled. To reactivate the debug migration trace if it is disabled, set one of the following trace options:

  – `all traces*=all`
  – `com.ibm.ws.migration.WASUpgrade=all`

About this task

WebSphere Application Server Version 8.5 requires Apache Derby Version 10.3 or later. Apache Derby Version 10.3 is a pure Java database server that combines the Derby runtime with the opportunity to use the full services of IBM Software Support. For comprehensive information about Apache Derby Version 10.3, read the Apache Derby website.

For help, read Chapter 14, “Troubleshooting migration,” on page 109.

Important: Derby-to-Derby migration performs a file-system copy of the data at a given point in time. This snapshot will not remain in sync with the database in the previous installation. If you roll back to the previous release, any updates to the database that you made after migration will not be reflected in the previous installation.

Procedure

1. Migrate the configuration to Version 8.5.
2. Verify the automatic migration of Cloudscape database instances or copying of Derby database instances.

   When you migrate from WebSphere Application Server Version 6.1 or above to Version 8.5, the migration tools automatically upgrade Cloudscape or Derby database instances that are accessed through the embedded framework by some internal components such as the UDDI registry. The tools also attempt to upgrade Cloudscape or Derby instances that your applications access through the embedded framework. You must verify these migration results after running the migration tools.

   - To distinguish between a partially and a completely successful Cloudscape-to-Derby migration, verify the automatic-migration results by performing the following tasks:

     a. Check the general migration post-upgrade log for database error messages.

        These exceptions indicate database migration failures. The migration tool references all database exceptions with the prefix `DSRA`.

     b. Check the individual database migration logs.

        These logs have the same timestamp as that of the general migration post-upgrade log. The individual logs display more detail about errors that are listed in the general post-upgrade log as well as expose errors that are not documented by the general log.

        The path name of each database log is *app_server_root*`/profiles/`*profileName*`/logs/`*myFulldbPathName*`_migrationLog`*timestamp*`.log`.

     c. Look at the debug log that corresponds with the database migration log.

        The WebSphere Application Server migration utility triggers a *debug migration trace* by default; this trace function generates the database debug logs.

        The full path name of each debug log is *app_server_root*`/profiles/`*profileName*`/logs/`*myFulldbPathName*`_migrationDebug`*timestamp*`.log`.

   Performing these tasks gives you vital diagnostic data to troubleshoot the partially migrated databases as well as those that fail automatic migration completely. Ultimately, you must migrate databases that were not completely migrated automatically through a manual process. The log messages contain the exact old and new database path names that you must use to run the manual migration. Note these new path names precisely.

Read the "Verifying the Cloudscape automatic migration" article in the information center for more information.

- Verify that any Derby database instances that are stored in the previous release's WebSphere Application Server configuration tree were copied to the new release's configuration tree

  Check the general migration post-upgrade log for database error messages. These exceptions indicate database migration failures. The migration tool references all database exceptions with the prefix `DSRA`..

3. Manually migrate Cloudscape database instances or copy Derby database instances where necessary.
   - The Version 8.5 migration tools do not attempt to migrate database instances that transact with applications through the Cloudscape Network Server or the Apache Derby Network Server framework. This exclusion eliminates the risk of corrupting third-party applications that access the same database instances as those accessed by WebSphere Application Server.

     To minimize the risk of migration errors for databases that were only partially upgraded during the automatic migration process, delete the new database. Troubleshoot the original database according to the log diagnostic data, then perform manual migration of the original database.

     Read the "Upgrading Cloudscape manually" article in the information center for more information.
   - The Version 8.5 migration tools do not copy any Derby database instances outside the WebSphere Application Server configuration tree.

     If migration does not copy a Derby database instance automatically, copy the database instance manually.
4. Manually migrate your UDDI registry if it uses a database on the Cloudscape Network Server or the Apache Derby Network Server framework.

   Read the "Migrating the UDDI registry" article in the information center for more information.

What to do next

Service integration bus-enabled web services use a Service Data Objects (SDO) repository for storing and serving WSDL definitions. If you migrate a configuration that uses a Cloudscape database as the SDO repository, the SDO application will still be configured to use Cloudscape in the new configuration. Migration converts the Cloudscape database to Derby, but you must still update any SDO application's backend ID to use the new database. After you migrate all of the nodes on a server with an SDO repository application that uses Cloudscape, perform the following actions to reset the database type used by the SDO application on the new configuration to Derby:

1. Read about the basic usage for the installSdoRepository.jacl script inside the script file.
2. Run the installSdoRepository.jacl script by changing to the *app_server_root*`/bin/` directory and running the following command:

   `wsadmin.`*`extension`*` -f `*`app_server_root`*`/bin/installSdoRepository.jacl -editBackendId DERBY_V10`

Read the "Installing and configuring the SDO repository" article in the information center for more information on upgrading the SDO repository application to Version 8.5 .

Migrating from the WebSphere Connect JDBC driver

WebSphere Application Server Version 8.5 does not include the WebSphere Connect JDBC driver for SQL Server. Use the **WebSphereConnectJDBCDriverConversion** command to convert data sources from the WebSphere Connect JDBC driver to the DataDirect Connect JDBC driver or the Microsoft SQL Server JDBC driver. The **WebSphereConnectJDBCDriverConversion** command processes `resources.xml` files, and there are many options that can be specified to indicate which `resources.xml` files to process.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6.

About this task

gotcha: If you are running in a mixed node environment, you must issue this command for each node, specifying a specific cell name for the cellName parameter, and a specific node name for the nodeName parameter. If you do not issue these commands, you will get the following exception for your back-level nodes:

```
IGR0480E: The node node–name specified for the -nodeName parameter
is not at the current release. Migrate this node to the current release and rerun the tool.
```

Syntax

IBM i

```
WebSphereConnectJDBCDriverConversion
   [-profileName profile_name]
   [-driverType MS | DD]
   [-classPath class_path]
   [-nativePath native_path]
   [-pathSeparator separator]
   [[-cellName ALL | cell_name [-clusterName ALL | cluster_name] |
     [-applicationName ALL | application_name] |
     [-nodeName ALL | node_name] [-serverName ALL | server_name]]]
   [-backupConfig true | false]
   [-username userID]
   [-password password]
   [[-traceString trace_spec [-traceFile file_name]]
```

Parameters

-profileName *profile_name*

This optional parameter is used to specify a specific profile configuration in the Version 8.5 environment.

If this parameter is not specified, the default profile is used. If the default is not set or cannot be found, an error is returned.

If the command is run from within the *profile_name*/bin directory, it is implicit that the *profile_name* profile should be migrated and this parameter is not needed.

-driverType MS | DD

This required parameter is used to indicate the type of conversion that you want to perform.

Specifying MS converts to the Microsoft SQL Server JDBC Driver, and specifying DD converts to the DataDirect Connect JDBC driver.

-classPath *class_path*

This required parameter is used to specify the class path for the new JDBC driver.

-nativePath *native_path*

This optional parameter is used to specify the native path for the new JDBC driver.

-pathSeparator *separator*

This optional parameter is used to specify a path separator other than the default.

The default is operating-system dependant and is listed in the command-line help.

-cellName ALL | *cell_name*

This optional parameter is used to specify the specific name of the cell to process or to specify all cells.

The default is ALL. If ALL is specified, the values specified for the nodeName, serverName, applicationName, and clusterName parameters are ignored, and the settings for these parameters revert to their default values of ALL.

-clusterName ALL | *cluster_name*
This optional parameter is used to specify the specific name of the cluster to process or to specify all clusters.

The default is ALL

-application nameName ALL | *application_name*
This optional parameter is used to specify applications to process. (Some `resource.xml` files might exist in applications if enhanced EAR file support is used.)

The default is ALL

-nodeName ALL | *node_name*
This optional parameter is used to specify the specific name of the node to process or to specify all node names in the configuration. If you specify a specific nodeName, you must also specify a specific cell name on the cellName parameter. If a specific cell name is not specified, the default value of ALL is used for this parameter.

If ALL is specified for this parameter, the value specified for the serverName parameter is ignored, and the setting for the serverName parameter reverts to its default value of ALL.

-serverName ALL | *server_name*
This optional parameter is used to specify the specific name of a server or to specify all server names in a node.

The default is ALL.

-backupConfig true | false
This parameter is used to specify whether (true) or not (false) to back up the existing configuration before the command makes changes to the configuration.

The default is true.

-username *userID*
This optional parameter is used to specify the user name to be used by this command.

-password *password*
This optional parameter is used to specify the password to be used by this command.

-traceString *trace_spec*
This optional parameter is used with -traceFile to gather trace information for use by IBM service personnel.

The value of traceString is "*=all=enabled" and must be specified with quotation marks to be processed correctly.

-traceFile *file_name*
This optional parameter is used with -traceString to gather trace information for use by IBM service personnel.

Rolling back environments

After migrating to a WebSphere Application Server Version 8.5 environment, you can roll back to a Version 6.1 or above environment. This returns the configuration to the state that it was in before migration. After rolling back the environment, you can restart the migration process.

Before you begin

About this task

Generally, migration does not modify anything in the configuration of the prior release; however, there are cases where minimal changes are made that are reversible.

Procedure

- To roll back a Version 8.5 deployment manager and its federated nodes to Version 6.1 or above, follow the instructions in “Rolling back a WebSphere Application Server, Network Deployment cell.”
- To roll back a Version 8.5 federated node to Version 6.1 or above, follow the instructions in “Rolling back a federated node” on page 63.
- To roll back a Version 8.5 stand-alone application server to Version 6.1 or above, follow the instructions in “Rolling back stand-alone application servers” on page 65.

Results

The configuration should now be returned to the state that it was in before migration.

What to do next

You can now restart the migration process if you want to do so.

Rolling back a WebSphere Application Server, Network Deployment cell

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Application Server Version 8.5 WebSphere Application Server, Network Deployment cell to Version 6.x or 7.x. This returns the configuration to the state that it was in before migration. After rolling back the WebSphere Application Server, Network Deployment cell, you can restart the migration process.

Before you begin

Best practice: When migrating a Version 6.x or 7.x WebSphere Application Server, Network Deployment cell, the best practice is to perform the following actions if you want to be able to roll it back to its previous state after migration:

1. Back up your existing configuration.
   - Run the **backupConfig** command or your own preferred utility to back up the Version 6.x or 7.x deployment manager configuration.

     Important: Make sure that you note the exact name and location of this backed-up configuration.

     Read the "backupConfig command" article in the information center for more information.
   - Run the **backupConfig** command or your own preferred utility to back up the Version 6.x or 7.x federated node configurations.

     Important: Make sure that you note the exact name and location of each of these backed-up configurations.

     Read the "backupConfig command" article in the information center for more information.
2. Migrate the WebSphere Application Server, Network Deployment cell.

Procedure

1. Stop all of the servers and node agents that are currently running in the Version 8.5 environment.
2. If you chose to disable the previous deployment manager when you migrated to the Version 8.5 deployment manager, perform one of the following actions.

   Note: Disablement is the default.

a. If you backed up your previous deployment manager configuration using the **backupConfig** command or your own preferred backup utility, run the **restoreConfig** command or your own preferred utility to restore the Version 6.x or 7.x configuration for the deployment manager.

   Important: Make sure that you restore the same backed-up configuration that you created just before you migrated the deployment manager.

   Read the "restoreConfig command" article in the information center for more information.

b. IBM i If you did not back up your previous deployment manager configuration, use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the Version 6.x or 7.x *profile_root*`/bin` directory of the deployment manager that you need to roll back from Version 8.5.

   In a Linux environment, for example, use the following parameters:

   ```
   ./wsadmin.sh -f migrationDisablementReversal.jacl -conntype NONE
   ```

   IBM i Use the following parameters:

   ```
   app_server_root/bin/wsadmin -instance instance -conntype NONE
   -f profile_root/bin/migrationDisablementReversal.jacl
   ```

   Tip: If you have trouble running the `migrationDisablementReversal.jacl` script, try to manually perform the steps in the script.

   1) Go to the following directory:

      profile_root`/config/cells/`*cell_name*`/nodes/`*node_name*

      where *node_name* is the name of the deployment manager node that you want to roll back.

   2) If you see a `serverindex.xml_disabled` file in this directory, perform the following actions:

      a) Delete or rename the `serverindex.xml` file.

      b) Rename the `serverindex.xml_disabled` file to `serverindex.xml`.

3. Perform one of the following actions for each of the WebSphere Application Server, Network Deployment cell's federated nodes that you need to roll back.

   a. If you backed up your previous federated node configuration using the **backupConfig** command or your own preferred backup utility, run the **restoreConfig** command or your own preferred utility to restore the Version 6.x or 7.x configuration for the federated node.

      Important: Make sure that you restore the same backed-up configuration that you created just before you migrated the federated node.

      Read the "restoreConfig command" article in the information center for more information.

   b. IBM i If you did not back up your previous federated node configuration, use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the Version 6.x or 7.x *profile_root*`/bin` directory of the federated node.

      IBM i Use the following parameters:

      ```
      app_server_root/bin/wsadmin -instance instance -conntype NONE
      -f profile_root/bin/migrationDisablementReversal.jacl
      ```

      Tip: If you have trouble running the `migrationDisablementReversal.jacl` script, try to manually perform the steps in the script.

      1) Go to the following directory:

         profile_root`/config/cells/`*cell_name*`/nodes/`*node_name*

         where *node_name* is the name of the federated node that you want to roll back.

      2) If you see a `serverindex.xml_disabled` file in this directory, perform the following actions:

         a) Delete or rename the `serverindex.xml` file.

b) Rename the `serverindex.xml_disabled` file to `serverindex.xml`.

4. Synchronize the federated nodes if they were ever running when the Version 8.5 deployment manager was running.

   Read the "Synchronizing nodes with the wsadmin tool" article in the information center for more information.

5. If you chose to keep the installed applications in the same location as the prior release during migration to Version 8.5 and any of the Version 8.5 applications are not compatible with the prior release, install applications that are compatible.

6. IBM i Delete the Version 8.5 profiles.

   Read the "Deleting a profile" article in the information center for more information.

7. Start the rolled-back deployment manager and its federated nodes in the Version 6.x or 7.x environment.

Results

The configuration should now be returned to the state that it was in before migration.

What to do next

You can now restart the migration process if you want to do so.

Rolling back a federated node

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Application Server Version 8.5 federated node to the state that it was in before migration. For each federated node that you want to roll back, you must roll back the federated node itself and the corresponding changes made to the primary repository located on the deployment manager.

Before you begin

Best practice: When migrating a Version 6.1 or above federated node, the best practice is to perform the following actions if you want to be able to roll it back to its previous state after migration:

1. Back up your existing configuration.
   a. Run the **backupConfig** command or your own preferred utility to back up the Version 8.5 deployment manager configuration.

      Important: Make sure that you note the exact name and location of this backed-up configuration.

      Read the "backupConfig command" article in the information center for more information.

   b. Run the **backupConfig** command or your own preferred utility to back up the Version 6.1 or above federated node configuration.

      Important: Make sure that you note the exact name and location of this backed-up configuration.

      Read the "backupConfig command" article in the information center for more information.

2. Migrate the federated node.
3. If necessary, you can now roll back the federated node that you just migrated.

Important: If you do not have a backup copy of your Version 8.5 deployment manager configuration as it was before you migrated the Version 6.1 or above federated node that you want to roll back, you cannot use the procedure described in this article and you must roll back your whole cell.

About this task

You must perform all of the backup and rollback actions for each migrated federated node before you proceed to migrate another federated node.

