IBM

@server

WebSphere Application Server for iSeries

Migration

Version 5.0.2

IBM

@server

WebSphere Application Server for iSeries

Migration

Version 5.0.2

> **Note**
>
> Before using this information and the product it supports, be sure to read the information in "Notices," on page 53.

Contents

Migration

Migration tasks and tools allow you to transfer your existing configuration settings and existing applications to the new version of the product.

Product migration functions are provided by the migration tools in IBM WebSphere Application Server, Version 5. This table shows the supported migration paths:

Existing version *	New version
3.5.x Standard Edition	5 Base and Network Deployment
3.5.x Advanced Edition	5 Base and Network Deployment
4.0.x Advanced Single Server Edition	5 Base and Network Deployment
4.0.x Advanced Edition	5 Base and Network Deployment

* See "Migration prerequisites" on page 15 for the minimum levels required for migration.

This document provides detailed step-by-step instructions to successfully migrate your WebSphere Application Server Version 3.5.x and Version 4.0.x instances to WebSphere Application Server Version 5. The following topics take you through the migration process. It is recommended that you read each topic in the order listed, and that you read through the migration documentation at least once before you start the migration process.

WebSphere Application Server Version 5 can coexist with all previous versions of WebSphere Application Server. Depending on the previous version of WebSphere Application Server, there may be port conflicts which need to be resolved. See Coexistence with previous versions of WebSphere Application Server ➡️ for more information .

For more information on the typical migration scenario, see **"Overview" on page 2**.

"Step 1: Plan your migration" on page 11
This topic describes the steps you should complete before starting the migration process. It also includes information on how to evaluate your current environment to ensure you meet the requirements for migration.

"Step 2: Migrate your applications" on page 16
This topic describes how to migrate your applications to Version 5. This involves modifying your applications to use the specifications supported by Version 5.

3. MIGRATING YOUR INSTANCES

"Step 3: Migrate your WebSphere Application Server instances" on page 23
This topic provides step-by-step instructions to migrate a WebSphere Application Server instance from an earlier version of WebSphere to a new WebSphere Application Server Version 5 instance or to the default WebSphere Application Server Version 5 instance. Additionally, instructions are provided for migrating to WebSphere Application Server Network Deployment Version 5.

4. COMPLETING THE MIGRATION

"Step 4: Complete the migration" on page 49
This topic describes the manual steps that you might need to perform to complete the migration process. These are steps that are not performed automatically by the WebSphere Application Server migration tools.

Overview

Migration consists of stages that include scoping, skills migration, code migration, and runtime migration. Refer to these articles for reference on these stages and other general information on the migration process:

- Part 1: Designing Software for Change
- Part 2: Stages of Migration
- Part 3: Migration assessment
- Migrating to WebSphere V5: An End-to-End Migration Guide

Note: These articles are not iSeries specific and may contain information not pertinent to iSeries.

WebSphere Application Server migration leverages the existing environment and applications and changes them to be compatible with the Version 5 environment. Existing application components and configuration settings are applied to the Version 5 environment during the migration process.

Migration involves modifying your applications so that they run in Version 5 and then migrating those applications and your environment to Version 5. The latter step can be performed by using the migration tools shipped with the product.

Migrating from Version 3.5.x to Version 5 involves migrating to the Java 2 Platform, Enterprise Edition (J2EE) 1.3 specification. Version 3.5.x applications are not J2EE compliant; hence, this process involves modifying the application.

Migrating from Version 4.x to Version 5 involves minimal change because both releases implement the Java 2 Platform, Enterprise Edition (J2EE) specification. Version 4 implements J2EE 1.2, while version 5 implements J2EE 1.3. Most Version 4.x applications run without change in Version 5.

Migrating from Version 5 Express to Version 5 (Base) or Network Deployment involves minimal changes because both releases implement a common set of the Java 2 Platform, Enterprise Edition (J2EE) specification.

Migration Tools

Product migration functions are provided by the the WebSphere Application Server migration tools. These tools perform migration from Version 3.5.x, 4.x and 5.0.x Express to Version 5 (Base) and Network Deployment. "Migration configuration mapping" describes how objects and attributes are mapped to the Version 5 environment by the migration tools. The migration tools are comprised of the following commands:

- **WASPreUpgrade** saves Version 3.5, 4.0.x, and 5.0.x Express configuration data and applications from a previous version to a backup directory. For more information, see The WASPreUpgrade script in the *Administration* topic.
- **WASPostUpgrade** restores Version 3.5, 4.0.x, and 5.0.x Express configuration data and applications into Version 5. This command uses the output from the WASPreUpgrade command. For more information, see The WASPostUpgrade script in the *Administration* topic.

The following tools are also available to assist you in migrating your applications:

- **clientUpgrade** 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 5. For more information, see The clientUpgrade scripts in the *Administration* topic.
- **earconvert** migrates J2EE 1.2 application modules and their respective resource references to J2EE 1.3 application modules in an enterprise archive (EAR) file so that these application clients can run in WebSphere Application Server Version 5. For more information, see The earconvert script in the *Administration* topic.
- **MigrateWC** converts JSP Specification .91 or 1.0 application components to JSP Specification 1.1 components. It also converts Servlet 2.1 specification components to Servlet 2.2 specification components. The converted components are written to a new file. This tool is not shipped with WebSphere Application Server. For more information and to download this tool, see the WebSphere Developer Domain . Search in the **Downloads** section for MigrateWC.
- **CACT** analyzes compiled servlet and enterprise bean Java class files and provides information on any API that is deprecated or not supported in WebSphere Application Server Version 4 or Version 5. This tool is not shipped with WebSphere Application Server. For more information and to download this tool, see the WebSphere Developer Domain . Search in the WSDD **Library** section for CACT.
- **WebSphere Development Studio Client (WDSC)** is part of the WebSphere Studio family of application development tools. This tool can be used to import your application source files, set the desired J2EE version in the product, for example to J2EE Version 1.3, and rebuild the source files. Use of changed, not supported, and deprecated APIs are reported through compilation errors. After it is built, the application can be tested with the version of WebSphere Application Server that is included with the tool. This tool is not shipped with WebSphere Application Server. For more information on this tool, see the WebSphere Development Studio Client for iSeries Web site .

Migration configuration mapping

This topic describes how the migration tools map the configuration from the previous release to Version 5 for the basic migration scenario. The basic migration scenario involves migrating a single WebSphere Application Server Version 3.5, Version 4, or Version 5.0.x Express instance to a WebSphere Application Server Version 5 instance located on the same iSeries server.

The migration tools map objects and attributes existing in the version you are migrating from to the corresponding objects and attributes in the Version 5 environment.

- **Stdin, stdout, stderr, passivation and working directories**
  The location for these directories is typically the logs directory under the WebSphere Application Server instance directory. For example in Version 4 Advanced Edition, the logs directory for the default instance is /QIBM/UserData/WebASAdv4/default/logs. For Version 5, the default location for stdin, stdout, and stderr is still the logs directory located under the WebSphere Application Server instance directory. For example, the logs directory for the default instance is /QIBM/UserData/WebAS5/Base/default/logs. The migration tools attempt to migrate existing passivation and working directories. Otherwise, appropriate Version 5 defaults are used. For more information on passivation directories, see the help topic EJB container settings . For more information on working directories, see the help topic Process definition settings .

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

- **Property files from Version 3.5.x and 4.0.x**
  Migration does not process property files from Version 3.5.x and 4.0.x that are located in the properties directory for an instance. You must manually convert settings in these files to the Version 5 equivalent configuration.

- **Command line parameters**
  The migration tools convert appropriate command line parameters to JVM settings in the server process definition. However some settings, such as memory heap sizes, are not migrated because their role in the Version 5 configuration either does not exist, has different meaning, or has different scope. For information on how to change the process definition settings, see the help topic Process definition . For information on how to change the Java virtual machine settings, see the help topic Java virtual machine .

- **Bootstrap port**
  If the administrative server for your Version 3.5.x or Version 4 Advanced Edition WebSphere Application Server instance is using port 900 for the bootstrap port, the migration tools map this port to port 2809 for the bootstrap port (also known as the name service port) in the Version 5 instance. Also if the application server name of the Version 3.5.x or Version 4 instance matches the Version 5 instance, the port number remains unchanged from the value of the name service port used when the Version 5 instance was created. If the instance you are migrating from uses a port other than 900 for the bootstrap port, the migration tools use that same port for the bootstrap port of the Version 5 instance. However, if the -portBlock was specified during the call to WASPostUpgrade, a new port value is given to each application server that is migrated to Version 5. The bootstrap port is specified in the admin.properties file located in the properties file WebSphere Application Server Version 3.5.x or Version 4.0.x Advanced Edition instance.
  If you are migrating from Version 4 Advanced Single Server Edition, the bootstrap port is specified in the server configuration file for the application server. The migration tools map the bootstrap port as indicated above for an Advanced Single Server instance.

- **Default Server**
  The name of the default server in Version 3.5.x and 4.0.x is **Default Server**. The name of the default server in Version5 is **server1**. Thus, the Default Server in Version 3.5.x and 4.0.x is migrated to the **server1** server in Version 5.

- **JDBC drivers and datasources**
  The JDBC connection manager from Version 4 is still provided as a configuration option. Using this configuration option enables J2EE 1.2 applications to run unaltered. If you migrate a Version 4 application to Version 5, using the Version 5 migration tools, the application automatically uses the Version 4 connection manager after migration. However, EJB 2.0 modules in J2EE 1.3 applications cannot use the JDBC connection manager from Version 4.

- **Name bindings**
  There is a new namespace structure in Version 5. Enterprise bean references in Version 5 are incompatible with the references that are expected by application components such as servlets or enterprise beans running in Version 3.5 or Version 4. This only affects topologies where applications running in previous WebSphere Application Releases must access enterprise beans running in Version

5. This does not impact topologies where all application components are running in Version 5. The following instructions describe how to externalize the EJB in Version 5 so that a Version 3.5 or Version 4 application component can access it:

1. Start the Version 5 administrative console.
2. Navigate to **Environment —> Naming —> Name Space Bindings**.
3. Click **New**.
4. Specify Ejb for the **Binding Type**.
5. Specified SHARED for the **Node**.
6. In the **Binding Identifier**, **Name in Name Space**, and **JNDI Name** fields, specify the reference used for enterprise bean in Version 3.5.x or Version 4.0.x.
7. Click **OK**.
8. Save your changes.

- **Node name**
  A Version 3.5.x or Version 4.0.x administrative repository can contain more than one node name. Each node may have WebSphere Application Server resources associated with it. The migration tools process only the node (and its associated resources) whose node name matches the TCP/IP host name of the iSeries on which the WebSphere Application Server instance resides. The migration tools identify the nodes names in any configuration files it is processing, and migrates only resources that belong to a node whose name matches the long or short host name of the iSeries. The host name of the iSeries can be found under CFGTCP option 12 (CHGTCPDMN).

- **PageList servlet**
  The configuration of the PageList servlet has changed in Version 5. Direct use of the servlet has been deprecated. The PageList servlet is available as part of the servlet extension configuration in the WAR file. All references are updated to the servlet configuration supported in Version 5.

  You can also use the Application Assembly Tool (AAT) to modify the servlet configuration. If you have used or extended the PageList servlet, you might see an error similar to this when running a migrated application that uses the servlet:

  ```
  Error 500: No PageList information is configured for servlet EmpInfoApp.SearchByDept
  ```

  Use the Application Assembly Tool (AAT) to correct the error, by moving the usage or extension to your migrated EAR file:

1. Start the AAT and load the EAR file that generates the error.
2. Open **Web modules** within the EAR file.
3. Expand the **Web module** that generated the error.
4. Open the **Web components** and find the component that generated the error.
5. Expand **Servlets**.
6. Add your extension information under **PageList Extensions**.
7. Save your EAR file and redeploy it.

  For more information on migrating your servlets to Version 5, see Migrate Web applications in the *Application Development* topic.

- **Properties directory, classes directory, and lib/app directory**
  Migration does not copy files from these prior version directories into the Version 5 configuration. Property files are not compatible between versions. The `classes` directory might contain incompatible files. You must migrate these files on an individual basis to move them into the Version 5 configuration.

  If you are migrating to WebSphere Application Server Version 5 or Version 5.0.1, then applications relying on the application extensions classloader or the `lib/app` directory must manually migrate to WebSphere Application Server Version 5. This does not apply if you are migrating to WebSphere Application Server Version 5.0.2 or later. Follow these steps to manually migrate:

1. Create the `lib/app` subdirectory in the Version 5 /QIBM/UserData/WebAS5/Base/*instance* directory, where *instance* is the name of your instance. QEJBSVR should be the owner and have *RWX authority to the directory and and files placed in the directory.

2. Do one of the following tasks:

   – Copy the files from the Version 4 lib/app subdirectory to the Version 5 lib/app subdirectory. Enter these commands from a Qshell command prompt:

      a.
      ```
      mkdir /qibm/userdata/webas5/base/instance/lib
      ```

      b.
      ```
      mkdir /qibm/userdata/webas5/base/instance/lib/app
      ```

      c.
      ```
      cp /qibm/userdata/webasadv4/instance/lib/app/*
      /qibm/userdata/webas5/base/instance/lib/app
      ```

      d.
      ```
      chown -R QEJBSVR /qibm/userdata/webas5/base/instance/lib
      ```

   – Create shared libraries in Version 5 to contain the classes contained in the Version 4 applications extensions directory. For more information, see Administer shared libraries in the *Administration* topic.

      Note: This is the preferred method, because it is more portable and flexible than using the lib/app subdirectory.

- **Samples**

  There is no migration of samples from previous versions. There are equivalent Version 5 samples that you can use. The samples are not preinstalled. See Samples Gallery in the *Application Development* topic for more information on installing and using the samples.

- **Security**

  If security is enabled for applications in your Version 4 instance, the same applications may not function correctly in the Version 5 environment. This is because Java 2 Security is enabled by default when you enable security in Version 5. Java 2 Security requires you to grant security permissions explicitly. There are several techniques that you can use to define different levels of Java 2 Security in Version 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 5 `properties` directory to enterprise applications as they are being migrated. The migration tools perform this task while moving Version 4 applications into Version 5.

  Global security that uses LTPA authentication in Versions 3.5.x and 4.0.x is migrated to the base WebSphere Application Server product. However, although global security was enabled in Versions 3.5.x and 4.0.x, it is disabled during and after migration to Version 5. After migration, LTPA keys are generated when security is enabled in the administrative console. Security is left enabled when migrating instances that use the LocalOS authentication mechanism.

  Version 4.x introduced properties to support tuning the JNDI search timeout value along with LDAP reuse connection. These two properties are now settings in the Security Center of the Version 5 administrative console. There is no migration of Version 4.x property values to the Version 5 settings.

  – The jndi.LDAP.SearchControl.TimeLimit property is equivalent to the Version 5 Search Timeout setting, which is `300` by default in Version 5.

  – The jndi.LDAP.URLContextImplementation property is equivalent to the Version 5 Reuse Connection setting, which is `true` by default in Version 5.

  Use the Version 5 administrative console to change these settings to match your Version 4 property values, if necessary.

  For more information on migrating your security configurations to Version 5, see Migrate security configurations from previous releases in the *Security* topic.

- **Servlet package name changes**

  The package that contains the DefaultErrorReporter, SimpleFileServlet, and InvokerServlet servlets has

changed for Version 5. In Versions 3.5.x and 4.0.x, the servlets are in the com.ibm.servlet.engine.webapp class. In Version 5, the servlets are in the com.ibm.ws.webcontainer.servlet class. For more information, see Migrate Web applications in the *Application Development* topic.

- **Transport ports**
  The migration tools migrate all ports. If the -portBlock parameter is specified in the WASPostUpgrade script a new value is assigned to each transport that is migrated. For more information, see The WASPostUpgrade script in the *Administration* topic. The tools log port conflict warnings if a port is already defined in the configuration. You must resolve port conflicts before running the servers that are in conflict, at the same time. To resolve port conflicts, use the The chgwassvr script.

  The default transport type of the Servlet Engine in Version 3.5.x is Open Servlet Engine (OSE). Because Version 5 no longer supports OSE transport, the migration tools map these transports to Web Container transports, using the same port assignments.

  You must manually add VirtualHost alias entries for each port. For more information, see Configure the virtual host in the *Installation* topic.

- **Web modules**
  The Version 5 level of J2EE requires Web container behavior changes in regard to setting content type. If a default servlet writer does not set the content type, not only does the Version 5 Web container no longer default to it, the Web container returns the call as "null". This might cause some browsers to display resulting Web container tags incorrectly. Migration sets the autoResponseEncoding IBM extension to true for Web modules as it migrates enterprise applications. This prevents the problem. For more information on migrating your web applications to Version 5, see Migrate Web applications in the *Application Development* topic.

- **Version 3.5 to Version 5 migration**
  The migration tools assist in the transition from Version 3.5.x to Version 5, by migrating system configurations and creating J2EE artifacts, including J2EE security roles mapping. The migration tools create initial J2EE enterprise applications based on Version 3.5.x configurations. However, because of the significant change in application structures, plan to carefully test and fine tune migrated applications, using development and deployment tools, to determine exactly how the applications function in Version 5.

  Analyze the WASPostUpgrade.log file for detailed information about migrated enterprise beans. The J2EE programming model specifies an architecture for how applications are created and deployed. Because applications in Version 3.5.x do not have the same architecture, the WASPostUpgrade tool recreates applications. It creates all migrated Web resources and enterprise beans in J2EE applications. It maps all enterprise applications from the Version 3.5.x installation into J2EE applications with the same name, deployed in the same server.

  The WASPostUpgrade tool maps Web resources and enterprise beans that are not included in an enterprise application, into a default J2EE application that includes the name of the server. The tool maps Web applications to J2EE WAR files. The tool deploys enterprise beans as EJB 1.1 beans in J2EE JAR files. The tool combines resources in a J2EE EAR file and deploys it in the Version 5 configuration. There are some differences between the EJB 1.0 and EJB 1.1 Specifications, but in most cases, EJB 1.0 beans can run successfully as EJB 1.1 beans.

- **Version 3.5 to Version 5 mapping details**
  Note: See "API and Specifications for version 3.5.x" on page 11 to plan your application migration requirements.

  - **datasources.xml**
    You can use a Version 3.5.x datasources.xml file to augment datasource configuration settings. Version 3.5.x stores the file in the properties directory. The migration tools migrate an existing datasources.xml file by merging properties in the file into the datasource and JDBC driver configuration. For more information on migrating your data access applications to Version 5, see "Migrate a Version 4 data access application to Version 5" on page 19.

  - **Enterprise applications**
    The Version 3.5.x enterprise application entries are optional; they are most often used to organize sets of objects together for Security definitions. The enterprise bean and Web application portions of

the enterprise application point to their respective entries in other portions of the XML file. Each enterprise application is processed to create a J2EE application with the same name. The enterprise bean and Web application entries are used as pointers to the definitions of enterprise beans and Web applications. The details of these entries are then used to build a J2EE application.

For enterprise bean files, the JAR file definition is used to find the JAR files to redeploy and add to the J2EE application The Web application document root entries are used to find the resources used within the Web application (HTML, JSP pages, and so forth) These files are copied to the WAR file within the J2EE application. The Web application classpath entries are used to find servlets and JAR files that are copied to the WAR file within the J2EE application.

Enterprise applications are created during the migration from Version 3.5.x. These are created as J2EE 1.2 compatible enterprise applications and contain EJB 1.1-, Servlet 2.2- and JSP 1.1-level modules. This provides the most straight forward compatibility and enables interoperability with previous WebSphere Application Server versions.

– **Enterprise beans**
Version 3.x supports only the EJB 1.0 Components Specification level. Version 5 supports EJB 1.1 and 2.0 components. However, many EJB 1.0 beans can successfully deploy as EJB 1.1 beans. The migration tools redeploy enterprise beans automatically as part of the application migration phase. Check the `WASPostUpgrade.log` file for deployment details of these enterprise beans. Make any necessary changes and redeploy. No redeployment is required when moving EJB 1.1 JAR files from Version 4. Specify only one backend datastore vendor per JAR file. If there are enterprise beans that use different backends, package them into separate JAR files. For more information on migrating your enterprise beans to Version 5, see Migrate enterprise bean code to the supported specification in the *Application development* topic.

– **J2EE security**
The security authorization model in version 3.5.x is based on the notion of Enterprise Application and Method Groups. The cross product of the enterprise application and the method groups is a WebSphere Application Server permission. J2EE specifies an authorization model based on roles.

To convert from the WebSphere Application Server permission model in version 3.5.x to the role based authorization model in Version 5, the migration tools create a one-to-one mapping from a WebSphere Application Server permission to a new role under that application. Therefore, for each enterprise application and each method group in Version 3.5.x, the migration tools create a role in Version 5, contained in the J2EE application deployment descriptor. The authorized subjects for each role are contained in an authorization table found in the J2EE application binding.

J2EE specifies an authorization model based on roles. WebSphere Application Server interprets the role to mean a set of permissions to access a resource. In the case of an enterprise bean method invocation, the permission to access the method on a particular bean is specified by a method permission. This method permission is associated with one or more roles in the deployment descriptor in the bean jar file.

In the case of accessing Web resources, the permission to access a Web URI and invoke a HTTP method on that URI is specified in terms of Web resource collections and security constraints in J2EE. The deployment descriptor of the Web application WAR file contain the security constraints and Web resource collections. For more information on migrating your security configurations to Version 5, see Migrate security configurations from previous releases in the *Security* topic.

– **JSP levels**
Version 5 runs JSP 1.0 and 1.1 objects as JSP 1.2 objects, which is the only supported level. For more information, see Migrate Web applications in the *Application Development* topic.