Procedure

1. Run the **backupConfig** command or your own preferred utility to back up the Version 8.5 deployment manager configuration at the current state.

   Important: Make sure that you note the exact name and location of this backed-up configuration.

   Read the "backupConfig command" article in the information center for more information.
2. Stop all servers and the node agent on the Version 8.5 federated node that you want to roll back.
3. Restore your previous configuration.
   a. Run the **restoreConfig** command or your own preferred utility to restore the previous Version 8.5 deployment manager configuration.

      Important:

      - Make sure that you restore the same backed-up configuration that you created just before you migrated the federated node.
      - If you have made changes to your environment (application or configuration changes for example), these changes are rolled back at the same time and cause the other nodes to force synchronization with the deployment manager.

      Read the "restoreConfig command" article in the information center for more information.
   b. Perform one of the following actions to restore the Version 6.1 or above configuration for the federated node.
      - Run the **restoreConfig** command or your own preferred utility to restore the Version 6.1 or above configuration.

        Important: Make sure that you restore the same backed-up configuration that you created just before you migrated this federated node.

        Read the "restoreConfig command" article in the information center for more information.
      - IBM i Use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the Version 6.1 or above *profile_root*`/bin` directory of the federated node.

        IBM i Use the following parameters:

        ```
        app_server_root/bin/wsadmin -instance instance -conntype NONE
        -f profile_root/bin/migrationDisablementReversal.jacl
        ```

        Tip: If you have trouble running the `migrationDisablementReversal.jacl` script, try to manually perform the steps in the script.

        1) Go to the following directory:

           profile_root`/config/cells/`*cell_name*`/nodes/`*node_name*

           where *node_name* is the name of the federated node that you want to roll back.
        2) If you see a `serverindex.xml_disabled` file in this directory, perform the following actions:
           a) Delete or rename the `serverindex.xml` file.
           b) Rename the `serverindex.xml_disabled` file to `serverindex.xml`.
4. Start the Version 8.5 deployment manager.
5. Perform any application maintenance that is required.
6. Synchronize the Version 6.1 or above federated node with the deployment manager.

Read the "Synchronizing nodes with the wsadmin tool" article in the information center for more information.

7. If you chose to keep the installed applications in the same location as the prior release during migration to Version 8.5 and any of the Version 8.5 applications are not compatible with the prior release, install applications that are compatible.
8. Start the rolled-back Version 6.1 or above federated node and servers.
9. Validate that the configuration is satisfactory.

   This is the last chance to undo the rollback action by restoring the deployment-manager configuration that you backed up in the first step.
10. IBM i Delete the Version 8.5 profile for the federated node that you rolled back to Version 6.1 or above.

    Read the "Deleting a profile" article in the information center for more information.

Results

The configuration should now be returned to the state that it was in before migration.

What to do next

You can now restart the migration process if you want to do so.

Rolling back stand-alone application servers

You can use the **restoreConfig** and **wsadmin** commands to roll back a migrated WebSphere Application Server Version 8.5 stand-alone application server to the state that it was in before migration.

Before you begin

Best practice: When migrating a Version 6.x or 7.x stand-alone application server, the best practice is to perform the following actions if you want to be able to roll it back to its previous state after migration:

1. Run the **backupConfig** command or your own preferred utility to back up the Version 6.x or 7.x stand-alone application server configuration.

   Important: Make sure that you note the exact name and location of this backed-up configuration.

   Read the "backupConfig command" article in the information center for more information.
2. Migrate the stand-alone application server.
3. If necessary, you can now roll back the stand-alone application server that you just migrated.

Procedure

1. Stop all of the servers that are currently running in the Version 8.5 environment.
2. Perform one of the following actions to restore the Version 6.x or 7.x configuration for the stand-alone application server.
   - Run the **restoreConfig** command or your own preferred utility to restore the Version 6.x or 7.x configuration.

     Important: Make sure that you restore the same backed-up configuration that you created just before you migrated this stand-alone application server.

     Read the "restoreConfig command" article in the information center for more information.

- IBM i Use the **wsadmin** command to run the `migrationDisablementReversal.jacl` script from the Version 6.x or 7.x *profile_root*`/bin` directory of the stand-alone application server.

  IBM i Use the following parameters:

  ```
  app_server_root/bin/wsadmin -instance instance -conntype NONE
  -f profile_root/bin/migrationDisablementReversal.jacl
  ```

3. If you chose to keep the installed applications in the same location as the prior release during migration to Version 8.5 and any of the Version 8.5 applications are not compatible with the prior release, install applications that are compatible.
4. IBM i Delete the Version 8.5 profile for the stand-alone application server.

   Read the "Deleting a profile" article in the information center for more information.
5. Start the rolled-back stand-alone application server in the Version 8.5 environment.

Results

The configuration should now be returned to the state that it was in before migration.

What to do next

You can now restart the migration process if you want to do so.

Chapter 5. Scenario 1: Migrating a cell using the command line tools

Before you begin

Review the migration planning information. See Knowledge Collection: Migration planning for WebSphere Application Server.

This scenario covers migrating cells on the same host. If you intend to migrate cells to a different host, see Scenario 2: Migrating cells across operating systems using the command-line tools.

About this task

This task describes how to use the command line tools to migrate a cell from a previous version of WebSphere Application Server to Version 8.5. The cell configuration consists of a deployment manager with one or more nodes, a web server, and an application client. All ports are migrated forward into the new configuration. This procedure assumes that the previous configuration is running. For more information, see clientUpgrade command.

Note: Ensure that your setting for the maximum number of open files is 10000 or greater. If the number of open files is too low, this can cause a variety of migration failures.

Procedure

1. Run the backupConfig command on the deployment manager and all old nodes In case of failure during the migration, save the current deployment manager and node configuration to a file that you can use later for recovery purposes.
   a. Change to the *<deployment manager profile root>/bin* directory.
   b. Run the backupConfig command with the appropriate parameters and save the current profile configuration to a file. For example: IBM i

   ```
   /QIBM/UserData/WebSphere/AppServer/V70/ND/profiles/v70dmgr01/bin/backupConfig /mybackupdir/
   v70dmgr01backupBeforeV85migration.zip -username myuser -password mypass -nostop
   ```

   c. For each node in the configuration, change to the *<node profile root >/bin* directory.
   d. Run the backupConfig command with the appropriate parameters, and save the current profile configuration to a file. For example: IBM i

   ```
   /QIBM/UserData/WebSphere/AppServer/V70/ND/profiles/v70node01/bin/backupConfig
   /mybackupdir/v70node01rbackupBeforeV85migration.zip -username myuser -password mypass -nostop
   ```

2. Install WebSphere Application Server Version 8.5 Install WebSphere Application Server Version 8.5 onto each target host in a new directory. For more information, see How do I install an application serving environment?
3. Create the target deployment manager profile The target deployment manager profile is a new deployment manager profile that will be the target of the migration.

   Note: The Version 8.5 profile nodeName and cellName must match the previous Version 7.0 or above nodeName and cellName. If the Version 8.5 deployment manager cellName or nodeName are different, the migration will fail.

   a. Run the manageprofiles command with the appropriate parameters to create a new deployment manager profile. For example: IBM i

   ```
   /QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/manageprofiles -create -profileName
   currentDmgrProfileName -templatePath /QIBM/ProdData/WebSphere/AppServer/V85/ND/
   profileTemplates/management -serverType DEPLOYMENT_MANAGER -nodeName
   currentDmgrNodeName -cellName currentCellName -hostName mydmgrhost.company.com
   ```

4. Run the WASPreUpgrade command from the new deployment manager profile bin directory Save the current deployment manager configuration to the migration backup directory. The WASPreUpgrade command does not make any changes to the Version 7.0 or above configuration. If you are migrating

from Version 8.0 to Version 8.5 and your profile is a deployment manager, the Version 8.0 profile is stopped during WASPreUpgrade and then restarted.

a. Run the WASPreUpgrade command to save the current deployment manager configuration information to a migration backup directory. For example: IBM i

```
/QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/WASPreUpgrade /mybackup/v70tov85dmgr01
/QIBM/UserData/WebSphere/AppServer/V85/ND/profiles/myCurrentDmgrProfile
```

5. Verify the console output and the WASPreUpgrade logs for success, warnings, or failure After the WASPreUpgrade command is complete, check the console output and logs for any warnings or errors. If there are errors, all errors must be fixed and then WASPreUpgrade must be run again. Also check the warnings to see if they will impact any other migration or runtime activities on Version 8.5. If the command completed successfully, then it is not necessary to check the logs for errors or warnings.

   a. Check the WASPreUpgrade console output for "Failed with errors" or "Completed with warnings".

   b. Look in the following logs for warnings or errors:

      - WASPreUpgrade.< oldProfile >.< timestamp >.log
      - WASPreUpgrade.trace

6. Run the WASPostUpgrade command from the new deployment manager profile bin directory Use the WASPostUpgrade command to restore the previous deployment manager configuration that you saved in the migration backup directory. If you use the options shown in the example, all the ports will be carried forward, the old deployment manager will be shutdown and disabled, and all applications will be installed.

   a. Run the WASPostUpgrade command to restore the saved deployment manager configuration into the new Version 8.5 deployment manager profile. For example: IBM i

```
/QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/WASPostUpgrade /mybackup/v70tov85dmgr01
-profileName myCurrentDmgrProfile -oldProfile myCurrentDmgrProfile -replacePorts TRUE
-backupConfig TRUE -includeApps TRUE -scriptCompatibility TRUE -keepDmgrEnabled FALSE
-username myuser -password mypass
```

   When creating profiles, only one profile is considered the default per installation.

   The WASPostUpgrade command migrates the default profile for the source and the target installations if the -oldProfile or the -profileName parameters are not specified, respectively.

   The default profiles can be identified by looking in the `profileRegistry.xml` file found in the WAS_HOME/properties directory. The source `profileRegistry.xml` is copied to the migration backup directory as part of the WASPreUpgrade command.

   gotcha:

      - The -oldProfile and -profileName parameters should always be specified when running the WASPostUpgrade command.
      - The script compatibility flag on your deployment manager must be the same as the flag that you use on your nodes. Save the value of the script compatibility flag for later use.

7. Verify the logs for success Once the WASPostUpgrade command is completed, check the console output and logs for any warnings or errors. If there are errors, check the error codes against the Version 8.5 Information Center for more information. Check the warnings to see if they will impact any node migration or runtime activities once the Version 8.5 deployment manager is started. If the command completed successfully, then it is not necessary to check the logs for errors or warnings.

   a. Check the WASPostUpgrade console output for "Failed with errors" or "Completed with warnings".

   b. Look in the following logs for warnings or errors:

      - <migration backupdir>/logs/WASPostUpgrade.<target profile name>.< timestamp >.log
      - <migration backupdir>/logs/WASPostUpgrade.<target profile name>.trace

8. Run the backupConfig command on the Version 8.5 deployment manager Save the Version 8.5 migrated deployment manager configuration to a file.

Note: This is an important step in the cell migration plan. If there are any node migration failures, the cell configuration can be restored to the point prior to the failure, remedial actions can be applied, and the node migration can be attempted again.

a. Change to the *<deployment manager profile root>/bin* directory

b. Run the backupConfig command with the appropriate parameters and save the Version 8.5 profile configuration to a file. For example: IBM i

```
/QIBM/UserData/WebSphere/AppServer/V85/ND/profiles/myCurrentDmgrProfile/bin/
backupConfig.sh /mybackupdir/v70tov85dmgr01backupMigratedDmgrOnly.zip
-username myuser -password mypass
```

9. Start the Version 8.5 deployment manager Ensure that the previous version of the deployment manager is not running.

   a. Change to the new Version 8.5 deployment manager profile bin directory.

   b. Run the startManager command.

   c. While the deployment manager is running, check the SystemOut.log file for warnings or errors.

      Note: This topic references one or more of the application server log files. As a recommended alternative, you can configure the server to use the High Performance Extensible Logging (HPEL) log and trace infrastructure instead of using `SystemOut.log` , `SystemErr.log`, `trace.log`, and `activity.log` files on distributed and IBM i systems. You can also use HPEL in conjunction with your native z/OS logging facilities. If you are using HPEL, you can access all of your log and trace information using the LogViewer command-line tool from your server profile bin directory. See the information about using HPEL to troubleshoot applications for more information on using HPEL.

   d. Check all of the node's nodeagent and application server logs for new warnings or errors. If "Automatic synchronization" is enabled, then allow the node to synchronize, allow the applications to restart, and then check the logs for new warnings or errors.

10. Migrate plug-ins for web servers

    a. Ensure that the Version 8.5 deployment manager is running.

    b. Upgrade the version of the web server plugin that is used in the cell.

    c. See the supporting information that is applicable to your web server type and version.

11. Run a migration on application client installations Migrate client resources to Version 8.5 level resources.

    a. Install the WebSphere Version 8.5 Application client.

    b. Run the clientUpgrade command on your application client ear files. For more information, see clientUpgrade command.

    c. Run the Version 8.5 WASPreUpgrade command to save the Application client security settings to a migration backup directory. For example:

    ```
    /opt/AppClientV85/bin/WASPreUpgrade.sh /mybackup/v70clientTov85 /opt/AppClientV70
    ```

    d. Run the Version 8.5 WASPostUpgrade command to restore the Application client security settings to the new Version 8.5 client. For example:

    ```
    /opt/AppClientV85/bin/WASPostUpgrade.sh /mybackup/v70clientToV85
    ```

12. Migrate nodes Ensure that the Version 8.5 deployment manager is running. Use the migration tooling to migrate the previous versions of the nodes in the configuration to Version 8.5. For each node that you plan to migrate to Version 8.5, perform the following steps:

    Note: For the migration to be successful, you must use the same source node name and cell name for each node that you migrate to Version 8.5.

    a. Create the target node profile. Run the manageprofiles command with the appropriate parameters to create a new managed profile. For example: IBM i

    ```
    /QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/manageprofiles -create
    -profileName currentNode1Name -templatePath /QIBM/ProdData/WebSphere/AppServer
    /V85/ND/profileTemplates/managed -nodeName currentNode1Name -cellName currentCellName
    -hostName mynode1host.company.com
    ```

b. Run the WASPreUpgrade command to save the current node X configuration information to a migration backup directory. Choose a new directory for the backup files. For example: IBM i

```
/QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/WASPreUpgrade /mybackup/
v70tov85node1 /QIBM/UserData/WebSphere/AppServer/V70/ND/profiles/currentNode1Name
```

c. If the WASPreUpgrade command completed with Success, then checking the logs for errors or warnings is not necessary.

d. Check the WASPreUpgrade console output for the following messages: "Failed with errors" or "Completed with warnings".

e. Look in the following logs for warnings or errors:

- <migration backupdir>/logs/WASPreUpgrade.< oldProfile >< timestamp >.log
- <migration backupdir>/logs/WASPreUpgrade.trace

f. Stop the node agent. If you have Version 7.0 or above nodes running during a migration to Version 8.5, you must stop the node agent on the node being migrated. If you do not stop the node agent, you might encounter corruption problems.

g. Run the WASPostUpgrade command to restore the saved node X configuration into the new Version 8.5 managed profile. For example: IBM i

```
/QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/WASPostUpgrade /mybackup/
v70tov85node1 -profileName currentNode1Name -replacePorts TRUE -backupConfig
TRUE -scriptCompatibility TRUE -username myuser -password mypass
```

Note: The script compatibility flag on your deployment manager must be the same as the flag that you use on your nodes.

h. If the command completed with Success, then checking the logs for errors or warnings is not necessary.

i. Check the WASPostUpgrade console output for the following messages: "Failed with errors" or "Completed with warnings".

j. Look in the following logs for errors or warnings:

- <migration backupdir>/logs/WASPostUpgrade.<target profile>.< timestamp >.log
- <migration backupdir>/logs/WASPostUpgrade.< target profile name >.trace

Note: If the WASPostUpgrade command fails, you may have to restore the Version 8.5 deployment manager from the backupConfig file. If the WASPostUpgrade processing executed the syncNode command, then the deployment manager is aware that the node X has been migrated. The node X cannot be migrated again until the deployment manager has been restored to the state before the node X migration.

k. Check the Version 8.5 deployment manager SystemOut.log for warnings or errors.