– **Servlet Redirector**
Version 5 does not support the Servlet Redirector from previous versions. The migration tools ignore these objects.

– **Servlet package name changes when migrating from Version 3.5.x to Version 5**
If the Version 3.5.x configuration defines the SimpleFileServlet servlet, the servlet is not migrated. The migration tools set the FileServingEnabled attribute in the ibm-web-ext.xmi Web module file to `true`.

If the Version 3.5 configuration defines the InvokerServlet servlet, the servlet is not migrated. The migration tools set the ServeServletsByClassnameEnabled attribute in the ibm-web-ext.xml Web module file to `true`.

If the Version 3.5.x configuration defines the DefaultErrorReporter servlet, the servlet is migrated into the web.xml Web module file. Migration uses the new package to set the class name. See Migrate Web applications in the *Application Development* topic for more information on migrating your servlets to Version 5.

- **Version 4 to Version 5 migration**

  This migration is much less complicated than moving from Version 3.5.x. The Version 4.0.x configuration is already at the J2EE 1.2 level. Although Version 5 is at the J2EE 1.3 level, J2EE 1.2 objects are supported.

  Note: See "API and Specifications for version 4.x" on page 13 to plan your application migration requirements.

  – **Enterprise beans**

  No redeployment is required when moving EJB 1.1 JAR files from Version 4.0.

  Specify only one backend datastore vendor per JAR file. If there are enterprise beans that use different backends, package them into separate JAR files. See Migrate enterprise bean code to the supported specification in the *Application development* topic for more information on migrating your enterprise beans to Version 5.

  – **JMS Resources**

  All JMS resources from Version 4 are mapped into generic JMS resources in the Version 5 configuration. Reconfigure JMS resources that use IBM WebSphere MQ as IBM WebSphere MQ specific resources. MQ JMS resources have better integration with System Management. There is no need to manually define entries in the name space. You can see the backing MQ queue definitions through MQ JMS entries. See "Migrate JMS applications" on page 22 for more information on migrating your applications that use JMS to Version 5.

  – **JSP precompiling**

  In Version 4.0.x, the classes generated from JSP pages are in a package based on the directory structure of the WAR file. Any JSP at the top of the context root is in the `unnamed` package. JSP pages in subdirectories of the root are in packages named after the subdirectories. In Version 5, the classes generated from JSP pages are all in the package `org.apache.jsp`. Therefore, the class files are not compatible between versions.

  When migrating an enterprise application from Version 4.0.x to Version 5, recompile the JSP pages to regenerate the class files into the correct packages.

  The migration tools provide this support, by using the `-preCompileJSPs` option of the **wsadmin** tool during the installation of the application.

  Use the same option to install any Version 4.0.x enterprise applications that you manually move to Version 5. See Migrate Web applications for more information on migrating your JSP pages to Version 5.

  – **J2EE Security**

  You can apply security in two Version 4.x locations to enterprise applications. Information in the repository has precedence over information in the enterprise application bindings. The migration tools migrate information in the repository to the enterprise application.

  – **Secure Sockets Layer (SSL) migration**

  The following SSLConfig attributes that point to user-defined key files are migrated from WebSphere 4.0.x to 5 as follows:

  - **Version 4 settings:**

    ```
    <key-file-name>dir_name/WASLDAPKeyring.jks</key-file-name>
    <trust-file-name>dir_name/WASLDAPKeyring.jks</trust-file-name>
    ```

    where *dir_name* is the directory where the WASLDAPKeyring.jks file was placed when it was created.

  - **Version 5 settings:**

```
keyFileName="dir_name/WASLDAPKeyring.jks"   trustFileName="dir_name/WASLDAPKeyring.jks"
```

where *dir_name* is the directory where the WASLDAPKeyring.jks file was placed when it was created.

Also, the migration tools used by the installation wizard do not copy the key files (for examples, .jks, or .kdb) to the corresponding directory in the base WebSphere Application Server Version 5 product or the Network Deployment product. You must complete migration of the SSL configuration by copying qualifying key store files to Version 5 instance directories and modifying Version 5 SSL configuration data.

Qualifying key store files are non product key files (i.e. key files you created) or product key files you modified by replacing the product certificates with ones you created or obtained from a commercial certificate authority. Copying product key files from Version 4 to Version 5 instances result in the Version 5 default product key files being overlaid.

To complete migration of the SSL configuration:

1. After migration has completed and before enabling security, manually copy qualifying key files to the corresponding directory under each Version 5 base product node and Network Deployment node in the configuration.

2. Ensure *PUBLIC has *EXCLUDE authority and that the user profile your Version 5 application server runs under (QEJBSVR is the default user profile) has *RWX authority to the copied key files. For .kdb files, grant profile QTMHHTTP *RX authority.

3. Use the Digital Certificate Manager (DCM) to restash the password if copying a qualifying .kdb file. Follow these steps to restash the password:

   a. Start Digital Certificate Manager.
   b. Click **Select a Certificate Store**.
   c. Select **Other System Certificate Store**.
   d. Click **Continue**.
   e. Enter the Certificate store path and filename.
   f. Enter the Certificate store password.
   g. Click **Continue**.
   h. In the navigation panel, click **Manage Certificate Store**.
   i. Select **Change password**.
   j. Click **Continue**.
   k. Enter and confirm a new password.
   l. Ensure that **Automatic login** is selected.
   m. Click **Continue**.

4. The WASPostUpgrade tool creates one or more SSL repertoires in the Version 5 instance. The exact number of repertoires depends on how SSL is configured in the Version 4 instance. SSL repertoires must be edited to provide the correct location of the key files. The WASPostUpgrade tool populates the SSL repertoires in Version 5 instances with the Version 4 internal file system pathnames of the key files, hence pointing to the original key files in the Version 4 instances instead of the copies in the Version 5 instances. Ensuring the SSL repertoire information is correct is a required manual step. Incorrect SSL information may cause the server to fail, regardless of whether global security is enabled. Use the administrative console to edit the SSL repertoires.

   a. Login to the administrative console.
   b. Click **Security —> SSL**.
   c. For each SSL repertoire
      1) Click the repertoire to edit it.
      2) Edit the **KeyFileName** and **TrustFileName** fields as needed to point to the appropriate key file for the SSL repertoire. For example, when migrating the server1 instance from WebSphere Application Server Advanced Edition Version 4 to Version 5 in the

KeyFileName field change
/QIBM/UserData/WebASAdv4/default/etc/DummyServerKeyFile.jks to
${USER_INSTALL_ROOT}/etc/DummyServerKeyFile.jks. The server substitutes
/QIBM/UserData/WebAS5/Base/default for ${USER_INSTALL_ROOT} at runtime.

3) Click **OK**.

d. Save the configuration (synchronize nodes for Network Deployment).

e. Restart the server.

See Migrate security configurations from previous releases in the *Security* topic for more information on migrating your security configurations to Version 5.

– **Servlet package name changes when migrating from Version 4 to Version 5**
If the Version 4 web.xml Web module file defines the SimpleFileServlet servlet, the migration tools update the class name to reflect the Version 5 package. The tools also set the FileServing Enabled attribute to `true`.

If the Version 4 web.xml Web module file defines the InvokerServlet servlet, the migration tools update the class name to reflect the Version 5 package. The tools also set the ServeServletsByClassnameEnabled attribute to `true`.

If the Version 4 web.xml Web module file defines the DefaultErrorReporter servlet, the migration tools update the class name to reflect the Version 5 package. See Migrate Web applications in the *Application development* topic for more information on migrating your servlets to Version 5.

Step 1: Plan your migration

Complete these steps before you migrate to WebSphere Application Server Version 5.

1. Before you can migrate, you must install WebSphere Application Server Version 5. For more information on installing WebSphere Application Server Version 5 (Base), see WebSphere Application Server installation in the *Installation* topic. For more information on on installing WebSphere Application Server Network Deployment Version 5 (ND or Network Deployment), see WebSphere Application Server Network Deployment installation in the *Installation* topic.

2. In addition to migrating your instances and applications, you also need to familiarize yourself with the tools and features of Version 5. The Administration topic describes administrative tasks and the tools that are provided to perform those tasks.

3. Verify that you have the minimum prerequisites required for migration. For more information, see "Migration prerequisites" on page 15.

4. Evaluate the changes to API specification levels to determine which applications you need to migrate. Use these topics to plan your application migration requirements:
   • "API and Specifications for version 3.5.x"
   • "API and Specifications for version 4.x" on page 13

5. Evaluate the changes to configuration settings. For more information, see "Migration configuration mapping" on page 3.

API and Specifications for version 3.5.x

If your existing applications currently support different specification levels than are supported by this version of the product, it is likely 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 5.

From Version 3.5.x to version 5, main migration areas concern IBM entensions and migrating to Java 2 Platform, Enterprise Edition (J2EE).

The following table summarizes potential migration areas due to changes in supported specifications:

Specification	Support in Version 3.5	Support in Version 5	Must migrate from Version 3.5?	Details
Enterprise beans	EJB 1.0	EJB 2.0	Yes	Many EJB 1.0 applications can run unchanged in Version 5 although some changes may be required or recommended. For more information, see Migrate enterprise bean code to the supported specification in the *Application Development* topic.
JDBC	JDBC 1.0	JDBC 2.0	Yes	Many applications can run unchanged in Version 5 although some changes may be required or recommended.
JavaServer Pages	JSP .91	JSP 1.2	Yes	JSP 1.0 and 1.1 APIs are a pure subset of JSP 1.2. For more information, see Migrate Web applications in the *Application Development* topic.
	JSP 1.0		No	
	JSP 1.1		No	
Servlets	Servlet 2.1	Servlet 2.3	Yes	Many Servlet 2.1 applications can run unchanged in Version 5 although changes might be required or recommended. For more information, see Migrate Web applicationsin the *Application Development* topic.
	Servlet 2.2		No	Servlet 2.2 APIs are a pure subset of Servlet 2.3. For more information, see Migrate Web applicationsin the *Application Development* topic.

The following table summarizes potential migration areas due to changes in supported APIs:

API	Must migrate from Version 3.5?	Details
Security	Yes	Changes may be required due to J2EE security. For more information, see Migrate security configurations from previous releases in the *Securty* topic.
Sessions	Yes	Many applications can run unchanged in Version 5, although changes may be required or recommended. For more information, see "Migrate HTTP sessions" on page 21.

The following table summarizes potential migration areas due to changes in supported tools:

Tool	Must migrate from Version 3.5?	Details
XML Configuration Tool	Yes	Use JMX supported provided by WSAdmin. For more information, see The wsadmin administrative tool in the *Administration* topic.
WebSphere Control Program	Yes	Use JMX supported provided by WSAdmin. For more information, see The wsadmin administrative tool in the *Administration* topic.

API and Specifications for version 4.x

If your existing applications currently support different specification levels than are supported by this version of the product, it is likely 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 5.

The following table summarize potential migration areas due to changes in supported specifications:

Specification	Support in Version 4.0.x	Support in Version 5	Must migrate from Version 4.0.x?	Details
Enterprise beans	EJB 1.1	EJB 2.0	No	Full support for EJB 1.1 is provided. For more information, see Migrate enterprise bean code to the supported specification in the *Application Development* topic.

Specification	Support in Version 4.0.x	Support in Version 5	Must migrate from Version 4.0.x?	Details
JavaServer Pages	JSP 1.1	JSP 1.2	No	JSP 1.1 APIs are a pure subset of JSP 1.2. For more information, see Migrate Web applications in the *Application Development* topic.
Servlets	Servlet 2.2	Servlet 2.3	No	Servlet 2.2 APIs are a pure subset of Servlet 2.3. For more information, see Migrate Web applications in the *Application Development* topic.

The following table summarize potential migration areas due to changes in supported APIs:

API	Must migrate from Version 4.0.x?	Details
Java 2 Connectors	Yes	The preliminary Java2 Connector support in Version 4 has been completed in Version 5. Some changes may be necessary to take full advantage of this support. For more information, see "Migrate J2EE Connector Architecture" on page 18.
Security	Yes	Changes may be required due to J2EE security. For more information, see Migrate security configurations from previous releases in the *Securty* topic.
Sessions	Yes	Many applications can run unchanged in Version 5, although changes may be required or recommended. For more information, see "Migrate HTTP sessions" on page 21.
Web Services	Yes	Changes may be required to migrate from Apache SOAP 2.1 that was supported in Version 4 to Apache SOAP 2.2. For more information, see Migrate web services.
XML Parser	Yes	XML4J V3.1 was supported in 4.0.x. Changes are required to move to the supported API XML4J Version 4.0.6 level.

The following table summarize potential migration areas due to changes in supported tools:

Tool	Must migrate from Version 4?	Details
XML Configuration Tool	Yes	Use JMX supported provided by WSAdmin. For more information, see The wsadmin administrative tool in the *Administration* topic.
WebSphere Control Program	Yes	Use JMX supported provided by WSAdmin. For more information, see Migrate from wscp to wsadmin in the *Administration* topic.

Migration prerequisites

Before you migrate your old version of WebSphere Application Server, verify that you meet these requirements:

- **Minimum version of source WebSphere Application Server required**
  - If you are migrating from Version 3.5.x, you must be at Version 3.5.6 or higher.
  - If you are migrating from Version 4.0.x, you must be at Version 4.0.4 or higher.
  - If you are migrating from WebSphere Application Server Express Version 5.0.x, you must be at Version 5.0.2 or higher. The Express version must match the version of WebSphere Application Server (Base) or Network Deployment that you are migrating to.

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

Version 3.5.x/Version 4.0.x:

1. Enter this command on the OS/400 command line:

   WRKLNK '/QIBM/ProdData/*product_dir*/properties/com/ibm/websphere/product.xml'

   where *product_dir* is:
   - **WebASAdv** for Version 3.5.x
   - **WebASAdv4** for Version 4.0.x Advanced Edition
   - **WebASAEs4** for Version 4.0.x Advanced Single Server Edition

2. Specify option 5 (Display) next to the product.xml file to view the contents. The number within the <version> tags show the current version you have installed.

Version 5:

1. Enter this command on the OS/400 command line:

   WRKLNK '/QIBM/ProdData/*product_dir*/properties/*version*/*product_file*

   where *product_dir* is:
   - **WebASE/ASE5** for Version 5.0.x Express
   - **WebAS5/Base** for Version 5.0.x (Base)
   - **WebAS5/ND** for Version 5.0.x Network Deployment

   where *product_file* is:
   - **BASE.product** for Version 5.0.x Express
   - **BASE.product** for Version 5.0.x (Base)
   - **ND.product** for Version 5.0.x Network Deployment

2. Specify option 5 (Display) next to the product file to view the contents. The number within the <version> tags show the current version you have installed.

If you do not meet the minimum version, obtain the latest group PTF. See WebSphere Application Server PTFs for iSeries for information on the correct group PTF for your OS/400 release level

and WebSphere Application Server Version 3.x product, WebSphere Application Server Version 4.x product, or WebSphere Application Server Express V5.0.x product.

- **WebSphere Application Server Version 5 must be installed.**
  Follow the instructions in the Installation topic for more information. WebSphere Application Server Version 5 (Base) and optionally Network Deployment must be installed on the same iSeries server as the Version 3.5, Version 4, or Version 5 Express product to be migrated. Network Deployment needs to be installed if you are migrating to that edition.
- ***ALLOBJ authority is required.**
  When you call the WasPreUpgrade and WasPostUpgrade migration tools, your user profile must have *ALLOBJ authority.

Step 2: Migrate your applications

As technology advances, particularly in the area of Java components, new WebSphere Application Server product versions advance to support and extend the most recent open specification levels. If your existing applications currently support different specification levels than are supported by Version 5, it is likely you need to update at least a few aspects of the applications to comply with the new specifications.

From Version 3.5.x to Version 5, main migration areas concern IBM extensions and the Java 2 Platform, Enterprise Edition. Little migration is necessary to move from Version 4.x to Version 5.

See these topics for instructions on how to migrate your applications:

"Application assembly in Version 5" on page 17
This topic describes the changes in application assembly in WebSphere Application Server Version 5.

Migrate enterprise bean code to the supported specification
This topic describes how to determine what, if any, migration changes are required for your enterprise bean code.

"Migrate J2EE Connector Architecture" on page 18
This topic describes how to determine what, if any, migration changes are required to migrate your JCA for full support of the implementation.

"Migrate JNDI configurations" on page 18
This topic determines what, if any, migration changes are required to migrate application components that access the Java Naming and Directory Interface (JNDI) name server.

"Migrate a Version 4 data access application to Version 5" on page 19
This topic describes how to determine what, if any, migration changes are required for your data sources.

Migrate Web applications
This topic described how to determine what, if any, migration changes are required for your Web applications.

Migrate your thin or pluggable application clients
This topic describes how to determine what, if any, migration changes are required for your application clients.

Migrate security configurations from previous releases
This topic describes how to determine what, if any, migration changes are required for your security functions.

Application assembly in Version 5

Version 3.5.x developers use the administrative console to create, edit, and view application configurations. Version 4.x and Version 5 developers use the Application Assembly Tool (AAT) to package, edit, and view J2EE applications. For more information, see Application Assembly Tool in the *Application Development* topic.

Note: Any development tool that is capable of building and packaging J2EE 1.3 applications may be used instead of the AAT to assemble applications. The WebSphere Studio Application Developer (WSAD) and WebSphere Development Studio Client (WDSC) development tools include robust tools (such as wizards and panels) that allow applications to be packaged into J2EE Web archive (WAR) and enterprise archive (EAR) files. If such tools are used to package applications, then they are used in place of the AAT.

Packaging J2EE applications includes:
• Copying appropriate files into the enterprise archive (EAR) file, including classes, JSP files, HTML, and image files.
• Defining deployment descriptor files for modules and applications.

In Version 5, the Application Assembly Tool supports both steps by enabling users to copy files with appropriate relative paths into the archive, as well as by providing a GUI method for defining deployment descriptors. Developers also can set environment-specific binding information through the AAT. These bindings are defaults for the administrator to use when installing the application through the administrative console.

You can define IBM extensions to the J2EE specification, such as to allow servlets to be served by class name. To ensure portability to other application servers, these extensions are saved in an XML file that is separate from the standard J2EE deployment descriptor.

Migration of APIs and specifications means moving to the current Java component level and to other technologies that IBM WebSphere Application Server, Version 5 supports. Migration of APIs and specifications also includes moving to the most recent open specification levels.

IBM WebSphere Application Server supports a wide variety of technologies to build enterprise applications. As technology advances, particularly in the area of Java components, new application server product versions advance to support and extend the most recent open specification levels.

Migrate J2EE Connector Architecture

Version 4 Advanced Edition provided an initial implementation of the J2EE Connector Architecture (JCA) specification, Version 1.0. This implementation provided basic run time support based on the final JCA 1.0 Specification, but it was not a complete implementation.

Version 5 now provides a complete implementation of the JCA 1.0 Specification, which supports:
* Connection sharing (*res-sharing-scope*).
* Get/use/close programming model for connection handles.
* Get/use/cache programming model for connection handles.
* XA, *Local*, and *No Transaction* models of resource adapters, including XA recovery.
* EJB container caching options A and C per the specification.

If you move from one of the earlier implementations of the J2EE Connector Architecture to the current implementation, be aware of the following:
* This version supports the *res-sharing-scope* tag within the resource reference (resource-ref) element. This tag was not available in previous versions and defaulted to *shareable* connections. Version 5 supports **both** shareable and unshareable connections.
* The current product supports the Web container. Both enterprise bean and Web components can utilize the J2EE Connector Architecture.
* Both connection handle usage patterns (get/use/close and get/use/cache) are supported. The get/use/close pattern indicates that a connection is retrieved, used, and closed all within the same transaction or method boundary. The get/use/cache pattern indicates that you can cache a connection across transaction or method boundaries.
* The current version supports additional authentication mechanisms. The capability to support Options A and C per the JCA specification is provided, as well as support for *res-auth* settings of either *Application* or *Container*. In previous versions, the *res-auth* setting was basically ignored, therefore it was treated as if *res-auth* was set to *Application*. If your existing applications had *res-auth* set to *Container*, they might behave differently if you install them into a current environment without any changes.
* You can no longer specify pool and subpool names. The pool name is based on the data source or connection factory's Java Naming and Directory Interface (JNDI) name. Subpools were eliminated to provide better performance.

For applications that use WebServices and JCA Connectors, there are some additional points to be aware of:
* Applications generated on WebSphere Studio Application Developer — Integration Edition Version 4.1.1 can run on WebSphere Application Server Version 4.
* Applications generated on WebSphere Studio Application Developer — Integration Edition Version 5.0 can run on WebSphere Application Server Version 5.
* Applications generated on WebSphere Studio Application Developer — Integration Edition Version 4.1.1 can only run on WebSphere Application Server Version 5 if the applications are regenerated using WebSphere Studio Application Developer — Integration Edition Version 5.0 tools.

This limitation is because of the wsd14j.jar file. The wsd14j.jar file shipped with WebSphere Application Server Version 5 is compliant. However, because most of the classes have the same package names and interfaces, but not all, the two wsd14j.jar files cannot co-exist in the same WebSphere Application Server installation.

Migrate JNDI configurations

A new InitialContext object must be instantiated before a resource can be looked up in the Java Naming and Directory Interface (JNDI) name server. In WebSphere Application Server Version 4.x, you can set the java.naming.provider.url or Context.PROVIDER_URL property for the IntialContext object to `iiop:///` for application components (such as servlets or enterprise beans) that run on the server. For example:

```
        Properties props = new Properties();

        props.put("java.naming.provider.url", "iiop:///");
        // or
        props.put(Context.PROVIDER_URL, "iiop:///");

        ctx = new IntialContext(props);
```

The way the `iiop:///` provider URL is resolved has changed in WebSphere Application Server Version 5. In Version 4.x, this provider URL resolves to the bootstrap host and port for the application server where the application component runs. In Version 5, this provider URL resolves to the server that runs on the local host (port 2809). If you specify `iiop:///` for the provider URL, it may not resolve to the correct name server. The following outlines the resolution criteria:

- For Version 5 (Base), the `iiop:///` provider URL correctly resolves if the application server runs in the default instance and the default ports have not been changed. The provider URL does not resolve to the correct name server if you use a non-default instance or use an instance where the default ports are changed.