Note: This topic references one or more of the application server log files. As a recommended alternative, you can configure the server to use the High Performance Extensible Logging (HPEL) log and trace infrastructure instead of using `SystemOut.log` , `SystemErr.log`, `trace.log`, and `activity.log` files on distributed and IBM i systems. You can also use HPEL in conjunction with your native z/OS logging facilities. If you are using HPEL, you can access all of your log and trace information using the LogViewer command-line tool from your server profile bin directory. See the information about using HPEL to troubleshoot applications for more information on using HPEL.

l. Start the migrated Version 8.5 node X agent.

m. Check the Version 8.5 deployment manager and node X SystemOut.log for warnings or errors.

n. Synchronize the cell.

o. Stop all the application servers on the Version 8.5 migrated node X.

p. Start the appropriate application servers on the Version 8.5 migrated node X.

q. Run the backupConfig command with the appropriate parameters and save the Version 8.5 profile configuration to a file. For example: IBM i

```
/QIBM/UserData/WebSphere/AppServer/V85/profiles/v70tov85node1/bin/backupConfig
/mybackupdir/v70tov85node1.zip -username myuser -password mypass -nostop
```

Each time you run the backupConfig command, use a new backup file name.

r. Save the Deployment Manager configuration using the backupConfig command. On the Version 8.5 Deployment Manager host, change to the *<deployment manager profile root>*/bin directory. Run the backupConfig command with the appropriate parameters and save the Version 8.5 profile configuration to a file. For example: IBM i

```
/QIBM/UserData/WebSphere/AppServer/V85/profiles/currentDmgrName/bin/
backupConfig.sh /mybackupdir/v70tov85dmgr01backupMigratedDmgrPlusNodeX.zip
-username myuser -password mypass
```

Note: If you are migrating a node to a different host, refer to these steps in Scenario 2: Migrating cells across operating systems using the command-line tools.

Note: For each node migrated, backup the Version 8.5 Deployment Manager configuration to a new backup file.

13. Troubleshooting See the information center documentation on troubleshooting migration.

Results

You have migrated from a previous version to WebSphere Application Server Version 8.5 using the migration tools.

Chapter 6. Scenario 3: Flexible Management: Migrating an administrative agent profile and its registered set of managed base application servers

Administrative agent profiles manage multiple base application servers in environments such as development, unit test or that portion of a server farm that resides on a single machine. Before you can migrate managed base application servers from Version 7.0 or above to Version 8.5, you must first migrate the administrative agent.

Before you begin

Review the migration planning information. See Knowledge Collection: Migration planning for WebSphere Application Server.

You can migrate administrative agent profiles and the registered set of managed base application servers from Version 7.0 or above to Version 8.5.

About this task

This task describes how to use the command line tools to migrate an administrative agent and its associated set of managed base application servers from WebSphere Application Server Version 7.0 or above to Version 8.5. A base application server becomes managed when it is registered with a single administrative agent. An administrative agent may manage one or more base application servers and must be at the same release level and on the same machine as the base application servers it is managing. Because of this restriction, the administrative agents on both the old and new release run simultaneously until all managed base application servers are migrated. The migration of an administrative agent does not bring forward its old port values, however, all other configuration data is migrated. Access the Version 8.5 admin agent console using the WC_ adminhost or WC_ adminhost_ secure ports as defined in the new Version 8.5 admin agent's serverindex.xml file. Additionally, the Version 7.0 or above administrative agent must not be shut down or disabled during this procedure. For migrating the managed base application server in a flexible management environment, ensure that the node names are the same on Version 8.5 and previous releases.

For migrating the managed base application server in a flexible management environment, ensure that the node names are the same on Version 8.5 and previous releases.

Note: Ensure that your setting for the maximum number of open files is 10000 or greater. If the number of open files is too low, this can cause a variety of migration failures.

Procedure

1. Install WebSphere Application Server Version 8.5 Install WebSphere Application Server Version 8.5 onto the target host in a new directory.
   a. Install the Network Deployment or Base version of WebSphere Application Server onto the target host. For more information, see How do I install an application serving environment?
2. Create the target administrative agent profile The target administrative agent profile is a new administrative agent profile that will be the target of the administrative agent migration.
   a. Run the manageprofiles command with the appropriate parameters to create a new administrative agent profile. For example: IBM i

   ```
   /QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/manageprofiles -create -profileName AdminAgent01 -templatePath /QIBM/ProdData/WebSph
   ```
3. Ensure that all the "In Progress" jobs are completed on the managed profiles Before running WASPreUpgrade on a managed application server or deployment manager, all in progress jobs must be complete.

4. Stop polling the job manager To avoid problems, stop polling the job manager before running the WASPreUpgrade on profiles that are getting jobs from the job manager. Before you start polling for jobs, complete WASPreUpgrade and WASPostUpgrade for the managed profile. For more information, see ManagedNodeAgent command group for the AdminTask object using wsadmin scripting.
5. Run the WASPreUpgrade command from the new WebSphere Application Server install root bin directory Save the current administrative agent configuration to the migration backup directory. The WASPreUpgrade command does not make any changes to the old configuration.
   a. Run the WASPreUpgrade command to save the current administrative agent configuration information to a migration backup directory. For example:
6. Verify the console output and the WASPreUpgrade logs for success, warnings, or failure After the WASPreUpgrade command is complete, check the console output and logs for any warnings or errors. If there are errors, all errors must be fixed and then WASPreUpgrade must be run again. Also check the warnings to see if they will impact any other migration or runtime activities on Version 8.5. If the command completed successfully, then it is not necessary to check the logs for errors or warnings.
   a. Check the WASPreUpgrade console output for "Failed with errors" or "Completed with warnings".
   b. Look in the following logs for warnings or errors:
      - *<migration backupdir>*/logs/WASPreMigrationSummary.log
      - WASPreUpgrade.< timestamp >.log
      - WASPreUpgrade.trace
7. Run the WASPostUpgrade command from the new WebSphere Application Server install root bin directory Use the WASPostUpgrade command to restore the previous administrative agent configuration that you saved in the migration backup directory.
   a. Run the WASPostUpgrade command to restore the saved administrative agent configuration into the new Version 8.5 administrative agent profile. For example:
8. Verify the logs for success Once the WASPostUpgrade command is completed, check the console output and logs for any warnings or errors. If there are errors, check the error codes against the Version 8.5 Information Center for more information. If the command completed successfully, then it is not necessary to check the logs for errors or warnings.
   a. Check the WASPostUpgrade console output for "Failed with errors" or "Completed with warnings".
   b. Look in the following logs for warnings or errors:
      - *<migration backupdir>*/logs/WASPostMigrationSummary.log
      - WASPostUpgrade.<target profile name>.< timestamp >.log
      - WASPostUpgrade.<target profile name>.trace
9. Start the Version 8.5 administrative agent Ensure that both Version 7.0 or above and Version 8.5 administrative agents are running.
   a. Change to the new Version 8.5 administrative agent profile bin directory.
   b. Run the startServer adminagent command.
   c. Check the SystemOut.log file for warnings or errors.

Note: This topic references one or more of the application server log files. As a recommended alternative, you can configure the server to use the High Performance Extensible Logging (HPEL) log and trace infrastructure instead of using `SystemOut.log` , `SystemErr.log`, `trace.log`, and `activity.log` files on distributed and IBM i systems. You can also use HPEL in conjunction with your native z/OS logging facilities. If you are using HPEL, you can access all of your log and trace information using the LogViewer command-line tool from your server profile bin directory. See the information about using HPEL to troubleshoot applications for more information on using HPEL.

10. Migrate managed base application servers Ensure that the Version 8.5 administrative agent is running. For each managed base application server that you plan to migrate to Version 8.5, perform the following steps:

    Note: For the migration to be successful:
    - Managed base application servers must be located on the same machine as the associated administrative agent.
    - The node names must be the same on the Version 8.5 and previous releases.

    a. Create the target base application server profile. Run the manageprofiles command with the appropriate parameters to create a new managed profile. For example: IBM i

    ```
    /QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/manageprofiles -create -profileName
    AppSrv01 -templatePath /QIBM/ProdData/WebSphere/AppServer/V85/ND/profileTemplates/default
    -nodeName AppSrv01Node01 -cellName AppSrv01Cell01 -hostName mynode1host.company.com
    ```

    b. Run the WASPreUpgrade command to save the current managed base application server information to a migration backup directory. Choose a new directory for the backup files. For example: IBM i

    ```
    /QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/WASPreUpgrade /mybackup/
    WAS70Appserver01backup /QIBM/UserData/WebSphere/AppServer/V70/ND/profiles/AppSrv01
    ```

    c. If the WASPreUpgrade command completed with Success, then checking the logs for errors or warnings is not necessary.

    d. Check the WASPreUpgrade console output for the following messages: "Failed with errors" or "Completed with warnings".

    e. Look in the following logs for warnings or errors:
    - *<migration backupdir>*/logs/WASPreMigrationSummary.log
    - WASPreUpgrade.< timestamp >.log
    - WASPreUpgrade.trace

    f. Run the WASPostUpgrade command to restore the saved managed application server profile configuration into the new Version 8.5 base application server profile.

    Note: This command requires additional parameters and the following example assumes that security is enabled on both administrative agents.

    For example: IBM i

    ```
    /QIBM/ProdData/WebSphere/AppServer/V85/ND/bin/WASPostUpgrade
    /mybackup/WAS70Appserver01backup -profileName AppSrv01
    -oldAdminAgentProfilePath /QIBM/UserData/WebSphere/AppServer
    /V70/ND/profiles/AdminAgent01 -oldAdminAgentHostname myhostname
    -oldAdminAgentSoapPort 8879 -oldAdminAgentUsername myusername
    -oldAdminAgentPassword mypassword -newAdminAgentProfilePath
    /QIBM/UserData/WebSphere/AppServer/V85/ND/profiles/AdminAgent01
    -newAdminAgentHostname myhostname -newAdminAgentSoapPort 8887
    -newAdminAgentUsername myusername1 -newAdminAgentPassword mypassword1
    ```

    g. If the command completed with Success, then checking the logs for errors or warnings is not necessary.

    h. Check the WASPostUpgrade console output for the following messages: "Failed with errors" or "Completed with warnings".

    i. Look in the following logs for errors or warnings:
    - *<migration backupdir>*/logs/WASPostMigrationSummary.log
    - WASPostUpgrade. <target profile>.< timestamp >.log
    - WASPostUpgrade.< target profile name >.trace

    j. Start the migrated Version 8.5 managed application server.

    k. Check the Version 8.5 managed application server SystemOut.log for warnings or errors.

    Note: This topic references one or more of the application server log files. As a recommended alternative, you can configure the server to use the High Performance Extensible Logging

(HPEL) log and trace infrastructure instead of using `SystemOut.log` , `SystemErr.log`, `trace.log`, and `activity.log` files on distributed and IBM i systems. You can also use HPEL in conjunction with your native z/OS logging facilities. If you are using HPEL, you can access all of your log and trace information using the LogViewer command-line tool from your server profile bin directory. See the information about using HPEL to troubleshoot applications for more information on using HPEL.

11. Troubleshooting See the information center documentation on troubleshooting migration.

Results

You have migrated an administrative agent profile and its associated managed base application servers from WebSphere Application Server Version 7.0 or above to Version 8.5 using the migration tools. You can stop the Version 7.0 or above administrative agent and you can assign the Version 7.0 or above ports to the Version 8.5 administrative agent.

Chapter 7. Scenario 4: Flexible Management: Migrating a job manager profile and its registered set of servers

You can migrate a job manager profile and its registered set of servers from Version 7.0 or above to Version 8.5.

Before you begin

Review the migration planning information. See Knowledge Collection: Migration planning for WebSphere Application Server.

About this task

Job manager profiles can have one or more of the following server types registered:

- Deployment manager servers
- Managed base application servers (which are also registered to an administrative agent)

Note:

1. Managed base application servers and deployment manager servers cannot accept jobs from a job manager that is of a previous version. To avoid problems, migrate your job manager profiles to Version 8.5 before you migrate managed base application servers and deployment manager servers to Version 8.5.
2. When migrating the managed base application server or managed deployment manager in a flexible management environment, the node names must be the same in Version 8.5 and previous releases.

This task outlines the steps needed to migrate the job manager and the set of registered servers.

You can migrate job manager profiles and the registered set of servers from Version 7.0 or above to Version 8.5.

Note: Ensure that your setting for the maximum number of open files is 10000 or greater. If the number of open files is too low, this can cause a variety of migration failures.

Procedure

1. Install WebSphere Application Server Version 8.5 Install WebSphere Application Server Network Deployment Version 8.5 onto the target host in a new directory. For more information, see How do I install an application serving environment?
2. Create the target job manager profile The target job manager profile is a new job manager profile that is located on the target of the job manager migration.
   a. Run the manageprofiles command with the appropriate parameters to create a new job manager profile. For example:
3. Stop the old job manager Any jobs that exist within the old job manager database will be migrated as part of the migration.
4. Run the WASPreUpgrade command from the new WebSphere Application Server install root bin directory Save the current job manager configuration to the migration backup directory. The WASPreUpgrade command does not make any changes to the old configuration.
   a. Run the WASPreUpgrade command to save the current job manager configuration information to a migration backup directory. For example:
5. Verify the console output and the WASPreUpgrade logs for success, warnings, or failure After the WASPreUpgrade command is complete, check the console output and logs for any warnings or errors. If there are errors, all errors must be fixed and then WASPreUpgrade must be run again. Also

check the warnings to see if they will impact any other migration or runtime activities on Version 8.5. If the command completed successfully, then it is not necessary to check the logs for errors or warnings.

a. Check the WASPreUpgrade console output for "Failed with errors" or "Completed with warnings".
b. Look in the following logs for warnings or errors:
   - *<migration backupdir>*/logs/WASPreMigrationSummary.log
   - WASPreUpgrade.< timestamp >.log
   - WASPreUpgrade.trace

6. Run the WASPostUpgrade command from the new WebSphere Application Server install root bin directory Use the WASPostUpgrade command to restore the previous job manager configuration that you saved in the migration backup directory.

   Note: To avoid database inconsistencies, run WASPostUpgrade immediately after completing WASPreUpgrade. As part of WASPreUpgrade, a backup of the database is created. If you restart the old job manager before running WASPostUpgrade, the database in the backup and the database in the old job manager will be out of sync.

   a. Run the WASPostUpgrade command to restore the saved job manager configuration into the new Version 8.5 administrative agent profile. For example:
7. Verify the logs for success Once the WASPostUpgrade command is completed, check the console output and logs for any warnings or errors. If there are errors, check the error codes against the Version 8.5 Information Center for more information. If the command completed successfully, then it is not necessary to check the logs for errors or warnings.
   a. Check the WASPostUpgrade console output for "Failed with errors" or "Completed with warnings".
   b. Look in the following logs for warnings or errors:
      - *<migration backupdir>*/logs/WASPostMigrationSummary.log
      - WASPostUpgrade.<target profile name>.< timestamp >.log
      - WASPostUpgrade.<target profile name>.trace
8. Start the Version 8.5 job manager Ensure that both Version 7.0 or above and Version 8.5 of the job manager are running.
   a. Change to the new Version 8.5 job manager profile bin directory.
   b. Run the startServer jobmgr command.
   c. Check the SystemOut.log file for warnings or errors.