- For Version 5 Network Deployment, the `iiop:///` provider URL resolves to the node agent for a federated default instance, if the default instance is federated and the default ports are in use.

To correct problems that may occur, do not specify the java.naming.provider.url or Context.PROVIDER_URL properties when creating an InitialContext() object from application components that run on the server. For example:

```
        ctx = new IntialContext();
```

The provider URL is then automatically resolved to the name server that runs in the local application server process.

Migrate a Version 4 data access application to Version 5

When the WebSphere Application Server migration tools migrate Version 4 applications to Version 5 applications, the resulting application is packaged as a J2EE 1.2 application. The Version 4 datasources are migrated, because the connection manager is used with the Version 5 runtime to support these applications.

To use the connection management infrastructure in WebSphere Application Server Version 5 and the J2C connection manager, you must package your application as a J2EE 1.3 application. Your Version 4 datasources do not work and must be replaced by Version 5 datasources, which are dependent on new Version 5 JDBC resource providers. This re-packaging requires that the Version 4 applications be re-installed as a J2EE 1.3 application and all components in that application must match the required specifications for servlets and EJBs.

Perform these steps to repackage your Web modules to the Servlet 2.3 Specification and your EJB modules to the EJB 2.0 Specification, create the JDBC provider and data source, and install the applications.

Convert a Servlet 2.2 Specification Web module to a Servlet 2.3 Specification Web module

Use the following steps to migrate each of your Web modules.

1. Open the Application Assembly Tool.
2. Create a new Web module by selecting **File —> New —> Web Module**.
3. Add any required class files to the new module.
   a. Expand **Files**.
   b. Right-click **Class Files**, and select **Add Files**.
   c. In the Add Files window click **Browse**.

d. Navigate to your Version 4 EAR file, and click **Select**.
   e. In the upper left pane of the Add Files window, navigate to your WAR file, and expand the **WEB-INF —> classes** directories.
   f. Select each of the directories and files in the classes directory and click **Add**.
   g. When you have added all of the required class files, click **OK**.
4. Add any required *JAR* files to the new module.
   a. Expand the **Files** portion of the tree.
   b. Right click **Jar Files** and select **Add Files**.
   c. Navigate to your Version 4 EAR file and click **Select**.
   d. In the upper left pane of the Add Files window, navigate to your WAR file and expand the **WEB-INF** and **lib** directories.
   e. Select each JAR file and click **Add**.
   f. When you have added all of the required JAR files, click**OK**.
5. Add any required resource files, such as HTML files, JSP files, GIFs, and so on, to the new module.
   a. Expand the **Files** portion of the tree.
   b. Right click **Resource Files** and select **Add Files**.
   c. Navigate to your Version 4 EAR file and click **Select**.
   d. In the upper left pane of the Add Files window, navigate to your WAR file.
   e. Select each of the directories and files in the WAR, excluding META-INF and WEB-INF, and click **Add**.
   f. When you have added all of the required resource files, click**OK**.
6. Import your Web Components.
   a. Right click on **Web Components** and select **Import**.
   b. In the Import Components window, click **Browse**.
   c. Navigate to your Version 4 EAR file and click **Open**.
   d. In the left top pane of the **Import Components** window highlight the WAR file you are migrating.
   e. Highlight each of the components displayed in the right top pane and click **Add**.
   f. When all of your Web components show up in the Selected Components pane of the window, click **OK**.
   g. Verify that your Web components are correctly imported under the Web Components section of your new Web module.
7. Add servlet mappings for each of your Web components.
   a. Right click **Servlet Mappings** and select **New**.
   b. Fill in a URL pattern for the Web component.
   c. Select the Web component from the Servlet drop down box.
   d. Click **OK**.
8. Add any necessary resource references by following the instructions in Creating or changing a resource reference 🔵 .
9. Add any other Web module properties that are required. Click **Help** for a description of the settings.
10. **Save** the Web module.

Converting an EJB 1.1 module to an EJB 2.0 module

Use the following steps to migrate each of your EJB modules.
1. Open the Application Assembly Tool.
2. Create a new EJB Module by selecting **File —> New —> EJB Module**.

3. Add any required class files to the new module.
   a. Right click **Files object** and select **Add Files**.
   b. In the Add Files window click **Browse**.
   c. Navigate to your Version 4 EAR file and click **Select**.
   d. In the upper left pane of the Add Files window, navigate to your EJB JAR file.
   e. Select each of the directories and class files and click **Add**.
   f. When you have added all of the required class files, click **OK**.
4. Create your session beans and entity beans. To find help on this subject, Migrate enterprise bean code to the supported specification in the *Application Development* topic, the documentation for WebSphere Studio Application Developer, or the documentation for WebSphere Studio Application Developer Integration Edition.
5. Add any necessary resource references by following the instructions in Creating or changing a resource reference ❓ .
6. Add any other EJB module properties that are required. Click **Help** for a description of the settings.
7. **Save** the EJB module.
8. Generate the deployed code for the EJB by selecting **File —> Generate Code for Deployment**.
9. Fill in the appropriate fields and click **Generate Now**.

Add the EJB Modules and Web Modules to an EAR file
1. Open the Application Assembly Tool.
2. Create a new Application by selecting **File —> New —> Application**.
3. Add each of your EJB modules.
   a. Right click **EJB Modules** and select **Import**.
   b. Navigate to your converted EJB Module and click **Open**.
   c. Click **OK**.
4. Add each of your Web modules.
   a. Right click **Web Modules** and select **Import**.
   b. Navigate to your converted Web module and click **Open**.
   c. Fill in a **Context root** and click **OK**.
5. Fill in any other application properties. Click **Help** for a description of the settings.
6. Save the *EAR* file.

Install the application in WebSphere Application Server
1. Create a JDBC provider and a Version 5 data source object following the instructions in JDBC resources in the *Administration* topic.
2. Install the application, following the instructions in Install and uninstall applications in the *Administration* topic and bind the resource references to the data source you created.

Migrate HTTP sessions

Note: In Version 5 default write frequency mode is TIME_BASED_WRITES, which is different from Version 4 and 3.5 default mode of END_OF_SERVICE.

Migrate from Version 4
No programmatic changes are required to migrate from Version 4 to Version 5.

Migrate from Version 3.5
If you have Version 3.5 applications running in Servlet 2.1 mode, some of the following Version 5 differences might influence how you choose to track and manage sessions.

1. During application development, modify session-related APIs as needed. Some API changes are required in order to redeploy existing applications on Version 5. These include changes to the HttpSession API itself as well as issues associated with moving to support for the Servlet 2.3 specification. Certain Servlet 2.1 API methods have been deprecated in Servlet 2.3 API. These deprecated APIs still work in Version 5, but they may be removed in a future version of the API. Changes are summarized in the following list:
   - Replace instances of getValue() with getAttribute()
   - Replace instances of getValueNames() with getAttributeNames()
   - Replace instances of removeValue() with removeAttribute()
   - Replace instances of putValue() with setAttribute()

2. During application development, modify Web application behavior as needed. In accordance with the Servlet 2.3 specification, HttpSession objects must be scoped within a single Web application context; they may not be shared between contexts. This means that a session can no longer span Web applications. Objects added to a session by a servlet or JSP in one Web application cannot be accessed from another Web application. The same session ID may be shared (because the same cookie is in use), but each Web application has a unique session associated with the session ID. Version 5 provides a feature that can be used to extend scope of a session to enterprise application.

3. Use administrative tools to configure Session Manager security settings as needed. Relative to session security, the default Session Manager setting for Integrate Security is now false. This is different from the default setting in some earlier releases.

4. Use administrative tools to configure the JSP enabler and application server as needed. In Version 3.5 of the product, JSP files that contained the usebean tag with scope set to session did not always work properly when session persistence was enabled. Specifically, the JSP writer needed to write a scriplet to explicitly set the attribute (that is, to call setAttribute()) if it was changed as part of JSP processing.

   Two new features in Version 5 help address this problem:
   - You can set DoSetAttribute to true on the JSP InitParameter.
   - You can set the Write Contents option to Write all.

   The differences between the two solutions are summarized in the following table:

	Applies to	Configured at	Action
DoSetAttribute set to *true*	JSP	JSP enabler	Assures that JSP session-scoped beans always call setAttribute()
Write Contents option set to *Write all*	servlet or JSP	application server	All session data (changed or unchanged) is written to the external location

   If session persistence is enabled and a class reload for the Web application occurs, the sessions associated with the Web application are maintained in the persistent store and are available after the reload.

Migrate JMS applications

Both WebSphere Application Server Version 4.0.x and Version 5 support the Java Message Service (JMS) V1.0.2 specification. Version 4.0.x applications written to this specification do not require modification in order to run in Version 5.

Version 5 JMS migration involves migrating the JMS resources such as connection factories, queues, and topics used by such applications. Three different JMS providers are available in Version 5:
- Generic JMS provider
- Embedded WebSphere JMS provider
- WebSphere MQ JMS provider

See Administer JMS resources in the *Administration* topic for more information on these Version 5 JMS providers.

By default, all JMS resources from Version 4.0.x are mapped to generic JMS provider resources by the Version 5 migration tools. For more information on generic JMS providers, see Administer generic JMS providers in the *Administration* topic. If the WebSphere MQ or MQSeries products were used as your back-end message server in Version 4.0.x, then you must ensure that you have the versions supported by Version 5 before you migrate:

- WebSphere MQ for iSeries V5.3 (5724B41)
- WebSphere MQ classes for Java and JMS (5639C34)

 Note: Earlier versions of these products (e.g. MQSeries for iSeries V5.2) are not supported by Version 5. Additionally, you must possess a full license for these products if you plan to continue using these products with the generic JMS provider in Version 5.

The WebSphere MQ JMS provider provides better integration with the WebSphere MQ product. If you are using this product as your back-end message server, then you should consider remapping your JMS resources from the generic JMS provider to the WebSphere MQ JMS provider. For more information on the WebSphere MQ JMS provider and how to configure resources for it, see Administer resources for the WebSphere MQ JMS provider in the *Administration* topic.

For new JMS applications which use WebSphere MQ that is installed on the same machine where WAS Version 5 in running, you should also consider using the embedded WebSphere JMS provider. For more information, see Administer resources for the embedded WebSphere JMS provider in the *Administration* topic.

Enterprise applications that use message beans with the JMS Listener in WebSphere Application Server Enterprise Edition Version 4.0.x must be migrated to use EJB 2.0 message-driven beans. See Migrate a JMS listener application to use message-driven beans in the *Application development* topic for more information.

Note: WebSphere Application Server Enterprise Edition Version 4.0.x is not available on iSeries. These instructions are provided for customers who are migrating such applications from Enterprise Edition Version 4.0.x running on a workstation platform to Version 5 running on iSeries.

Step 3: Migrate your WebSphere Application Server instances

After you have migrated your applications, you need to migrate your instance configurations. Chose a topic below for step-by-step instructions for migrating from your old version of WebSphere Application Server to a new or default Version 5 or Network Deployment instance.

Migrate from WebSphere Application Server Version 3.5

Migration of WebSphere Application Server Version 3.5 (Standard or Advanced Edition) to WebSphere Application Server Version 5 requires a product level of Version 3.5.6 or later.

See "Migration prerequisites" on page 15 to determine the currently installed product level of WebSphere Applicaiton Server Version 3.5.

Select the appropriate option to obtain instructions on how to migrate to another version of WebSphere Application Server:

> **"Migrate to WebSphere Application Server Version 5 default instance"**
> This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server Version 3.5.6 (or later) instance to a WebSphere Application Server Version 5 default instance.
>
> **"Migrate to WebSphere Application Server Version 5 new instance" on page 26**
> This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server Version 3.5.6 (or later) instance to a WebSphere Application Server Version 5 new instance.
>
> **"Migrate to WebSphere Application Server Version 5, Network Deployment new instance" on page 28**
> This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server Version 3.5.6 (or later) instance to a WebSphere Application Server Version 5, Network Deployment new instance.

Migrate to WebSphere Application Server Version 5 default instance

Tools for migrating administrative configurations are provided for Versions 3.5 and later. This support enables either edition of Version 3.5.6 (or later) to be upgraded to Version 5.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:
* Start the Version 3.5 instance being migrated (page 24).
* Save the Version 3.5 configuration (page 24).
* Restore the Version 3.5 configuration into a Version 5 instance (page 25).
* Start the Version 5 default instance (page 26).

Start the Version 3.5 instance being migrated.

1. Enter the following command from an OS/400 command line:
   ```
   STRQSH
   ```
   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the strwasinst script with the following parameters:
   ```
   /QIBM/ProdData/WebASAdv/bin/strwasinst -instance 3.5.xInstanceName
   ```
   where *3.5.xInstanceName* is the name of the Version 3.5.x instance that is being migrated. You must wait for the administrative server to start successfully before continuing.

Save the Version 3.5 configuration.

If you have servlets and/or JSPs in the Version 3.5.x **default_app** Web application, the WASPreUpgrade migration tool does not migrate them. If you wish to have these servlets and JSPs migrated, you must move them into a different Web Application before calling WASPreUpgrade. See Issues concerning the migration of JSPs and Servlets in the *Application Development* topic for more information.

To save the Version 3.5 configuration, perform the following steps:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the Version 5 WasPreUpgrade script as shown below:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   /backup/myBackupDirectory
   /QIBM/UserData/WebAsAdv/3.5.xInstanceName
   adminNodeName
   -nameServiceHost adminNodeName
   -nameServicePort port_number
   ```

   where:

   - */backup/myBackupDirectory* (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 WasPreUpgrade.

   - /QIBM/UserData/WebAsAdv/*3.5.xInstanceName* (required parameter) is the fully qualified path of the Version 3.5.x administrative instance being migrated.

   - *adminNodeName* (required parameter) is the name of the administration node for the Version 3.5.x instance. Generally, this is the iSeries host name from **CFGTCP** option 12. The WASPreupgrade tool invokes the Version 3.5 XMLConfig tool using this parameter.

   - The -nameServiceHost and -nameServicePort parameters are also passed to XMLConfig. They are needed to override the default host name and port number used by XMLConfig and are required parameters when the Version 3.5.x instance being migrated is not the default instance. The value for -nameServiceHost is the TCP/IP host name of the iSeries server. The value for the -nameServicePort is the bootstrap port for the Version 3.5.x administrative server. The default bootstrap port is 900.

   For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade script in the *Administration* topic.

Restore the Version 3.5 configuration into a Version 5 instance

Perform the following steps to restore the Version 3.5 configuration into the Version 5 default instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
   backupDirectoryName
   [-instance 5_instance_name]
   [-portBlock port_starting_number]
   ```

   The first argument is required. Supported arguments include:

   - **backupDirectory**
     Required 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. The WASPreUpgrade tool creates this directory if it does not already exist. This parameter is equivalent to the -W migrationInformationPanelBean.migrationBackupDir="/tmp/migrationbackup" parameter in the silent installation options response file.

   - **-portBlock**
     This is an optional parameter. The value portblock specifies the first number of a block of port numbers that your instance uses. Specify the first port in a group of unused ports on your iSeries server. You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used. This parameter is case sensitive.

Note: Although this is an optional parameter, it is recommended that you always specify the parameter (or the specific port parameters, described below) if you do not want your instance's ports to conflict with the default instance's ports.

For the -portBlock parameter, the script checks instances of WebSphere Application Server Version 5 and WebSphere Application Server Version 5, Network Deployment. The script is not able to detect port usage by other applications, including previous versions of WebSphere Application Server.

For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 default instance

Perform the following steps to start the Version 5 default instance:

1. If the QEJBAS5 subsystem has not been started, start the default instance by entering the following command on the OS/400 command line:

   `STRSBS QEJBAS5/QEJBAS5`

2. If the QEJBAS5 subsystem is already started, and the SERVER1 job is present, stop the server and start it again.

3. If the QEJBAS5 subsystem is already started, but the SERVER1 job is not present, start the server using the startServer script:

   a. Enter the following command from an OS/400 command line:

      `STRQSH`

      This starts the Qshell environment so that you can run WebSphere Application Server scripts.

   b. Enter the startServer command at the Qshell command prompt.

      `/QIBM/ProdData/WebAS5/Base/bin/startServer`

Migrate to WebSphere Application Server Version 5 new instance

Tools for migrating administrative configurations are provided for Versions 3.5 and later. This support enables either edition of Version 3.5.6 (or later) to be upgraded to Version 5.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows.
- Create a Version 5 instance (page 26).
- Start the Version 3.5 instance that is being migrated (page 27).
- Save the Version 3.5 configuration (page 27).
- Restore the Version 3.5 configuration into a Version 5 instance (page 28).
- Start the Version 5 instance (page 28).

Create a Version 5 instance

Perform the following steps to create a WebSphere Application Server Version 5 instance:

1. Enter the following command from an OS/400 command line:

   `STRQSH`

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the crtwaswinst script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/crtwasinst
   -instance 50InstanceName
   -portblock starting_port_number
   -nodefaultapps
   -server 35xApplicationServerName
   ```

where *50InstanceName* is the name of your WebSphere Application Server Version 5 instance, *starting_port_number* is the first of a block of 13 consecutive ports, and *35xApplicationServerName* is the name of your WebSphere Application Server Version 3.5.x.

For details on the syntax and parameters of the crtwasinst script, see Create a new instance in the *Administration* topic.

Start the Version 3.5 instance that is being migrated

Perform the following steps to start the WebSphere Application Server Version 3.5 instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the strwasinst script with the following parameters:

   /QIBM/ProdData/WebASAdv/bin/strwasinst -instance *3.5.xInstanceName*

   where *3.5.xInstanceName* is the name of the Version 3.5.x instance that is being migrated. You must wait for the administrative server to start successfully before continuing.

Save the Version 3.5 configuration

If you have servlets or JSPs in the Version 3.5.x **default_app** Web application, the WASPreUpgrade migration tool does not migrate them. If you wish to have these servlets and JSPs migrated, you must move them into a different Web Application before calling WASPreUpgrade. See Issues concerning the migration of JSPs and Servlets in the *Application Development* topic for more information.

Perform the following steps to save the Version 3.5 configuration:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the Version 5 WasPreUpgrade script as shown below:

   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   /backup/*myBackupDirectory*
   /QIBM/UserData/WebAsAdv/*3.5.xInstanceName*
   adminNodeName
   -nameServiceHost *adminNodeName*
   -nameServicePort *port_number*

   where:

   - */backup/myBackupDirectory* (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 WasPreUpgrade.
   - /QIBM/UserData/WebAsAdv/*3.5.xInstanceName* (required parameter) is the fully qualified path of the Version 3.5.x administrative instance being migrated.
   - *adminNodeName* (required parameter) is the name of the administration node for the Version 3.5.x instance. Generally, this is the iSeries host name from **CFGTCP** option 12. The WASPreupgrade tool invokes the Version 3.5 XMLConfig tool using this parameter.
   - The -nameServiceHost and -nameServicePort parameters are also passed to XMLConfig. They are needed to override the default host name and port number used by XMLConfig and are required parameters when the Version 3.5.x instance being migrated is not the default instance. The value for -nameServiceHost is the TCP/IP host name of the iSeries server. The value for the -nameServicePort is the bootstrap port for the Version 3.5.x administrative server. The default bootstrap port is 900.

   For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade script in the *Administration* topic.

Restore the Version 3.5 configuration into a Version 5 instance

Perform the following steps to restore the Version 3.5 configuration into a Version 5 instance:

1. Enter the following command from an OS/400 command line:

   ```
   STRQSH
   ```

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
   backupDirectoryName
   [-instance 5_instance_name]
   [-portBlock port_starting_number]
   ```

   The first argument is required. Supported arguments include:

   - **backupDirectory**
     Required 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. The WASPreUpgrade tool creates this directory if it does not already exist. This parameter is equivalent to the -W migrationInformationPanelBean.migrationBackupDir="/tmp/migrationbackup" parameter in the silent installation options response file.

   - **-portBlock**
     This is an optional parameter. The value portblock specifies the first number of a block of port numbers that your instance uses. Specify the first port in a group of unused ports on your iSeries server. You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used. This parameter is case sensitive.

     Note: Although this is an optional parameter, it is recommended that you always specify the parameter (or the specific port parameters, described below) if you do not want your instance's ports to conflict with the default instance's ports.

   For the -portBlock parameter, the script checks instances of WebSphere Application Server Version 5 and WebSphere Application Server Version 5, Network Deployment. The script is not able to detect port usage by other applications, including previous versions of WebSphere Application Server.