      Note: This topic references one or more of the application server log files. As a recommended alternative, you can configure the server to use the High Performance Extensible Logging (HPEL) log and trace infrastructure instead of using `SystemOut.log` , `SystemErr.log`, `trace.log`, and `activity.log` files on distributed and IBM i systems. You can also use HPEL in conjunction with your native z/OS logging facilities. If you are using HPEL, you can access all of your log and trace information using the LogViewer command-line tool from your server profile bin directory. See the information about using HPEL to troubleshoot applications for more information on using HPEL.
9. Migrate the registered servers The Version 8.5 job manager can manage Version 7.0 or above registered servers. For the Version 7.0 or above topology to function with the Version 8.5 job manager, you are not required to migrate the registered servers. For each registered server that you plan to migrate to Version 8.5, perform the following steps:
   - If the registered server you wish to migrate is of type deployment manager, see Scenario 1: Migrating a cell using the command line tools
   - If the registered server you wish to migrate is of type managed base application server, see Flexible Management Scenario 1: Migrating an administrative agent profile and its registered set of managed base application servers
10. Troubleshooting See the information center documentation on troubleshooting migration.

Results

You have migrated a job manager profile and its associated managed base application servers from WebSphere Application Server Version 7.0 or above to Version 8.5 using the migration tools.

Chapter 8. Migrating Compute Grid or Feature Pack for Modern Batch on distributed operating systems

This section documents the migration of WebSphere Extended Deployment Compute Grid Version 6.1.1 on WebSphere Application Server Version 6.x, 7.x or 8.0 to WebSphere Application Server Version 8.5. You can also use these steps to migrate the Feature Pack for Modern Batch 1.0.0.x.

Migration overview

Migrate WebSphere Extended Deployment Compute Grid Version 6.1.1 and above on WebSphere Application Server Version 6.x, 7.x or 8.0 to WebSphere Application Server Version 8.5. You can also use these steps to migrate the Feature Pack for Modern Batch 1.0.0.x.

Before you begin

To migrate WebSphere Extended Deployment Compute Grid Version 6.1.1 and above to Version 8.5, you must have WebSphere Extended Deployment Compute Grid Fixpack 3 and cumulative interim fix PM39203 installed. Migration from any version of the Feature Pack for Modern Batch 1.0.0.x is supported.

About this task

The following list is an overview of the steps required to migrate from WebSphere Extended Deployment Compute Grid Version 6.1.1 and above or the Feature Pack for Modern Batch Version 1.0.0.x to Version 8.5. Depending on your specific configuration, you might not follow the migration steps in the order of this overview.

If your migration involves WebSphere Application Server Version 8.5, see Special considerations for migrating to WebSphere Application Server Version 8.5.

Procedure

- Migrate the nodes. Migrate the nodes in the following order:
  1. Migrate the deployment manager.

     Note: You must always migrate the deployment manager first.
  2. Migrate the databases.
  3. Migrate the schedulers one at a time.
  4. Migrate the endpoints one at a time.
- Migrate your environment.

  Perform the following steps to migrate the deployment manager:
  1. Run the backup script.
  2. Stop the deployment manager.
  3. Unaugment WebSphere Extended Deployment Compute Grid Version 6.1.1.3 and above or the Feature Pack for Modern Batch Version 1.0.0.x.
  4. Uninstall WebSphere Extended Deployment Compute Grid Version 6.1.1.3 and above or the Feature Pack for Modern Batch Version 1.0.0.x.
  5. Install WebSphere Application Server Version 8.5.
  6. Augment the deployment manager using the deployment manager profile.
  7. Start the deployment manager.
  8. Run the restore script.

  Perform the following steps to migrate the scheduler and endpoint nodes:

1. Stop the server and node.
2. Unaugment the profile.
3. Uninstall WebSphere Extended Deployment Compute Grid Version 6.1.1.3 and above or the Feature Pack for Modern Batch Version 1.0.0.x.
4. Install WebSphere Application Server Version 8.5.
5. Augment the scheduler node.
6. Run the restore script.
7. Start the server.

Preparing cells for migration to Version 8.5

Before migrating to Version 8.5, you must perform prerequisite steps to prepare Feature Pack for Modern Batch or WebSphere Extended Deployment Compute Grid Version 6.1.1 cells for migration to WebSphere Application Server Version 8.5 cells.

Before you begin

To Migrate WebSphere Extended Deployment Compute Grid Version 6.1.1 to Version 8.5, you must have WebSphere Extended Deployment Compute Grid Fixpack 3 and cumulative interim fix PM39203 installed. Migration from any version of the Feature Pack for Modern Batch 1.0.0.x is supported.

About this task

Use the following steps to prepare your batch Version 6.1.1 cells for migration to WebSphere Application Server Version 8.5 cells.

Procedure

1. Run the prerequisite wsadmin script.
   a. On the Deployment Manager node, run the addCGSystemAppVariables.py script; for example:
      ```
      <WASHOME>/bin>wsadmin.sh –lang jython –f addCGSystemAppVariables.py
      ```
      Note: [-level wcg/wcg8/was8/fep/ default is wcg]
      - wcg if you are migrating from Compute Grid 6.1
      - wcg8 if you are migrating from Compute Grid 8.0
      - was8 if you are migrating from WebSphere Application Server 8.0
      - fep if you are migrating a Feature Pack for Modern Batch node

   b. Restart all the servers after running the script.
2. Back up the existing Version 6.1.1.3 configuration. From the <WASHOME>/bin directory of the deployment manager, run the ./wsadmin.[sh|bat] –lang jython –f migrateWCGConfigTo8.py --backup script.

   Optional parameters:
   - -configBackupDir: Use this optional parameter to change the location where the product configuration is backed up. The default is `<WASHOME>/temp`.
   - -profileName: Use this optional parameter to change the Deployment Manager profile name. The default is `Dmgr01`.

Migrating the deployment manager

After you have prepared your cells for migration and ensured that you have met all prerequisites, migrate the deployment manager.

Before you begin

To Migrate WebSphere Extended Deployment Compute Grid Version 6.1.1 to Version 8.5, you must have WebSphere Extended Deployment Compute Grid Fixpack 3 and cumulative interim fix PM39203 installed. Migration from any version of the Feature Pack for Modern Batch 1.0.0.x is supported.

About this task

You must migrate the deployment manager first. To migrate the deployment manager, perform the following steps:

Procedure

1. Stop the deployment manager. Run the <WAS_INSTALL_ROOT>/profiles/<profileName>/bin/stopManager.sh command.
2. Install WebSphere Application Server Version 8.5.
3. Migrate your old Deployment Manager. Use the migration tools to migrate the old deployment manager to WebSphere Application Server Version 8.5.
   a. Perform --wasmigrate on your migrated deployment manager from its WAS_HOME/bin dir. For example:

   ```
   wsadmin.sh -conntype NONE -lang jython -f .migrateConfigTo85.py --wasmigrate -oldWASHome <oldWASHome> -newWASHome <newWASHome>
   -cellName <cell> -nameOfProfile <profile> -configBackupDir <PathToBackupLocation> -level <level>
   ```
4. Restore your saved configurations on the deployment manager. To restore a configuration, run the following script from the <WASHOME>/bin directory of the deployment manager:

   Optional parameters:
   - -configBackupDir : Use this parameter to change the location where the batch configuration is backed up. The default is `<WASHOME>/temp`
   - - lreeJNDIName: Use this parameter to change the default JNDI name used by the batch endpoints. The default is `jdbc/lree`
   - -lreeSchema: Use this parameter to change the default database schema used by the batch endpoints. The default is `LREESCHEMA`
   - -profileName: Use this parameter to change the default profile name. The default is `Dmgr01`

Migrating databases

You can migrate your WebSphere Extended Deployment Compute Grid or Feature Pack for Modern Batch databases to Version 8.5.

Before you begin

- To Migrate WebSphere Extended Deployment Compute Grid Version 6.1.1 to Version 8.5, you must have WebSphere Extended Deployment Compute Grid Fixpack 3 and cumulative interim fix PK35389 installed.
- You can also use these steps to migrate the Feature Pack for Modern Batch 1.0.0.x.
- Migrate the deployment manager before performing the following tasks.

About this task

To migrate the databases, run the migration DDL corresponding to the database type.

Procedure

- WebSphere Extended Deployment Compute Grid
  1. For DB2: Run the following commands:
     – db2 –tvf MigrateLRSCHEDTablesDB2v6TOv85.ddl

- db2 –tvf MigrateLREETablesDB2.ddl

2. For Derby: Run the following commands:
   - >java -Djava.ext.dirs=<WAS_INSTALL_ROOT> /derby/lib -Dij.protocol=jdbc:derby: org.apache.derby.tools.ij MigrateLRSCHEDTablesDerbyv6Tov85.ddl
   - >java -Djava.ext.dirs=<WAS_INSTALL_ROOT> /derby/lib -Dij.protocol=jdbc:derby: org.apache.derby.tools.ij MigrateLREETablesDerby.ddl
3. For Oracle: Run the following scripts:
   - MigrateLRSCHEDTablesOraclev6TOv85.ddl
   - MigrateLREETablesOracle.ddl

- Feature Pack for Modern Batch
  1. For DB2: Run the following command:
     - db2 –tvf MigrateLRSCHEDTablesDB2FePTOv85.ddl
  2. For Derby: Run the following command:
     - >java -Djava.ext.dirs=<WAS_INSTALL_ROOT> /derby/lib -Dij.protocol=jdbc:derby: org.apache.derby.tools.ij MigrateLRSCHEDTablesDerbyFePTov85.ddl
  3. For Oracle: Run the following script:
     - MigrateLRSCHEDTablesOraclevFePTOv85.ddl

Migrating servers

You can migrate your WebSphere Extended Deployment Compute Grid or Feature Pack for Modern Batch servers to Version 8.5.

Before you begin

To Migrate WebSphere Extended Deployment Compute Grid Version 6.1.1.3 to Version 8.5, you must have WebSphere Extended Deployment Compute Grid Fixpack 3 and cumulative interim fix PM39203 installed.

About this task

Migrate the schedulers one at a time and then migrate the endpoints. Repeat the following steps for each Compute Grid or batch server:

Procedure

1. Stop the server. You can stop the server either by using the administration console or by running the following commands from the <USER_INSTALL_ROOT>/bin directory:
2. Install WebSphere Application Server Version 8.5.
3. Migrate your old server. Use the migration tools to migrate the old server to WebSphere Application Server Version 8.5.
4. Augment the deployment manager with the Compute Grid or batch deployment manager profile. To augment a profile, use the manageprofiles utility. In the <WAS_INSTALL_ROOT>/bin directory, run the following commands:
5. Restore your saved configurations. Start the node by running the <USER_INSTALL_ROOT>/bin/ startNode.bat command. To restore a configuration, run the following script from the <WASHOME>/bin directory:

   Required parameter:

   - -node: The name of the WebSphere Application Server node of the current server.
6. Start the server. Run the following command from the <USER_INSTALL_ROOT> directory:

Chapter 9. Migrating web server configurations

You can migrate a web server so that it supports the latest version of WebSphere Application Server.

Before you begin

IBM i

Procedure

1. IBM i Configure an HTTP server instance.

   Read the "Configuring an HTTP server instance" article in the information center for more information.

   There are two options from which to choose:

   - Create a new HTTP server instance to be used by the WebSphere Application Server Version 8.5 profile.

     This method allows WebSphere Application Server Version 6.1 or above and Version 8.5 profiles to continue operating correctly.
   - Update the HTTP server instance configuration for the WebSphere Application Server Version 6.1 or above profile that is being migrated.

     This method changes the HTTP instance configuration to work with the WebSphere Application Server Version 8.5 profile and makes the WebSphere Application Server Version 6.1 or above profile no longer usable.
2. IBM i Configure the virtual host for the WebSphere Application Server Version 8.5 profile.

   Read the "Configuring virtual hosts" article in the information center for more information.

   This step ensures that both the host and HTTP transport port number exist in the virtual host list.

   If you created a new HTTP server in the previous step or if you used the -portBlock parameter when performing the migration, the virtual host will not contain the correct port for communication with your HTTP server. You need to add a host alias for the port used by your HTTP server.
3. IBM i Configure communication with web servers.

   Read the "Communicating with web servers" article in the information center for more information.

   This step regenerates the plug-in configuration file, plugin-cfg.xml. It needs to be done after any configuration changes have been made.

   Additional configuration is required if Secure Sockets Layer (SSL) is enabled on a plug-in transport. In addition to copying the .kdb file to the Version 8.5 profile, you must edit the plug-in to specify the .kdb file required for the plug-in to use the transport.

   For more information on copying the .kbd files to the Version 8.5 profile, read the section on J2EE security in “Configuration mapping during product-configuration migration” on page 22.

IBM i

What to do next

Migrating from WebSphere Application Server Version 6.1 or above:

- Only web servers defined on managed nodes are migrated to WebSphere Application Server Version 8.5.
- If you are migrating a web server and plug-ins from WebSphere Application Server Version 6.1 or above to Version 8.5 and the web server is defined on an unmanaged node, the web server creation and application mapping must be done manually.

  To create the web server definition manually, perform one of the following actions:

  – Use the administration console wizard.

To generate mapping to all applications that are installed at web server creation, use the mapping `ALL` option in the wizard.

– Use the **wsadmin** command.

```
$AdminTask createWebServer -interactive
```

and reply `ALL` to the mapping applications prompt.

– Use the `configureWebserverDefinition.jacl` script.

This script maps all installed applications to the web server. The script updates all of the information related to the web server plug-in such as the locations of the plug-in installation root, log file, configuration file, and key stores on the web server system. However, the script does not update other properties related to the web server if the web server definition already exists.

– The Web server might have migrated with the application server. After migrating, Verify that your web server is functioning correctly. If not, delete the web server definition and follow the instructions in this topic to redefine it.

Chapter 10. Migrating administrative scripts

You can migrate administrative scripts using scripting and the wsadmin tool.

Before you begin

About this task

WebSphere Application Server Version 8.5 supports migrating administrative scripts from Version 6.1 or above.

Procedure

If you are migrating administrative scripts from Version 6.1 or above, see:
"Migrating administrative scripts from Version 6.1 or above to 8.5" on page 91

Migrating administrative scripts from a previously Version 5.1.x application server

There are some changes you should be aware of when migrating from a version of WebSphere Application Server that was previously Version 5.1.x.

Before you begin

About this task

There are a few changes to be aware of that are required for your existing scripts when moving to WebSphere Application Server Version 8.5. In general, the administration model has changed very little. However, there are some changes required when moving from a previously Version 5.1.x to Version 8.5.