   For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance

Perform the following steps to start the Version 5 instance:

1. Enter the following command from an OS/400 command line to start the QEJBAS5 subsystem if it is not already started:

   ```
   STRSBS QEJBAS5/QEJBAS5
   ```

2. Enter the following command from an OS/400 command line:

   ```
   STRQSH
   ```

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

3. Run the startServer script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/startServer
   -instance 50InstanceName 35xApplicationServerName
   ```

   where *50InstanceName* is the name of the Version 5 instance created in an earlier step, and *35ApplicationServerName* is the name of the Version 5 application server created in an earlier step.

Migrate to WebSphere Application Server Version 5, Network Deployment new instance

Tools for migrating administrative configurations are provided for Versions 3.5 and later. This support enables either edition of Version 3.5.6 (or later) to be upgraded to Network Deployment.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:
1. Migrate Version 3.5 to a Version 5 Base instance (page 29).
2. Create a Network Deployment instance (page 29).
3. Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance (page 29).
4. Add migrated resources to the Network Deployment instance (page 29).
5. Start the Version 5 instance that receives Version 3.5 configuration (page 30).

Migrate Version 3.5 to a Version 5 Base instance

Complete the instructions in "Migrate to WebSphere Application Server Version 5 new instance" on page 26 or "Migrate to WebSphere Application Server Version 5 default instance" on page 24. When you have finished, continue with the steps in this topic.

Create a Network Deployment instance

Perform the following steps to create a Network Deployment instance:
1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the crtnewinst script with the following parameters:

   /QIBM/ProdData/WebAS5/ND/bin/crtwasinst
   -instance *50NDInstanceName*
   -portblock *starting_port_number*

   where *50NDInstanceName* is the name of your WebSphere Application Server Version 5, Network Deployment instance, and *starting_port_number* is the first of a block of 13 consecutive ports.

   For details on the syntax and parameters of the crtwasinst script, see Create a new instance in the *Administration* topic.

Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance

Perform the following steps to add the Base instance into Network Deployment:
1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the addNode script as follows:

   /QIBM/ProdData/WebAS5/Base/bin/addNode
    -instance baseInstance *host soapPort* -includeapps

   where *host* is the host name of the system the Network Deployment instance is running on and *soapPort* is the SOAP port for the Network Deployment instance. If you have already used addNode to add the default instance to a Network Deployment instance, specify -startingport *portValue*, where *portValue* is the first port in a block of 14 unused ports.

   For details on the syntax and parameters of the addNode script, see The addNode script in the *Administration* topic.

Add migrated resources to the Network Deployment instance

Perform the following steps to populate the Network Deployment instance with the migrated resources such as JDBC providers and data sources.:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the Network Deployment Version 5 WASPostUpgrade script as follows:

   ```
   /QIBM/ProdData/WebAS5/ND/bin/WASPostUpgrade
   backupDirectoryName
   [-instance 5_instance_name]
   [-cellName nd_cell_name]
   [-nodeName base_node_name]
   ```

   The first argument is required. Supported arguments include:

   - **backupDirectoryName**
     Required 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. The WASPreUpgrade tool creates this directory if it does not already exist.

   - **-cellName**
     Optional parameter to specify the cell name for the program to update. If not specified, the program inspects the configuration for cell names. When one cell name exists, the program uses it. Otherwise, the program returns an error.

   - **-nodeName**
     Optional parameter to specify the node name for the program to update. If not specified, the program inspects the configuration for node names. When one node name exists, the program uses it. Otherwise, the program returns an error.

   For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance that receives Version 3.5 configuration

Perform the following steps to start the Version 5 instance that receives Version 3.5 configuration:

1. Start the Network Deployment instance if it is not already started.

2. Start the Network Deployment administrative console. For more information, see Start the administrative console for the Deployment Manager in the *Installation* topic.

3. Navigate to **System Administration —> Node agents**.

4. Verify that the node agent for the Base instance you added above is started. For more information, see Verify that the WebSphere Application Server node agent has started in the *Installation* topic.

5. If it is not, use the following script to start it:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/startNode -instance 50baseInstance
   ```

6. Once started successfully, start application server from Network Deployment administrative console. Navigate to **Server —> Application Servers —> *yourServer***, and then click **Start**.

Migrate from WebSphere Application Server Version 4

Migration of WebSphere Application Server Version 4 (Advanced Edition or Advanced Single Server Edition) to WebSphere Application Server Version 5 requires a product level of Version 4.0.4 or later.

This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server Version 4.0.4 (or later) Advanced Single Server Edition instance to a WebSphere Application Server Version 5 instance.

Migrate Version 4 Advanced Edition to WebSphere Application Server Version 5

Migration of WebSphere Application Server Version 4 Advanced Edition to WebSphere Application Server Version 5 or WebSphere Application Server Version 5, Network Deployment requires a product level of version 4.0.4 or later.

See "Migration prerequisites" on page 15 to determine the currently installed product level of WebSphere Application Server Version 4.0.

Select the appropriate option to obtain instructions on how to migrate to another version of WebSphere Application Server:

"Migrate to WebSphere Application Server Version 5 default instance"
This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server Version 4.0.4 (or later) instance to a WebSphere Application Server Version 5 default instance.

"Migrate to WebSphere Application Server Version 5 new instance" on page 33
This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server Version 4.0.4 (or later) instance to a WebSphere Application Server Version 5 new instance.

"Migrate to WebSphere Application Server Version 5, Network Deployment new instance" on page 35
This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server Version 4.0.4 (or later) instance to a WebSphere Application Server Version 5, Network Deployment new instance.

Migrate to WebSphere Application Server Version 5 default instance: Tools for migrating administrative configurations are provided for Versions 4 and later. This support enables either edition of Version 4.0.4 (or later) to be upgraded to Version 5.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:
1. Start the Version 4 instance being migrated (page 31).
2. Save the Version 4 configuration (page 31).
3. Restore the Version 4 configuration into a Version 5 instance (page 32).
4. Start the Version 5 default instance (page 32).

Start the Version 4 instance being migrated

Perform the following steps to start the WebSphere Application Server Version 4 instance being migrated:
1. Enter the following command from an OS/400 command line:
   STRQSH
   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the strwasinst script with the following parameters:
   /QIBM/ProdData/WebASAdv4/bin/strwasinst -instance *4xInstanceName*
   where *4xInstanceName* is the name of the Version 4.x instance that is being migrated.

Save the Version 4 configuration

If you have servlets and/or JSPs in the Version 4.x **default_app** Web application, the WASPreUpgrade migration tool does not migrate them. If you wish to have these servlets and JSPs migrated, you must move them into a different Web Application before calling WASPreUpgrade. See Issues concerning the migration of JSPs and Servlets in the *Application Development* topic for more information.

Perform the following steps to save the WebSphere Application Server Version 4 configuration:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPreUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   /backup/myBackupDirectory
   /QIBM/UserData/WebAsAdv4/4.xInstanceName
   adminNodeName
   -nameServiceHost adminNodeName
   -nameServicePort portNumber
   ```

   where:

   - */backup/myBackupDirectory* (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 WasPreUpgrade.

   - /QIBM/UserData/WebAsAdv4/*4.xInstanceName* (required parameter) is the fully qualified path of the Version 4.x administrative instance being migrated.

   - *adminNodeName* (required parameter) is the name of the administration node or server name for the Version 4 installed product. XMLConfig is called using this parameter.

   - -nameServiceHost and -nameServicePort parameter are passed to XMLConfig. They are needed to override the default host name and port number used by XMLConfig and are required parameters when the Version 4 instance being migrated is not the default instance. The value of -nameServiceHost *adminNodeName* is the name of the administration node or server name. The value of -nameServicePort *port number* must be bootstrapPort number of the Version 4 administration instance being migrated.

   For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade script in the *Administration* topic.

Restore the Version 4 configuration into a Version 5 instance

Perform the following steps to restore the Version 4 configuration into a Version 5 instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
   /backup/myBackupDirectory
   ```

   where /backup/*myBackupDirectory* (required parameter) is the fully qualified path to the integrated file system directory where the the WasPreUpgrade migration tool had previously been used to save the Version 4 administration instance configuration.

   For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 default instance

1. Enter the following command from an OS/400 command line:

   STRQSH

This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. If the QEJBAS5 subsystem has not been started, start the default instance by entering the following command on an OS/400 command line:

```
STRSBS QEJBAS5/QEJBAS5
```

3. If the QEJBAS5 subsystem is already started, and the SERVER1 job is present, stop the server and start it again.

4. If the QEJBAS5 subsystem is already started, but the SERVER1 job is not present, start the server using the startServer script. Enter the following commands on an OS/400 command line:

```
STRQSH

/QIBM/ProdData/WebAS5/Base/bin/startServer
```

Migrate to WebSphere Application Server Version 5 new instance: Tools for migrating administrative configurations are provided for Versions 4 and later. This support enables either edition of Version 4.0.4 (or later) to be upgraded to Version 5.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:

1. Create a Version 5 instance to receive the Version 4 configuration (page 33).
2. Start the Version 4 instance being migrated (page 33).
3. Save the Version 4 configuration (page 33).
4. Restore the Version 4 configuration into a Version 5 instance (page 34).
5. Start the Version 5 instance that receives the Version 4 configuration (page 35).

Create a Version 5 instance to receive the Version 4 configuration

1. Enter the following command from an OS/400 command line:

```
STRQSH
```

This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the crtwasinst script with the following parameters:

```
/QIBM/ProdData/WebAS5/Base/bin/crtwasinst
-instance 50InstanceName
-portblock starting_port_number
-nodefaultapps
-server 4xApplicationServerName
```

where *50InstanceName* is the name of your WebSphere Application Server Version 5 instance, *starting_port_number* is the first of a block of 13 consecutive ports, and *4xApplicationServerName* is the name of your WebSphere Application Server Version 4.x.

For details on the syntax and parameters of the crtwasinst script, see Create a new instance in the *Administration* topic.

Start the Version 4 instance being migrated

Perform the following steps to start the Version 4 instance being migrated:

1. Enter the following command from an OS/400 command line:

```
STRQSH
```

This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the strwasinst script with the following parameters:

```
/QIBM/ProdData/WebASAdv4/bin/strwasinst -instance 4.xInstanceName
```

where *4.xInstanceName* is the name of the Version 4.x instance that is being migrated.

Save the Version 4 configuration

If you have servlets and/or JSPs in the Version 4.x **default_app** Web application, the WASPreUpgrade migration tool does not migrate them. If you wish to have these servlets and JSPs migrated, you must move them into a different Web Application before calling WASPreUpgrade. See Issues concerning the migration of JSPs and Servlets in the *Application Development* topic for more information.

Perform the following steps to save the Version 4 configuration:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPreUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   /backup/myBackupDirectory
   /QIBM/UserData/WebAsAdv4/4.xInstanceName
   adminNodeName
   -nameServiceHost adminNodeName
   -nameServicePort port_number
   ```

   where:

   - */backup/myBackupDirectory* (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 WasPreUpgrade.

   - */QIBM/UserData/WebAsAdv4/4.xInstanceName* (required parameter) is the fully qualified path of the Version 4 administrative instance being migrated.

   - *adminNodeName* (required parameter) is the name of the administration node or server name for the Version 4 installed product. XMLConfig is called using this parameter.\

   - -nameServiceHost and -nameServicePort parameters are passed to XMLConfig. They are needed to override the default host name and port number used by XMLConfig and are required parameters when the Version 4 instance being migrated is not the default instance. The value of -nameServiceHost *adminNodeName* is the name of the administration node or server name. The value of -nameServicePort *port_number* must be the bootstrapPort number of the Version 4 administration instance being migrated.

   For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade script in the *Administration* topic.

Restore the Version 4 configuration into a Version 5 instance

Perform the following steps to restore the Version 4 configuration into a Version 5 instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
   backupDirectoryName
    [-instance 5_instance_name]
    [-portBlock port_starting_number]
   ```

   The first argument is required. Supported arguments include:

   - **backupDirectory**
     Required 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. The WASPreUpgrade tool creates this directory if it does not already exist. This parameter is equivalent to the -W migrationInformationPanelBean.migrationBackupDir="/tmp/migrationbackup" parameter in the silent installation options response file.