Procedure

- IBM i Be aware of the implications of migrating JMS applications from the embedded messaging in WebSphere Application Server Version 5.1.x to the default messaging provider in WebSphere Application Server Version 8.5.
- IBM i A new version of Jacl (1.3.2) was shipped beginning with WebSphere Application Server Version 6.1. With this Jacl version, **regexp** command supports only Tcl 8.0 **regexp** command syntax. If your existing Version 5.1.x Jacl script uses the **regexp** command syntax that is supported in Jacl 1.2.6 but not anymore in Jacl 1.3.2, you might not get a match anymore or you might 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
       while executing
  "regexp {(?x)
       ..."
       ("if" test expression)
       invoked from within
  "if {[regexp {(?x)
       ..."
       (file testregexp.jacl line 2)
       (file line 2)
       invoked from within
  "source testregexp.jacl"
  ```

  There is no workaround for this regression problem. Jacl has indicated that this is by design and there is no simple patch to fix this design decision.

- IBM i For WSADMIN $AdminConfig: The PME CacheInstanceService type is no longer used. If your scripts contain code to set the **enable** attribute on the CacheInstanceService type, remove the code. It is not needed in Version 8.5.
- There are a few changes to be aware of that are required for your existing Version 5.1.x scripts when moving to WebSphere Application Server Version 8.5. These types of changes can be evolved directly without the assistance of script compatibility support. The data can be accessed from multiple locations, including the old and new locations. As long as the new location is not updated, the data is accessed from the old location. Once the new location is updated, it becomes the current data and is used for further accesses and updates. Warning messages are logged when the old location is still being used.

  Read “Example: Migrating - Changing transaction log directory using wsadmin scripting” on page 89 for more information.
- IBM i There are a few changes to be aware of that are required for your existing scripts when moving from Version 5.1.x to WebSphere Application Server Version 8.5. These changes are assisted by the compatibility mode provided by “WASPostUpgrade command” on page 37. During migration, the default is to migrate using compatibility mode. If this option is taken, then the old object types are migrated into the new configuration; all existing scripts will run unchanged.

  See the “Example: Migrating - Changing process definitions using scripting” on page 90 article for more information.
- Be aware of removed features that might have an impact on administration scripts.

  These might include the following:
  - 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
  - Support for 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

    Read “Migrating from the WebSphere Connect JDBC driver” on page 58 for information on using the WebSphereConnectJDBCDriverConversion command to convert data sources from the WebSphere Connect JDBC driver to the DataDirect Connect JDBC driver or the Microsoft SQL Server JDBC driver.

  For more information, see the "Deprecated and removed features" article in the information center.

Example: Migrating - Allowing configuration overwrite when saving configurations

These examples demonstrate how to enable configuration overwrite in network deployment for WebSphere Application Server Version 6.x.

Use the following examples:

- wsadmin Version 6.x
  1. Enable configuration repository to allow configuration overwrite:

     depfeat: Using Jacl:

```
set s1AdminService [$AdminConfig getid /Server:dmgr/AdminService:/]

set configRepository [$AdminConfig showAttribute $s1AdminService configRepository]
set props [$AdminConfig showAttribute $configRepository properties]
set foundAllowConfigOverwrites ""
if {$props != "{}"} {
 foreach prop $props {
  if {[$AdminConfig showAttribute $prop name] == "allowConfigOverwrites"} {
   set foundAllowConfigOverwrites $prop
   break
```

```
  }
 }
}

if {$foundAllowConfigOverwrites == ""} {
 $AdminConfig create Property $configRepository {{name allowConfigOverwrites} {value true}}
} else {
 $AdminConfig modify $foundAllowConfigOverwrites {{value true}}
}

$AdminConfig save
```

Using Jython:

```
s1AdminService = AdminConfig.getid('/Server:dmgr/AdminService:/')
configRepository = AdminConfig.showAttribute(s1AdminService, 'configRepository')
props = AdminConfig.showAttribute(configRepository, 'properties')
foundAllowConfigOverwrites = ''
if props != '[]':
 properties = props[1:len(props)-1].split(' ')
 for prop in properties:
  name = AdminConfig.showAttribute(prop, 'name')
  if name == 'allowConfigOverwrites':
   foundAllowConfigOverwrites = prop
   break

if len(foundAllowConfigOverwrites) != 0:
 AdminConfig.modify(foundAllowConfigOverwrites, [['value', 'true']])
else:
 AdminConfig.create('Property', configRepository, [['name', 'allowConfigOverwrites'], ['value', 'true']])

AdminConfig.save()
```

2. Restart the deployment manager. From the `bin` directory of the deployment manager profile, run the following:
3. Allow configuration overwrite, for example:

   depfeat: Using Jacl:

   ```
   $AdminConfig setSaveMode overwriteOnConflict
   ```

   Using Jython:

   ```
   AdminConfig.setSaveMode('overwriteOnConflict')
   ```

Example: Migrating - Changing transaction log directory using wsadmin scripting

Prepare for evolutionary changes without script compatibility support.

The location of the `transaction logs directory` attribute has changed from the `ApplicationServer::TransactionService` to the `ServerEntry::recoveryLogs`. As long as the new location is not used, the value from the old location will continue to be used. Scripts that modify the old location can still be used; that value will take effect until a value in the new location is set. The change to scripts to use the new location is as follows:

Old location:

- Using Jacl:

  ```
  set transService [$AdminConfig list TransactionService $server1]
  $AdminConfig showAttribute $transService transactionLogDirectory
  ```

New Location:

- Using Jython:

  ```
  AdminConfig.list("ServerEntry")

  # Select one entry from the list, e.g the entry for server1:
  ```

```
serverEntryId = AdminConfig.getid("/ServerEntry:server1")
serverEntry = AdminConfig.list("ServerEntry", serverEntryId)

recoveryLog = AdminConfig.showAttribute(serverEntry, "recoveryLog")
AdminConfig.showAttribute(recoveryLog, "transactionLogDirectory")
```

- Using Jacl:

```
$AdminConfig list ServerEntry $node
set serverEntry <select one of the ServerEntry from output of above command>
set recoveryLog [$AdminConfig showAttribute $serverEntry recoveryLog]
$AdminConfig showAttribute $recoveryLog transactionLogDirectory
```

Example: Migrating - Changing process definitions using scripting

Prepare for evolutionary changes with script compatibility support.

The following changes can be made with **with** script compatibility support.

- **HTTP transports:** the architecture uses the new channel framework. HTTP definitions are mapped on top of this support. When compatibility mode is chosen, the old HTTPTransport objects are migrated and mapped onto the channel architecture. Existing scripts can modify these objects and will run unchanged.
- IBM i **Process definition:** The name of this object is changed from processDef to processDefs. You can mitigate this change by using the compatibility mode mapping provided by the migration tools. The change to scripts to use the new location is as follows:
  - Old example using Jython:

    ```
    processDef = AdminConfig.list('JavaProcessDef', server1)
    print processDef
    ```

  - New example using Jython. Identify the process definition belonging to this server and assign it to the processDefs variable:

    ```
    processDefs = AdminConfig.list('JavaProcessDef', server1)
    print processDefs
    ```

Example: Migrating - Modifying web container port numbers

These examples demonstrate how to modify web container HTTP transport ports for WebSphere Application Server Version 6.x.

Use the following examples:

- wsadmin Version 5.1.x

  depfeat: Using Jacl:

  ```
  set httpPort  7575
  set server [$AdminConfig getid /Cell:myCell/Node:myNode/Server:server1/]
  set transports [$AdminConfig list HTTPTransport $server]
  set transport [lindex $transports 0]
  set endPoint [$AdminConfig showAttribute $transport address]
  $AdminConfig modify $endPoint [list [list port $httpPort]]
  $AdminConfig save
  ```

  Using Jython:

  ```
  httpPort = 7575
  server = AdminConfig.getid("/Cell:myCell/Node:myNode/Server:server1/")
  transports = AdminConfig.list("HTTPTransport", server).split(java.lang.System.getProperty("line.separator"))
  transport = transports[0]
  endPoint = AdminConfig.showAttribute(transport, "address")
  AdminConfig.modify(endPoint, [["port", httpPort]])
  AdminConfig.save()
  ```

- wsadmin Version 6.x

  depfeat: Using Jacl:

  ```
  set serverNm server1
  set newPort  7575
  set node [$AdminConfig getid /Cell:myCell/Node:myNode/]
  set TCS [$AdminConfig getid /Cell:myCell/Node:myNode/Server:server1/TransportChannelService:/]
  set chains [$AdminTask listChains $TCS {-acceptorFilter WebContainerInboundChannel}]
  ```

```
foreach chain $chains {
   set channels [lindex [$AdminConfig showAttribute $chain transportChannels] 0]
   foreach channel $channels {
      if {[catch {set channelEndPointName [$AdminConfig showAttribute $channel endPointName]} result]} {
         # ignore the error as not all channel has endPointName attribute
      } else {
         set serverEntries [$AdminConfig list ServerEntry $node]
         foreach serverEntry $serverEntries {
            set sName [$AdminConfig showAttribute $serverEntry serverName]
            if {$sName == $serverNm} {
               set specialEndPoints [lindex [$AdminConfig showAttribute $serverEntry specialEndpoints] 0]
               foreach specialEndPoint $specialEndPoints {
                  set endPointNm [$AdminConfig showAttribute $specialEndPoint endPointName]
                  if {$endPointNm == $channelEndPointName} {
                     set ePoint [$AdminConfig showAttribute $specialEndPoint endPoint]
                     $AdminConfig modify $ePoint [list [list port $newPort]]
                     break
                  }
               }
            }
         }
      }
   }
}

$AdminConfig save
```

Using Jython:

```
serverNm = "server1"
newPort = "7575"
node = AdminConfig.getid("/Cell:myCell/Node:myNode/")
TCS = AdminConfig.getid("/Cell:myCell/Node:myNode/Server:server1/TransportChannelService:/")
chains = AdminTask.listChains(TCS, "[-acceptorFilter WebContainerInboundChannel]").split(java.lang.System.getProperty("line.separator"))

for chain in chains:
   channelString = AdminConfig.showAttribute(chain, "transportChannels")
   channelList = channelString[1:len(channelString)-1].split(" ")
   for channel in channelList:
      try:
         channelEndPointName = AdminConfig.showAttribute(channel, "endPointName")
         serverEntries = AdminConfig.list("ServerEntry", node).split(java.lang.System.getProperty("line.separator"))
         for serverEntry in serverEntries:
            sName = AdminConfig.showAttribute(serverEntry, "serverName")
            if sName == serverNm:
               sepString = AdminConfig.showAttribute(serverEntry, "specialEndpoints")
               sepList = sepString[1:len(sepString)-1].split(" ")
               for specialEndPoint in sepList:
                  endPointNm = AdminConfig.showAttribute(specialEndPoint, "endPointName")
                  if endPointNm == channelEndPointName:
                     ePoint = AdminConfig.showAttribute(specialEndPoint, "endPoint")
                     AdminConfig.modify(ePoint, [["port", newPort]])
                     break
      except:
         # ignore the error as not all channel has endPointName attribute
         pass

AdminConfig.save()
```

Migrating administrative scripts from Version 6.1 or above to 8.5

There are some changes you should be aware of when migrating your administrative scripts from WebSphere Application Server Version 6.1 or above to Version 8.5.

Before you begin

Procedure

- Be aware of removed features that might have an impact on administration scripts.

  These might include the following:
  - 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
  - Support for 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

Read “Migrating from the WebSphere Connect JDBC driver” on page 58 for information on using the WebSphereConnectJDBCDriverConversion command to convert data sources from the WebSphere Connect JDBC driver to the DataDirect Connect JDBC driver or the Microsoft SQL Server 2005 JDBC driver.

For more information, see the "Deprecated and removed features" article in the information center.

- Be aware of the change required when creating Service Integration Bus (SIB) objects.

  For more information about the **createSIBus** command, see the "createSIBus command" article in the information center.

Updating SSL configurations to Version 8.5 configuration definitions after migration

When migrating to Version 8.5, you can update the format for SSL configuration or you can continue to use the format of the earlier version. If you encounter errors with your existing administration scripts for SSL configurations, use this task to manually convert your SSL configuration to the Version 8.5 format.

Before you begin

Note:

About this task

IBM i When migrating to Version 8.5, you can use the “WASPreUpgrade command” on page 34 to save the configuration of your previously installed version into a migration-specific backup directory. When migration is complete, you can use the “WASPostUpgrade command” on page 37 to retrieve the saved configuration and WASPostUpgrade script to migrate your previous configuration. The **-scriptCompatibility** parameter for the WASPostUpgrade command is used to specify whether to maintain the 6.1 or above configuration definitions or to upgrade the format to Version 8.5 configuration definitions. If you used the default value, or `-scriptCompatibility true` when migrating, you do not need to perform this task. If you set the scriptCompatibility parameter to `false` during migration, you may notice that your existing administration scripts for SSL configurations do not work correctly. If this occurs, use this task to convert your 6.1 or above SSL configuration definitions to Version 8.5 This process creates a new SSL configuration based on the existing configuration.

Follow the steps below to modify the existing SSL configuration:

```
<repertoire xmi:id="SSLConfig_1" alias="Node02/DefaultSSLSettings">
<setting xmi:id="SecureSocketLayer_1" keyFileName="$install_root/etc/MyServerKeyFile.jks"
keyFilePassword="password" keyFileFormat="JKS" trustFileName="$install_root/etc/MyServerTrustFile.jks"
trustFilePassword="password" trustFileFormat="JKS" clientAuthentication="false" securityLevel="HIGH"
enableCryptoHardwareSupport="false">
<cryptoHardware xmi:id="CryptoHardwareToken_1" tokenType="" libraryFile="" password="{custom}"/>
<properties xmi:id="Property_6" name="com.ibm.ssl.protocol" value="SSL"/>
<properties xmi:id="Property_7" name="com.ibm.ssl.contextProvider" value="IBMJSSE2"/>
</setting>
</repertoire>
```

Procedure

1. Create a key store that references the key store attributes in the old configuration.
   a. In the existing configuration, find the **keyFileName**, **keyFilePassword**, and **keyFileFormat** attributes.

      ```
      keyFileName="${install_root}/etc/MyServerKeyFile.jks" keyFilePassword="password" keyFileFormat="JKS"
      ```

   b. Use the **keyFileName**, **keyFilePassword**, and **keyFileFormat** attributes to create a new KeyStore object. For this example, set the name as "DefaultSSLSettings_KeyStore".

      depfeat: Using Jacl:

```
$AdminTask createKeyStore {-keyStoreName DefaultSSLSettings_KeyStore -keyStoreLocation
${install_root}/etc/MyServerKeyFile.jks -keyStoreType JKS -keyStorePassword
password -keyStorePasswordVerify password }
```

The resulting configuration object in the `security.xml` file is:

```
<keyStores xmi:id="KeyStore_1" name="DefaultSSLSettings_KeyStore" password="password"
provider="IBMJCE" location="$install_root/etc/MyServerKeyFile.jks" type="JKS" fileBased="true"
managementScope="ManagementScope_1"/>
```

Note: If you specify the cryptoHardware values in your configuration, create the KeyStore object using these values instead. Associate the -keyStoreLocation parameter with the libraryFile attribute, the -keyStoreType parameter with the tokenType attribute, and the -keyStorePassword parameter with the password attribute.

```
<cryptoHardware xmi:id="CryptoHardwareToken_1" tokenType="" libraryFile="" password=""/>
```

2. Create a trust store that references the trust store attributes from the existing configuration.
   a. Find the **trustFileName**, **trustFilePassword**, and **trustFileFormat** attributes in the existing configuration.

      ```
      trustFileName="$install_root/etc/MyServerTrustFile.jks" trustFilePassword="password"
      trustFileFormat="JKS"
      ```

   b. Use the **trustFileName**, **trustFilePassword**, and **trustFileFormat** attributes to create a new KeyStore object. For this example, set the name as "DefaultSSLSettings_TrustStore".

      depfeat: Using Jacl:

      ```
      $AdminTask createKeyStore {-keyStoreName DefaultSSLSettings_TrustStore -keyStoreLocation
      $install_root/etc/MyServerTrustFile.jks -keyStoreType JKS -keyStorePassword password
      -keyStorePasswordVerify password }
      ```

      The resulting configuration object in the `security.xml` file is:

      ```
      <keyStores xmi:id="KeyStore_2" name="DefaultSSLSettings_TrustStore" password="password"
      provider="IBMJCE" location="$install_root/etc/MyServerTrustFile.jks" type="JKS" fileBased="true"
      managementScope="ManagementScope_1"/>
      ```

3. Create a new SSL configuration using the new key store and trust store. Include any other attributes from the existing configuration which are still valid.

   Use a new alias for your updated SSL configuration. You can not create an SSL configuration with the same name as your existing configuration.

   depfeat: Using Jacl:

   ```
   $AdminTask createSSLConfig {-alias DefaultSSLSettings -trustStoreName DefaultSSLSettings_TrustStore
    -keyStoreName DefaultSSLSettings_KeyStore -keyManagerName IbmX509 -trustManagerName IbmX509
   -clientAuthentication true -securityLevel HIGH -jsseProvider IBMJSSE2 -sslProtocol SSL  }
   ```

Results

The new SSL configuration is:

```
<repertoire xmi:id="SSLConfig_1" alias="DefaultSSLSettings" managementScope="ManagementScope_1">
<setting xmi:id="SecureSocketLayer_1" clientAuthentication="true" securityLevel="HIGH" enabledCiphers=""
jsseProvider="IBMJSSE2" sslProtocol="SSL" keyStore="KeyStore_1" trustStore="KeyStore_2"
trustManager="TrustManager_1" keyManager="KeyManager_1"/>
</repertoire>
```

Note: The default management scope is used if it is not specified.

Chapter 11. Running multiple application server versions

You can create an environment in which multiple versions of WebSphere Application Server can run independently on the same system at the same time. A major consideration in coexistence is the avoidance of port conflicts.

Before you begin

IBM i Coexisting, as it applies to WebSphere Application Server products, is running a new release of a WebSphere Application Server product on the same machine at the same time as you run an earlier release or running two installations of the same release of a WebSphere Application Server product on the same machine at the same time.

Procedure

Read “Coexistence support.”
This article discusses which coexistence scenarios are supported.

Coexistence support

Coexistence is a state in which multiple installations and multiple nodes from different versions of WebSphere Application Server run independently in the same environment at the same time.

IBM i As it applies to WebSphere Application Server products, coexistence primarily refers to the ability of multiple installations of WebSphere Application Server to run independently on the same machine at the same time. Multiple installations include multiple versions and multiple instances of one version. Coexistence also implies various combinations of web server interaction.

IBM i WebSphere Application Server Version 8.5 products can coexist with the following supported versions:

- WebSphere Application Server Version 6.1 or above
- WebSphere Application Server, Network Deployment Version 6.1 or above

IBM i All combinations of Version 6.1 or above and Version 8.5 products can coexist so long as there are no port conflicts.

IBM i WebSphere Application Server Version 6.1 and above clients can coexist with Version 8.5 clients.

IBM i

Table 2. WebSphere Application Server Version 6.1, 7.0, 8.0, and Version 8.5 clients multiversion coexistence scenarios. The table lists the installed products and specifies if the products support WebSphere Application Server Version 8.5 clients.

Installed product	WebSphere Application Server Version 8.5 clients
WebSphere Application Server Version 6.1	Supported
WebSphere Application Server WebSphere Application Server, Network Deployment Version 6.1	Supported
WebSphere Application Server Version 7.0	Supported
WebSphere Application Server WebSphere Application Server, Network Deployment Version 7.0	Supported
WebSphere Application Server Version 8.0	Supported

Table 2. WebSphere Application Server Version 6.1, 7.0, 8.0, and Version 8.5 clients multiversion coexistence scenarios (continued). The table lists the installed products and specifies if the products support WebSphere Application Server Version 8.5 clients.

Installed product	WebSphere Application Server Version 8.5 clients
WebSphere Application Server WebSphere Application Server, Network Deployment Version 8.0	Supported

IBM i WebSphere Application ServerVersion 6.1, 7.0, and 8.0 products can coexist with Version 8.5 products.

IBM i

Table 3. WebSphere Application Server Version 6.1, 7.0, 8.0 and Version 8.5 multiversion coexistence support. The table lists the installed products and specifies if coexistence with other products and versions is supported.

Installed product	WebSphere Application Server Version 6.1, 7.0, and 8.0			WebSphere Application Server Version 8.5		
	Application Server	WebSphere Application Server, Network Deployment	WebSphere Application Server, Express	Application Server	WebSphere Application Server, Network Deployment	WebSphere Application Server, Express
WebSphere Application Server Version 6.1	Supported	Supported	Supported	Supported	Supported	Supported
WebSphere Application Server, Network Deployment Version 6.1	Supported	Supported	Supported	Supported	Supported	Supported
WebSphere Application Server Version 7.0	Supported	Supported	Supported	Supported	Supported	Supported
WebSphere Application Server, Network Deployment Version 7.0	Supported	Supported	Supported	Supported	Supported	Supported
WebSphere Application Server Version 8.0	Supported	Supported	Supported	Supported	Supported	Supported
WebSphere Application Server, Network Deployment Version 8.0	Supported	Supported	Supported	Supported	Supported	Supported

IBM i In addition to multiversion coexistence, WebSphere Application Server also lets you install multiple times on one machine (multiple installation instances) or install once and have multiple profiles. Multiple Version 8.5 installation instances on one machine include the following combinations:

- Multiple application server profiles from multiple installations of the WebSphere Application Server product
- Multiple application server profiles from a single installation of the WebSphere Application Serverr product

Chapter 12. Interoperating multiple application server versions

WebSphere Application Server Version 8.5 is interoperable with other versions of the product under certain conditions.

Before you begin

Read “Overview of migration, coexistence, and interoperability” on page 1 and “Premigration considerations” on page 6. For resources to help you plan and perform your migration, visit Knowledge Collection: Migration planning for WebSphere Application Server.

IBM i Interoperability applies to WebSphere administrative actions, administrative communication and to application client requests. WebSphere Application Server Version 8.5 is generally interoperable with Version 6.1 and above. However, there are specific requirements to address for each version. In general, you should be at the most recent group PTF level to support interoperability.

You can upgrade a portion of the nodes in a cell to WebSphere Application Server Version 8.5 while leaving others at the older release level. This means that, for a period of time, you might be administering servers that are at the current release and servers that are running the newer release in the same cell.

Restrictions on using mixed-release cells:

- A mixed-release environment where some members are at an older release level might have some restrictions. For details, read the "Creating application servers" article in the information center.

For web application clients, interoperability is provided by employing the HTTP server plug-in at the highest version of all of the application servers in the cell. For example, a cell comprised of WebSphere Application Server Version 7.0 and Version 6.1 application servers would use a supported HTTP server with a Version 7.0 plug-in. Refer to the TechnoteWeb server plug-in policy for WebSphere Application Server for more information.

Procedure

Upgrade the Software Development Kit (SDK) used to one supported by Version 8.5.
Read Recommended fixes for WebSphere Application Server for more information.

What to do next

This information is dynamic and might be augmented by information in technical articles that are available on the IBM DeveloperWorks WebSphere site. Check the site for the latest information.

Chapter 13. Configuring port settings

When you configure WebSphere Application Server resources or assign port numbers to other applications, you must avoid conflicts with other assigned ports. In addition, you must explicitly enable access to particular port numbers when you configure a firewall.

Before you begin

IBM i For more information about port numbers that your IBM i system currently uses, enter the NETSTAT *CNN command on the command line. Press **F14** to view assigned port numbers.

IBM i You can also use the port validator tool to find port conflicts between different WebSphere Application Server profiles, products, and servers. Read the *Port validator tool* article in the information center for more information.

Tip: Port conflicts might occur if you install WebSphere Application Server on multiple systems with deployment managers managing servers or clusters on different systems. The configuration-service port-resolution mechanism does not support cross profiles on different host machines.

- **Example 1:**
  1. On system A, create a cell profile that includes Dmgr and AppSrv01 (Node1).
  2. On system B, create AppSrv01 and federate AppSrv01 (Node2) to Dmgr on system A.
  3. Create server1 on Node1 and server2 on Node2.
  4. The server1 server and server2 server might contain duplicate server endpoint ports in the `serverindex.xml` file because Node1 and Node2 are located on different host systems.
- **Example 2:**
  1. On system A, create a cell profile that includes Dmgr and AppSrv01 (Node1).
  2. On system B, create AppSrv01 and federate AppSrv01 (Node2) to Dmgr on system A.
  3. On system B, create JobManager.
  4. Create a cluster and add two servers, server1 on Node1 and server2 on Node2.
  5. The server2 server and the JobManager server might contain duplicate server endpoint ports in the `serverindex.xml` file because server2 and JobManager are in cross profiles. The server2 server is under Dmgr, JobManager is under the JobManager profile. and the Dmgr and JobManager profiles are located on different machines.

Procedure

1. Review the port number settings, especially when you are planning to coexist.

   IBM i You can use the **dspwasinst** command-line tool to display the port information for a profile. Read the *dspwasinst command* article in the information center for more information.

   IBM i You can use the **dspwasinst** command-line tool to display the port information for a profile. See the information center.

2. Optional: Change the port number settings.

   IBM i You can set port numbers when configuring the product after installation.

   - During profile creation using the **manageprofiles** command, you can accept the default port values or you can specify your port settings. If you want to specify ports, you can do so in any of the following ways:
     – Specify the use of a port file that contains the port values.
     – Specify the use of a starting port value.
     – Specify the use of the default port values.

     Read the *manageprofiles command* article in the information center for more information.

- IBM i You can use the **chgwassvr** command to change the ports for an application server within a profile.

  Read the *chgwassvr command* article in the information center for more information.

Port number settings

You should be able to identify the default port numbers used in the various versions of WebSphere Application Server so that you can avoid port conflicts if you plan for an earlier version to coexist or interoperate with Version 8.5.

When you configure WebSphere Application Server resources or assign port numbers to other applications, you must avoid conflicts with other assigned ports. In addition, when you configure a firewall, you must explicitly enable access to particular port numbers.

If ports are already defined in a configuration being migrated, the migration tools fix the port conflicts in the Version 8.5 configuration and log the changes for your verification .

Notes: IBM i

- When you install WebSphere Application Server, the default instance is created with the default port values. When you create a WebSphere Application Server profile with the manageprofiles script, you can specify different port values.
- For more information about port numbers that your iSeries system currently uses, enter the NETSTAT *CNN command on the CL command line. Press F14 to view assigned port numbers.
- You can also use the port validator tool to find port conflicts between different WebSphere Application Server profiles, products, and servers. Read the "port validator tool" article in the information center for more information.

Version 8.5 port numbers

Table 4. Port definitions for WebSphere Application Server Version 8.5. The table lists port names and the default values of the port numbers.

Port Name	Default Value								Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Secure Proxy Server Administrative Agent	Administrative Subsystem	
Administrative Console Port (WC_adminhost)	9060	----	9060	9060	9960	----	----	----	serverindex.xml and virtualhosts.xml
Administrative Console Secure Port (WC_adminhost_secure)	9043	----	9043	9043	9943	----	----	----	
HTTP Transport Port (WC_defaulthost)	9080	9080	----	----	----	80	----	----	
HTTPS Transport Secure Port (WC_defaulthost_secure)	9443	9443	----	----	----	443	----	----	

Table 4. Port definitions for WebSphere Application Server Version 8.5 (continued). The table lists port names and the default values of the port numbers.

Port Name	Default Value								Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Secure Proxy Server Administrative Agent	Administrative Subsystem	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2809	2809	9809	9807	9808	----	9807	----	serverindex.xml
Cell Discovery Address (CELL_ DISCOVERY_ ADDRESS)	----	----	7277	----	----	----	----	----	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9402	9405	9402	9402	9402	----	----	----	
CSIV2 Server Authentication Listener Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9403	9406	9403	9403	9403	----	----	----	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9353	9353	9352	----	----	----	----	----	
Internal JMS Server Port (JMSSERVER_ SECURITY_ PORT)	5557	----	----	----	----	----	----	----	
IPC Connector Port (IPC_ CONNECTOR_ ADDRESS)	9633	9633	9632	9630	9631	9633	9630	9634	
MQ Transport Port (SIB_ MQ_ ENDPOINT_ ADDRESS)	5558	5558	----	----	----	----	----	----	
MQ Transport Secure Port (SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS)	5578	5578	----	----	----	----	----	----	

Table 4. Port definitions for WebSphere Application Server Version 8.5 (continued). The table lists port names and the default values of the port numbers.

Port Name	Default Value								Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Secure Proxy Server Administrative Agent	Administrative Subsystem	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9100	9100	9100	9098	9099	----	----	----	serverindex.xml
RMI Connector Port (RMI_ CONNECTOR_ ADDRESS)	----	----	----	----	----	----	----	9810	
JSR 160 RMI Connector Port (JSR160RMI_ CONNECTOR_ ADDRESS)	----	----	----	----	----	----	----	9811	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS (Deprecated)	9401	9404	9401	9401	9401	----	----	----	
Service Integration Port (SIB_ ENDPOINT_ ADDRESS)	7276	7276	----	----	----	----	----	----	
Service Integration Secure Port (SIB_ ENDPOINT_ SECURE_ ADDRESS)	7286	7286	----	----	----	----	----	----	
SIP Container Port (SIP_ DEFAULTHOST)	5060	5060	----	----	----	5060	----	----	
SIP Container Secure Port (SIP_ DEFAULTHOST_ SECURE)	5061	5061	----	----	----	5061	----	----	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8880	8880	8879	8877	8876	----	8877	8881	
DataPower® Appliance Manager Secure Inbound Port (DataPowerMgr_ inbound_ secure)	----	----	5555	----	----	----	----	----	

Table 4. Port definitions for WebSphere Application Server Version 8.5 (continued). The table lists port names and the default values of the port numbers.

Port Name	Default Value								Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Secure Proxy Server Administrative Agent	Administrative Subsystem	
Middleware Agent RPC port (XDAGENT_PORT)	----	----	7060	----	----	----	----	----	serverindex.xml
Administration Overlay UDP Port (OVERLAY_UDP_LISTENER_ADDRESS)	11003	11003	11005	----	----	----	----	----	
Administration Overlay TCP Port (OVERLAY_TCP_LISTENER_ADDRESS)	11004	11004	11006	----	----	----	----	----	
Status Update Listener Port (STATUS_LISTENER_ADDRESS)	----	----	9420	----	9425	----	----	----	
IBM HTTP Server Port	80	----	----	----	----	----	----	----	virtualhosts.xml, plugin-cfg.xml, and *web_server_root*/conf/httpd.conf
IBM HTTPS Server Administration Port	8008	----	----	----	----	----	----	----	*web_server_root*/conf/admin.conf

When you federate an application server node into a deployment-manager cell, the deployment manager instantiates the node agent server process on the application server node. The node agent server uses these port assignments by default.

Table 5. Port definitions for the Version 8.5 node agent server process. The table lists port names and the default values of the port numbers.

Port Name	Default Value	File
	Cell Node Agent	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2810	serverindex.xml
CSIV2 Server Authentication Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9201	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9202	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9354	
IPC Connector Port (IPC_ CONNECTOR_ ADDRESS)	9626	
Node Discovery Address (NODE_ DISCOVERY_ ADDRESS)	7272	
Node IPV6 Discovery Address (NODE_ IPV6_ MULTICAST_ DISCOVERY_ ADDRESS)	5001	
Node Multicast Discovery Address (NODE_ MULTICAST_ DISCOVERY_ ADDRESS)	5000	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9900	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS (Deprecated)	9901	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8878	
Node Middleware Agent RPC port (NODE_ XDAGENT_ PORT)	7061	
Node Administration Overlay UDP Port (NODE_ OVERLAY_ UDP_ LISTENER_ ADDRESS)	11001	
Node Administration Overlay TCP Port (NODE_ OVERLAY_ TCP_ LISTENER_ ADDRESS)	11002	

Version 7.0 and Version 8.0 port numbers

Table 6. Port definitions for WebSphere Application Server Version 7.0 and Version 8.0. The table lists port names and the default values of the port numbers.

Port Name	Default Value							Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
Administrative Console Port (WC_ adminhost)	9060	----	9060	9060	9960	----	----	serverindex.xml and virtualhosts.xml
Administrative Console Secure Port (WC_ adminhost_ secure)	9043	----	9043	9043	9943	----	----	
HTTP Transport Port (WC_ defaulthost)	9080	9080	----	----	----	80	----	
HTTPS Transport Secure Port (WC_ defaulthost_ secure)	9443	9443	----	----	----	443	----	

Table 6. Port definitions for WebSphere Application Server Version 7.0 and Version 8.0 (continued). The table lists port names and the default values of the port numbers.

Port Name	Default Value							Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2809	2809	9809	9807	9808	----	----	serverindex.xml
Cell Discovery Address (CELL_ DISCOVERY_ ADDRESS)	----	----	7277	----	----	----	----	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9402	9405	9402	9402	9402	----	----	
CSIV2 Server Authentication Listener Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9403	9406	9403	9403	9403	----	----	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9353	9353	9352	----	----	----	----	
Internal JMS Server Port (JMSSERVER_ SECURITY_ PORT)	5557	----	----	----	----	----	----	
IPC Connector Port (IPC_ CONNECTOR_ ADDRESS)	9633	9633	9632	9630	9631	9633	9634	
MQ Transport Port (SIB_ MQ_ ENDPOINT_ ADDRESS)	5558	5558	----	----	----	----	----	
MQ Transport Secure Port (SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS)	5578	5578	----	----	----	----	----	

Table 6. Port definitions for WebSphere Application Server Version 7.0 and Version 8.0 (continued). The table lists port names and the default values of the port numbers.

Port Name	Default Value							Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9100	9100	9100	9098	9099	----	----	serverindex.xml
RMI Connector Port (RMI_ CONNECTOR_ ADDRESS)	----	----	----	----	----	----	9810	
JSR 160 RMI Connector Port (JSR160RMI_ CONNECTOR_ ADDRESS)	----	----	----	----	----	----	9811	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS (Deprecated)	9401	9404	9401	9401	9401	----	----	
Service Integration Port (SIB_ ENDPOINT_ ADDRESS)	7276	7276	----	----	----	----	----	
Service Integration Secure Port (SIB_ ENDPOINT_ SECURE_ ADDRESS)	7286	7286	----	----	----	----	----	
SIP Container Port (SIP_ DEFAULTHOST)	5060	5060	----	----	----	5060	----	
SIP Container Secure Port (SIP_ DEFAULTHOST_ SECURE)	5061	5061	----	----	----	5061	----	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8880	8880	8879	8877	8876	----	8881	
IBM HTTP Server Port	80	----	----	----	----	----	----	virtualhosts.xml, plugin-cfg.xml, and *web_ server_ root*/conf/ httpd.conf
IBM HTTPS Server Administration Port	8008	----	----	----	----	----	----	*web_ server_ root*/conf/ admin.conf

When you federate an application server node into a deployment-manager cell, the deployment manager instantiates the node agent server process on the application server node. The node agent server uses these port assignments by default.

Table 7. Port definitions for the Version 7.0 and Version 8.0 node agent server process. The table lists port names and the default values of the port numbers.

Port Name	Default Value	File
	Cell Node Agent	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2810	serverindex.xml
CSIV2 Server Authentication Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9201	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9202	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9354	
IPC Connector Port (IPC_ CONNECTOR_ ADDRESS)	9626	
Node Discovery Address (NODE_ DISCOVERY_ ADDRESS)	7272	
Node IPV6 Discovery Address (NODE_ IPV6_ MULTICAST_ DISCOVERY_ ADDRESS)	5001	
Node Multicast Discovery Address (NODE_ MULTICAST_ DISCOVERY_ ADDRESS)	5000	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9900	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS (Deprecated)	9901	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8878	

Chapter 14. Troubleshooting migration

You might encounter problems while migrating from an older version of WebSphere Application Server.

Before you begin

Procedure

- IBM i If you encounter a problem when you are migrating from a previous version of WebSphere Application Server to Version 8.5, check your log files and other available information.
  1. Look for the log files, and browse them for clues.
     - *migration_backup_dir*/logs/WASPreUpgrade.*time_stamp*.log
     - *migration_backup_dir*/logs/WASPostUpgrade.*time_stamp*.log
     - *app_server_root*/logs/clientupgrade.*time_stamp*.log
  2. Look for `MIGR0259I: The migration has successfully completed` or `MIGR0271W: The migration completed with warnings` in one of the log files.
     - *migration_backup_dir*/logs/WASPreUpgrade.*time_stamp*.log
     - *migration_backup_dir*/logs/WASPostUpgrade.*time_stamp*.log
     - *app_server_root*/logs/clientupgrade.*time_stamp*.log

     If `MIGR0286E: The migration failed to complete` is displayed, attempt to correct any problems based on the error messages that appear in the log file. After correcting any errors, rerun the command from the bin directory of the product installation root.
     - If you are migrating cell-scoped Cloudscape databases in the WebSphere Application Server, Network Deployment environment, you might receive the following warning message on Version 8.5 servers:

       ```
       ExtendedMessage: BBOO0221W: WSVR0022W: The value,
       {CLOUDSCAPE_JDBC_DRIVER_PATH}/db2j.jar, with Resource ID,
       DataSource_1000001, located at
       file:/WebSphere/V6R1/AppServer/profiles/default/config/cells/
       <cell_name>/nodes/<node_name>/servers/<server_name>
       /resources.xml has an invalid variable
       ```

       You can safely ignore this error until you have finished migrating all of your nodes. After you have migrated all the nodes from Version 6.0, and no longer need support for a cell-level Cloudscape database (for older release nodes), you can delete the custom property CloudscapeOldDBPath in your cell-level Derby databases that points to `${CLOUDSCAPE_JDBC_DRIVER_PATH}/db2j.jar`.
  3. Open the service log of the server that is hosting the resource that you are trying to access, and browse error and warning messages.
  4. With WebSphere Application Server running, run the **dumpNameSpace** command and pipe, redirect, or “more” the output so that it can be easily viewed.

     This command results in a display of all objects in WebSphere Application Server's namespace, including the directory path and object name.
  5. If the object that a client needs to access does not appear, use the administrative console to verify the following conditions.
     - The server hosting the target resource is started.
     - The web module or enterprise Java bean container hosting the target resource is running.
     - The JNDI name of the target resource is properly specified.
  6. Enable tracing when you run the migration tools; then, analyze the trace data or forward it to the appropriate organization for analysis.

You can enable trace when you use the **WASPreUpgrade** command or the **WASPostUpgrade** command by specifying the following parameters:

-traceString
This is an optional parameter. The value *trace_spec* specifies the trace information that you want to collect.

– Specify `"*=all=enabled"` (with quotation marks) to gather all possible trace information.

  This produces a very large trace file; for example, it can exceed 1 GB for the **WASPostUpgrade** command.

– Specify `"Migration.*=all"` to gather only the migration information

– Specify `"Migration.Flow=all:Migration.*=finer"` to gather most of the migration information.

– Specify `"Migration.Flow=finer:Migration.*=fine"` to gather the minimum amount of migration data needed by support teams.

  This is the default.

If you do not specify the -traceString or -traceFile parameter, the command creates a trace file by default and places it in the *backupDirectory*`/logs` directory.

If you specify this parameter, you must also specify the -traceFile parameter.

-traceFile
This is an optional parameter. The value *file_name* specifies the name of the output file for trace information.

If you do not specify the -traceString or -traceFile parameter, the command creates a trace file by default and places it in the *backupDirectory*`/logs` directory.

If you specify the -traceString parameter but do not specify the -traceFile parameter, the script does not generate a trace file.

See the “Tracing and logging configuration” article for information on configuring your tracing and logging settings to help diagnose problems.

For current information available from IBM Support on known problems and their resolution, read the IBM Support page. IBM Support also has documents that can save you time gathering information needed to resolve this problem. Before opening a PMR, read the IBM Support page.

- IBM i During the migration process, problems might occur while you are using the **WASPreUpgrade** tool or the **WASPostUpgrade** tool.
  – Problems can occur when you are using the **WASPreUpgrade** tool.
    - A “Not found” or “No such file or directory” message is returned.

      This problem can occur if you are trying to run the **WASPreUpgrade** tool from a directory other than the Version 8.5 *app_server_root*`\bin`. Verify that the **WASPreUpgrade** script resides in the Version 8.5 *app_server_root*`\bin` directory, and launch the file from that location.
    - The DB2 JDBC driver and DB2 JDBC driver (XA) cannot be found in the drop-down list of supported JDBC providers in the administrative console.

      The administrative console no longer displays deprecated JDBC provider names. The new JDBC provider names used in the administrative console are more descriptive and less confusing. The new providers will differ only by name from the deprecated ones.

      The deprecated names will continue to exist in the `jdbc-resource-provider-templates.xml` file for migration reasons (for existing JACL scripts for example); however, you are encouraged to use the new JDBC provider names in your JACL scripts.
    - You receive the following message:

      `MIGR0108E: The specified WebSphere directory does not contain a WebSphere version that can be upgraded.`

      The following possible reasons for this error exist:

- If WebSphere Application Server Version 6.1 is installed, you might not have run the **WASPreUpgrade** tool from the `bin` directory of the Version 8.5 installation root.
  1. Look for something like the following message to display when the **WASPreUpgrade** tool runs: `IBM WebSphere Application Server, Release 6.x.`

     This message indicates that you are running the WebSphere Application Server Version 6.1 migration utility, not the Version 8.5 migration utility.
  2. Alter your environment path or change the current directory so that you can launch the Version 8.5 **WASPreUpgrade** tool.
- An invalid directory might have been specified when launching the **WASPreUpgrade** tool.

- The **WASPreUpgrade** tool might exit without backing up your previous environment.

  The tool might appear to run successfully as in the following example:

  ```
  MIGR0201I: The migration function initialized log file WASPreUpgrade.log.
  MIGR0300I: The migration function is starting to save the existing Application Server environment.
  MIGR0302I: The existing files are being saved.
  MIGR0303I: The existing Application Server environment is saved.
  MIGR0420I: The first step of migration completed successfully.
  ```

  You might also see a message similar to the following example in the migration trace file:

  ```
  [10/9/08 18:26:40:363 CDT] 00000000 Save     1
  Skipped instance dmgr01 because user root /opt/migration_backup/profiles/dmgr01 does not exist.
  ```

  The **WASPreUpgrade** tool writes out a copy of a `profileList.ser` file that contains pointers to the backup directory to be used by the **WASPostUpgrade** tool. If that file is not subsequently deleted by migration for any reason, the old paths are used instead of the real paths when you run the **WASPreUpgrade** tool in later migrations. To resolve this issue, you can safely delete the `profileList.ser` file and rerun the **WASPreUpgrade** tool.

For more information, read “WASPreUpgrade command” on page 34.

Note: When migrating a Version 6.1 federated node to Version 8.5, the **WASPreUpgrade**command might fail. You might receive an error similar to the following example:

```
[07/16/2011 11:07:10:357 CDT] MIGR0344I: Processing configuration file
/opt/WAS61fep/profiles/v6109node74_01/config/cells/ndcell/clusters/Station1EJBCluster
/resources.xml.
[07/16/2011 11:07:10:436 CDT] org.eclipse.emf.ecore.resource.Resource$IOWrappedExcept
ion: Unresolved reference 'DataSource_1310769433958'.
(file:/opt/WAS61fep/profiles/v6109node74_01/config/cells/ndcell/clusters/Station1EJBC
luster/resources.xml, 9, 323)
java.lang.Exception: org.eclipse.emf.ecore.resource.Resource$IOWrappedException:
Unresolved reference 'DataSource_1310769433958'.
(file:/opt/WAS61fep/profiles/v6109node74_01/config/cells/ndcell/clusters/Station1EJBC
luster/resources.xml, 9, 323)
at com.ibm.wsspi.migration.document.wccm.WCCMDocument.setInputStream(WCCMDocument.ja
va:162)
```

You might encounter this problem on a WebSphere Version 6.1 node when a DB2 database using IBM JCC Provider Driver has been created, and the WebSphere Version 6.1 node is synchronized to the Version 8.5 deployment manager. The Version 6.1 node does not support the Version 7.0 or above driver level. The node synchronization process is failing to remove all of the driver definitions.

To resolve this problem, backup any resources.xml files that are to be modified. Stop the Version 6.1 node agent process. Edit the WebSphere Version 6.1 node resources.xml files and remove the orphaned resources.jdbc:CMPConnectorFactory entries prior to running the WASPreUpgrade command. Do not edit the deployment manager copy.

– Problems can occur when you are using the **WASPostUpgrade** tool.

  - You might see an exception in the WASPostUpgrade logs after migrating a federated node that is similar to the exception that is highlighted in the following text:

```
MIGR0304I: The previous WebSphere environment is being restored.
MIGR0367I: Backing up the current Application Server environment.
CEIMI0006I Starting the migration of Common Event Infrastructure.
MIGR0486I: The Transports setting in file server.xml is deprecated.
MIGR0486I: The PMIService:initialSpecLevel setting in file server.xml is deprecated.
MIGR0486I: The PMIService:initialSpecLevel setting in file server.xml is deprecated.
MIGR0404W: Do not use the node agent in the old configuration. It has been disabled.
MIGR0351I: The migration function is attempting to synchronize with the deployment
           manager using the SOAP protocol.
MIGR0241I: Output of syncNode.
ADMU0116I: Tool information is being logged in file
           /usr/WAS80/profiles/AppSrv01/logs/syncNode.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU0401I: Begin syncNode operation for node aaixae15aNode01 with Deployment
           Manager packppc.rtp.raleigh.ibm.com: 8879
ADMU0016I: Synchronizing configuration between node and cell.
AWXJR0006E   The file, /usr/WAS80/java/jre/PdPerm.properties, was not found.
ArchiveUtil.toLocalURLs
ArchiveUtil.toLocalURLs
ArchiveUtil.toLocalURLs
ADMU0402I: The configuration for node aaixae15aNode01 has been synchronized
           with Deployment Manager packppc.rtp.raleigh.ibm.com: 8879
MIGR0352I: The synchronization with the deployment manager is successful.
CEIMI0007I The Common Event Infrastructure migration is complete.
MIGR0307I: The restoration of the previous Application Server environment is complete.
MIGR0271W: Migration completed successfully, with one or more warnings.
```

This exception occurs during the syncNode operation, and it is listed as an error; but it does not cause any failures. The overall action completes successfully, and the message does not reoccur. After the server on the migrated federated node is started, the file in question is regenerated. You can ignore this message.

- IBM i You might receive the following error messages:

```
MIGR0484E: No profiles or instances found with name -profileName wasio2651.
MIGR0272E: The migration function cannot complete the command.
```

The old and new profile names must match. Rerun the **WASPostUpgrade** command with a Version 8.5 profile that matches the name given as the value following -profileName in the MIGR0484E message.

- A “Not found” or “No such file or directory” message is returned.

This problem can occur if you are trying to run the **WASPostUpgrade** tool from a directory other than the Version 8.5 *app_server_root*\bin. Verify that the **WASPostUpgrade** script resides in the Version 8.5 *app_server_root*\bin directory, and launch the file from that location.

- You receive the following message:

```
MIGR0102E: Invalid Command Line. MIGR0105E: You must specify the primary node name.
```

The most likely cause of this error is that WebSphere Application Server Version 6.1 is installed and the **WASPostUpgrade** tool was not run from the bin directory of the Version 8.5 installation root.

To correct this problem, run the **WASPostUpgrade** command from the bin directory of theVersion 8.5installation root.

- When you migrate the federated nodes in a cell, you receive the following error messages:

```
MIGR0304I: The previous WebSphere environment is being restored.
 com.ibm.websphere.management.exception.RepositoryException:
 com.ibm.websphere.management.exception.ConnectorException: ADMC0009E:
   The system failed to make the SOAP RPC call: invoke
MIGR0286E: The migration failed to complete.
```

A connection timeout occurs when the federated node tries to retrieve configuration updates from the deployment manager during the WASPostUpgrade migration step for the federated node. Copying the entire configuration might take more than the connection timeout if the configuration that you are migrating to Version 8.5 contains any of the following elements:

- Many small applications

- A few large applications
- One very large application

The best practice is to modify the timeout value before running the **WASPostUpgrade** command to migrate a federated node.

1. Go to the following location in the Version 8.5 directory for the profile to which you are migrating your federated node:

   profile_root/properties
2. Open the soap.client.props file in that directory and find the value for the com.ibm.SOAP.requestTimeout property. This is the timeout value in seconds. The default value is 180 seconds.
3. Change the value of com.ibm.SOAP.requestTimeout to make it large enough to migrate your configuration. For example, the following entry would give you a timeout value of a half of an hour:

   com.ibm.SOAP.requestTimeout=1800

   Note: Select the smallest timeout value that will meet your needs. Be prepared to wait for at least three times the timeout that you select—once to download files to the backup directory, once to upload the migrated files to the deployment manager, and once to synchronize the deployment manager with the migrated node agent.
4. Go to the following location in the backup directory that was created by the **WASPreUpgrade** command:

   backupDirectory/profiles/*profile_name*/properties
5. Open the soap.client.props file in that directory and find the value for the com.ibm.SOAP.requestTimeout property.
6. Change the value of com.ibm.SOAP.requestTimeout to the same value that you used in the Version 8.5 file.

Alternatively, you might want to consider a solution in which you specify -includeApps script in the **WASPostUpgrade** command when you migrate the deployment manager to Version 8.5 if one or both of the following are true for your situation:

- You want to quickly migrate all nodes in the cell. After the entire cell is migrated, however, you are willing to manually run the application installation script for every application in the deployment manager backup directory and then synchronize the configuration with all migrated nodes.
- You are able to run without any applications installed.

Follow these steps to perform this alternative procedure:

1. Specify -includeApps script in the **WASPostUpgrade** command when you migrate the deployment manager to Version 8.5.
2. Migrate your entire cell to Version 8.5before installing any applications.
3. Run the **wsadmin** command to install each application.
   - Install the applications in the Version 8.5 configuration during normal operations or in applicable maintenance windows.
   - Specify -conntype NONE. For example:

     wsadmin -f *application_script* -conntype NONE
4. Synchronize the configuration with all of the migrated nodes.

Read “Migrating a large WebSphere Application Server, Network Deployment configuration with a large number of applications” on page 53 for more information on this alternative procedure.

- You receive the “Unable to copy document to temp file” error message. Here is an example:

```
MIGR0304I: The previous WebSphere environment is being restored.
com.ibm.websphere.management.exception.DocumentIOException: Unable to copy document to temp file:
  cells/sunblade1Network/applications/LARGEApp.ear/LARGEApp.ear
```

Your file system might be full. If your file system is full, clear some space and rerun the **WASPostUpgrade** command.

- You receive the following message:

  `MIGR0108E: The specified WebSphere directory does not contain WebSphere version that can be upgraded.`

  The following possible reasons for this error exist:

  - If WebSphere Application Server Version 6.1 is installed, you might not have run the **WASPostUpgrade** tool from the `bin` directory of the Version 8.5 installation root.
    1. Look for something like the following message to display when the **WASPostUpgrade** tool runs: `IBM WebSphere Application Server, Release 6.x`.

       This message indicates that you are running the WebSphere Application Server Release 6.1 migration utility, not the Version 8.5 migration utility.
    2. Alter your environment path or change the current directory so that you can launch the Version 8.5 **WASPostUpgrade** tool.
  - An invalid directory might have been specified when launching the **WASPreUpgrade** tool or the **WASPostUpgrade**.
  - The **WASPreUpgrade** tool was not run.

- You receive the following error message:

  `MIGR0253E: The backup directory` *`migration_backup_directory`* `does not exist.`

  The following possible reasons for this error exist:

  - The **WASPreUpgrade** tool was not run before the **WASPostUpgrade** tool.
    1. Check to see if the backup directory specified in the error message exists.
    2. If not, run the **WASPreUpgrade** tool.

       Read “WASPreUpgrade command” on page 34 for more information.
    3. Retry the **WASPostUpgrade** tool.
  - An invalid backup directory might be specified.

    For example, the directory might have been a subdirectory of the Version 6.1 tree that was deleted after the **WASPreUpgrade** tool was run and the older version of the product was uninstalled but before the **WASPostUpgrade** tool was run.
    1. Determine whether or not the full directory structure specified in the error message exists.
    2. If possible, rerun the **WASPreUpgrade** tool, specifying the correct full migration backup directory.
    3. If the backup directory does not exist and the older version it came from is gone, rebuild the older version from a backup repository or XML configuration file.
    4. Rerun the **WASPreUpgrade** tool.

- You decide that you need to run **WASPreUpgrade** again after you have already run the **WASPostUpgrade** command.

  During the course of a deployment manager or a federated node migration, **WASPostUpgrade** might disable the old environment. If after running **WASPostUpgrade** you want to run **WASPreUpgrade** again against the old installation, you must run the `migrationDisablementReversal.jacl` script located in the old *`app_server_root`*`/bin` directory. After running this JACL script, your Version 6.1 environment will be in a valid state again, allowing you to run **WASPreUpgrade** to produce valid results.

- A federated migration fails with message MIGR0405E.

  The migration that has taken place on your deployment manager as part of your federated migration has failed. For a more detailed reason for why this error has occurred, open the folder *`your_node_name`*`_migration_temp` located on your deployment manager node under the `...DeploymentManagerProfile/temp` directory. For example:

  `/websphere80/appserver/profiles/dm_profile/temp/nodeX_migration_temp`

The logs and everything else involved in the migration for this node on the deployment manager node are located in this folder. This folder will also be required for IBM support related to this scenario.

- Version 8.5 applications are lost during migration.

  If any of the Version 8.5 applications fail to install during a federated migration, they will be lost during the synchronizing of the configurations. The reason that this happens is that one of the final steps of **WASPostUpgrade** is to run a **syncNode** command. This has the result of downloading the configuration on the deployment manager node and overwriting the configuration on the federated node. If the applications fail to install, they will not be in the configuration located on the deployment manager node. To resolve this issue, manually install the applications after migration. If they are standard Version 8.5 applications, they will be located in the *app_server_root*/installableApps directory.

  To manually install an application that was lost during migration, use the **wsadmin** command to run the install_*application_name*.jacl script that the migration tools created in the backup directory.

  In a Linux environment, for example, use the following parameters:

  ```
  ./wsadmin.sh -f migration_backup_directory/install_application_name.jacl -conntype NONE
  ```

  IBM i Use the following parameters:

  ```
  app_server_root/bin/wsadmin -f migration_backup_directory/install_application_name.jacl -conntype NONE
  ```

- Version 8.5 applications fail to install.

  Manually install the applications using the **wsadmin** command after **WASPostUpgrade** has completed.

  To manually install an application that failed to install during migration, use the **wsadmin** command to run the install_*application_name*.jacl script that the migration tools created in the backup directory.

  In a Linux environment, for example, use the following parameters:

  ```
  ./wsadmin.sh -f migration_backup_directory/install_application_name.jacl -conntype NONE
  ```

  IBM i Use the following parameters:

  ```
  app_server_root/bin/wsadmin -f migration_backup_directory/install_application_name.jacl -conntype NONE
  ```

Read “WASPostUpgrade command” on page 37 for more information.

– The trace file exceeds its 400 megabytes allocation, but WASPostUpgrade is still running. If additional disk space is not available, the migration will fail.

  If you think you might encounter this problem during your migration, complete the following actions:

  1. Stop the Migration wizard before the WASPostUpgrade command is issued.
  2. Run the WASPostUpgrade command from the command line for each profile you are migrating.

     When you run the WASPostUpgrade command from the command line:

     - Include the **-oldProfile** and **-profileName** parameters to indicate the profile you want to migrate.
     - Add the **com.ibm.ejs.ras.TraceNLS*** parameter to the trace string to reduce the size of your trace log. For example you might want to specify the following trace setting:

       ```
       com.ibm.ejs.ras.TraceNLS*=info
       ```

• IBM i If you select the option for the migration process to install the enterprise applications that exist in the Version 6.1 configuration into the new Version 8.5 configuration, you might encounter some error messages during the application-installation phase of migration.

  The applications that exist in the Version 6.1 configuration might have incorrect deployment information—typically, invalid XML documents that were not validated sufficiently in previous WebSphere Application Server runtimes. The runtime now has an improved application-installation validation process and will fail to install these malformed EAR files. This results in a failure during the application-installation phase of **WASPostUpgrade** and produces an “E” error message. This is considered a “fatal” migration error.

  If migration fails in this way during application installation, you can do one of the following:

  – Fix the problems in the Version 6.1 applications, and then remigrate.

– Proceed with the migration and ignore these errors.

  In this case, the migration process does not install the failing applications but does complete all of the other migration steps.

  Later, you can fix the problems in the applications and then manually install them in the new Version 8.5 configuration using the administrative console or an install script.

- IBM i If you migrate a deployment manager to IBM Version 8.5 from a Version 6.1 configuration that was migrated from a Version 5.1 deployment manager, the **syncNode** command might fail on any Version 5.1 federated nodes in the cell.

  For example, you might see messages similar to the following text when you run the **syncNode** command on a Version 5.1 node:

```
bash-3.00# ./syncNode.sh  dmgrhostname 8879 -username
MyAdminUser -password MyAdminPassword

ADMU0116I: Tool information is being logged in file
           /usr/WebSphere/AppServer/logs/syncNode.log
ADMU0401I: Begin syncNode operation for node My511Node
           with Deployment Manager dmgrhostname: 8879
ADMU0111E: Program exiting with error:
           com.ibm.websphere.management.exception.
           AdminException:
ADMU2092E: The node and Deployment Manager must have
           the same product extensions, but they do
           not match.  The node product extension is
           BASE and the Deployment Manager product
           extension is PME.
ADMU0211I: Error details may be seen in the file:
           /usr/WebSphere/AppServer/logs/syncNode.log
ADMU1211I: To obtain a full trace of the failure, use
          the -trace option.
```

- IBM i Because of the inclusion of the javax.ejb.Remote annotation in the EJB 3.0 specification, certain EJB 2.1 beans might fail to compile if Enterprise Java Beans are written to import the entire javax.ejb and java.rmi packages. Compilation errors similar to those in the following example might occur:

```
ejbModule/com/ibm/websphere/samples/trade/ejb/QuoteHome.java(17): The type Remote is ambiguous
```

- IBM i When you install WebSphere Application Server Version 6.1 and federate a node to a Version 8.5 deployment manager, you might experience unexpected and continuous security exception messages.

  The `system.out` logs of the node agent contain the following exceptions:

```
[7/8/08 16:41:31:416 EDT] 0000001c DefaultTokenP E
HMGR0149E: An attempt to open a connection to core group
DefaultCoreGroup has been rejected. The sending process
has a name of wasinst101Cell01\ndrack104Node08\server1
and an IP address of /9.42.92.86. Global security in the
local process is Enabled. Global security in the sending
process is Enabled. The received token starts with
x2>W 9 Sv?. The exception is
com.ibm.websphere.security.auth.WSLoginFailedException:
Validation of LTPA token failed due to invalid keys or
token type.

at com.ibm.ws.security.ltpa.LTPAServerObject.
validateToken(LTPAServerObject.java:876)
at com.ibm.ws.security.token.WSCredentialTokenMapper.
validateLTPAToken(WSCredentialTokenMapper.java:1178)
at com.ibm.ws.hamanager.runtime.DefaultTokenProvider.
authenticateMember(DefaultTokenProvider.java:214)
at com.ibm.ws.hamanager.coordinator.impl.DCSPluginImpl.
authenticateMember(DCSPluginImpl.java:723)
at com.ibm.ws.dcs.vri.transportAdapter.rmmImpl.ptpDiscovery.
DiscoveryRcv.acceptStream(DiscoveryRcv.java:266)
```

```
at com.ibm.rmm.ptl.tchan.receiver.PacketProcessor.
fetchStream(PacketProcessor.java:470)
at com.ibm.rmm.ptl.tchan.receiver.PacketProcessor.
run(PacketProcessor.java:917)
```

The deployment manager uses Version 8.5 and all of the nodes and alias nodes are using Version 6.1. To resolve this problem, upgrade all Version 6.1 nodes to Version 6.1.0.17 or later.

New ports that are registered on a migrated Version 8.5 node agent include: WC_defaulthost, WC_defaulthost_secure, WC_adminhost, WC_adminhost_secure SIB_ENDPOINT_ADDRESS, SIB_ENDPOINT_SECURE_ADDRESS ,SIB_MQ_ENDPOINT_ADDRESS, SIB_MQ_ENDPOINT_SECURE_ADDRESS. These ports are not needed by the node agent, and can be safely deleted.

What to do next

If you did not find your problem listed, contact IBM support.

Chapter 15. IBM Suggests

Most of the following links will take you to information that is not part of the formal product documentation and is provided "as is." Some of these links go to non-IBM Web sites and are provided for your convenience only and do not in any manner serve as an endorsement by IBM of those Web sites, the material thereon, or the owner thereof.

Chapter 16. Migrating, coexisting, and interoperating

This section covers all aspects of migration, coexistence, and interoperability. Migrating is copying the configuration from a previous release of this product into a new release. Coexisting is running a new release of WebSphere Application Server on the same machine at the same time as you run an earlier release or running two installations of the same release of WebSphere Application Server on the same machine at the same time. Interoperating is exchanging data between two different systems, such as coexisting product installations.

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

APACHE INFORMATION. This information may include all or portions of information which IBM obtained under the terms and conditions of the Apache License Version 2.0, January 2004. The information may also consist of voluntary contributions made by many individuals to the Apache Software Foundation. For more information on the Apache Software Foundation, please see http://www.apache.org. You may obtain a copy of the Apache License at http://www.apache.org/licenses/LICENSE-2.0.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Intellectual Property & Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. For a current list of IBM trademarks, visit the IBM Copyright and trademark information Web site (www.ibm.com/legal/copytrade.shtml).

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

Index