- **-portblock**

  This is an optional parameter. The value portblock specifies the first number of a block of port numbers that your instance uses. Specify the first port in a group of unused ports on your iSeries server. You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used. This parameter is case sensitive.

  Notes:

  – Although this is an optional parameter, it is recommended that you always specify the parameter (or the specific port parameters, described below) if you do not want your instance's ports to conflict with the default instance's ports.

  – A WebSphere Application Server instance uses several ports for various functions. When you create a new instance, ports are assigned based on the following ordered conditions:

    - Specific port parameters
      If you specify values for specific port parameters, the script uses those values. Specific port parameters are -inthttp (Base application server), -admin, -jmsqueued (Base application server), -jmsdirect (Base application server), -jmssecure (Base application server), -soap, -orblistener (Network Deployment), -nameservice, -drsclient, and -celldiscovery (Network Deployment).

    - The -portblock parameter
      Services for which you have not specified a port number are assigned ports sequentially starting with the value of the -portblock parameter. If a script encounters a port that is specified by another parameter in the script, it skips that port number and continues with the next unused port.

    - Default values
      If -portblock is not specified, any services for which you have not specified a port parameter are assigned the default ports. See the specific port parameters for the default port numbers.

  For the -portblock parameter, the script checks only a master index of all instances of WebSphere Application Server Version 5 and WebSphere Application Server Version 5, Network Deployment. The script is not able to detect port usage by other applications, including previous versions of WebSphere Application Server.

  For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance that receives the Version 4 configuration

Perform the following steps to start the Version 5 instance that receives the Version 4 configuration:

1. Start the QEJBAS5 subsystem if it is not already started. Enter this command from an OS/400 command line:

   `STRSBS QEJBAS5/QEJBAS5`

2. Enter the following command from an OS/400 command line:

   `STRQSH`

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

3. Run the startServer script with the following parameters:

   `/QIBM/ProdData/WebAS5/Base/bin/startServer -instance` *50InstanceName 4xApplicationServerName*

   where *50InstanceName* is the name of the Version 5 instance created in an earlier step, and *4xApplicationServerName* is the name of the Version 5 application server created in an earlier step.

Migrate to WebSphere Application Server Version 5, Network Deployment new instance: Tools for migrating administrative configurations are provided for Versions 4 and later. This support enables either edition of Version 4.0.4 (or later) to be upgraded to Network Deployment.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:

1. Migrate Version 4 to a Version 5 Base instance (page 36).
2. Create a Network Deployment instance (page 36).
3. Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance (page 36).
4. Add migrated resources to the Network Deployment instance (page 36).
5. Start the Version 5 instance that receives the Version 4 configuration (page 37).

Migrate Version 4 to a Version 5 Base instance

Complete the instructions in "Migrate to WebSphere Application Server Version 5 new instance" on page 33. When you have finished, continue with the steps in this topic.

Create Network Deployment instance

Perform the following steps to create Network Deployment instance:
1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the crtnewinst script with the following parameters:

   /QIBM/ProdData/WebAS5/ND/bin/crtwasinst
   -instance *50NDInstanceName*
   -portblock *starting_port_number*

   where *50NDInstanceName* is the name of your WebSphere Application Server Version 5, Network Deployment instance, and *starting_port_number* is the first of a block of 13 consecutive ports.

   For details on the syntax and parameters of the crtwasinst script, see Create a new instance in the *Administration* topic.

Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance

Perform the following steps to add the Base instance to the Network Deployment instance:
1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the addNode script with the following parameters:

   /QIBM/ProdData/WebAS5/Base/bin/addNode
   -instance baseInstance *host soapPort* -includeapps

   where *host* is the host name of the system the Network Deployment instance is running on, and *soapPort* is the SOAP port for the Network Deployment instance. If you have already used addNode to add the default instance to a Network Deployment instance, specify -startingport *portValue*, where *portValue* is the first port in a block of 14 unused ports.

   For details on the syntax and parameters of the addNode script, see The addNode script in the *Administration* topic.

Add migrated resources to the Network Deployment instance

Perform the following steps to add migrated resources to the Network Deployment instance:
1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the WasPostUpgrade script with the following parameters:

```
/QIBM/ProdData/WebAS5/ND/bin/WASPostUpgrade
backupDirectoryName
 [-instance V5_instance_name]
 [-cellName nd_cell_name]
 [-nodeName base_node_name]
```

The first argument is required. Supported arguments include:

- **backupDirectoryName**
  Required 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. The
  WASPreUpgrade tool creates this directory if it does not already exist. This parameter is equivalent
  to the -W migrationInformationPanelBean.migrationBackupDir="/tmp/migrationbackup"
  parameter in the silent installation options response file.

- **-cellName**
  Optional parameter to specify the cell name for the program to update. If not specified, the
  program inspects the configuration for cell names. When one cell name exists, the program uses it.
  Otherwise, the program returns an error.

- **-nodeName**
  Optional parameter to specify the node name for the program to update. If not specified, the
  program inspects the configuration for node names. When one node name exists, the program uses
  it. Otherwise, the program returns an error.

For a full explanation of the WasPostUpgrade migration tool and parameters, see The
WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance that receives the Version 4 configuration

Perform the following steps to start the Version 5 instance that receives the Version 4 configuration:

1. Enter the following command from an OS/400 command line:

   `STRQSH`

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the strwasinst script with the following parameters:

   `/QIBM/ProdData/WebAS5/ND/bin/startServer -instance 50InstanceName server1`

   where *50InstanceName* is the name of the Version 5 instance created in an earlier step, and *server1* is
   the name of the Version 5 application server created in an earlier step.

Migrate Version 4 Advanced Single Server Edition to WebSphere Application Server Version 5

Migration of WebSphere Application Server Version 4 Advanced Single Server Edition to WebSphere
Application Server Version 5 requires a product level of version 4.0.4 level or later.

See "Migration prerequisites" on page 15 to determine the currently installed product level of WebSphere
Application Server Version 4.0.

Select the appropriate option to obtain instructions on how to migrate to another version of WebSphere
Application Server:

"Migrate to a WebSphere Application Server Version 5 default instance" on page 38
This topic contains detailed information and step-by-step instructions for migrating a Version 4.0.4
(or later) instance to a WebSphere Application Server Version 5 default instance.

"Migrate to a WebSphere Application Server Version 5 new instance" on page 39
This topic contains detailed information and step-by-step instructions for migrating a Version 4.0.4
(or later) instance to a WebSphere Application Server Version 5 new instance.

"Migrate to a WebSphere Application Server Version 5, Network Deployment new instance" on page 42
This topic contains detailed information and step-by-step instructions for migrating a Version 4.0.4 (or later) instance to a WebSphere Application Server Version 5, Network Deployment new instance.

Migrate to a WebSphere Application Server Version 5 default instance: Tools for migrating administrative configurations are provided for Versions 4 and later. This support enables either edition of Version 4.0.4 (or later) to be upgraded to Version 5.

Before performing the instructions on this page, perform the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:

1. Start the Version 4 instance being migrated (page 38).
2. Save the Version 4 configuration (page 38).
3. Restore the Version 4 configuration into a Version 5 instance (page 39).
4. Start the Version 5 default instance (page 39).

Start the Version 4 instance being migrated.

Perform the following steps to start the Version 4 instance being migrated:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the strwasinst script with the following parameters:

   /QIBM/ProdData/WebASAEs4/bin/strwasinst -instance *4.xInstanceName*

   where *4.xInstanceName* is the name of the Version 4.x instance that is being migrated.

Save the Version 4 configuration

If you have servlets or JSPs in the Version 4.x **default_app** Web application, the WASPreUpgrade migration tool does not migrate them. If you wish to have these servlets and JSPs migrated, you must move them into a different Web Application before calling WASPreUpgrade. See Issues concerning the migration of JSPs and Servlets in the *Application Development* topic for more information.

Perform the following steps to save the Version 4 configuration:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the WasPreUpgrade script with the following parameters:

   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   /backup/myBackupDirectory
   /QIBM/UserData/WebASAEs4/*4xInstanceName*

   where:

   - /backup/myBackupDirectory (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 WasPreUpgrade.
   - /QIBM/UserData/WebAsAEs4/*4.xInstanceName* (required parameter) is the fully qualified path of the Version 4.x administrative instance being migrated.

   For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade script in the *Administration* topic.

Restore the Version 4 configuration into a Version 5 instance

Perform the following steps to restore the Version 4 configuration into a Version 5 instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   /QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
   /backup/*myBackupDirectory*

   where /backup/*myBackupDirectory* (required parameter) is the fully qualified path to the integrated file system directory where the the WasPreUpgrade migration tool had previously been used to save the Version 4 instance configuration.

For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 default instance

Perform the following steps to start the Version 5 default instance:

1. If the QEJBAS5 subsystem has not been started, start the default instance by entering this command from an OS/400 command line:

   STRSBS QEJBAS5/QEJBAS5

2. If the QEJBAS5 subsystem is already started, and the SERVER1 job is present, stop the server and start it again.

3. If the QEJBAS5 subsystem is already started, but the SERVER1 job is not present, start the server using the startServer script at the Qshell prompt:

   /QIBM/ProdData/WebAS5/Base/bin/startServer

Migrate to a WebSphere Application Server Version 5 new instance: Tools for migrating administrative configurations are provided for Versions 4 and later. This support enables either edition of Version 4.0.4 (or later) to be upgraded to Version 5.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:

1. Analyze the Version 4 configuration (page 39).
2. Create a Version 5 instance to receive the Version 4 configuration (page 39).
3. Start the Version 4 instance being migrated (page 40).
4. Save the Version 4 configuration (page 40).
5. Restore the Version 4 configuration into a Version 5 instance (page 40).
6. Start the Version 5 instance that receives the Version 4 configuration (page 41)

Analyze the Version 4 configuration

Perform the following steps to analyze the Version 4 configuration:

1. Open the Version 4 console on the instance you would like to migrate. Analyze the environment, taking note of number of application servers and the names of each application server.

Create a Version 5 instance to receive the Version 4 configuration

Perform the following steps to create a Version 5 instance to receive the Version 4 configuration:

1. Enter the following command from an OS/400 command line:

```
STRQSH
```

This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the crtnewinst script with the following parameters:

```
/QIBM/ProdData/WebAS5/Base/bin/crtwasinst
-instance 50InstanceName
-portblock starting_port_number
-nodefaultapps
-server 4xApplicationServerName
```

where *50InstanceName* is the name of your WebSphere Application Server Version 5 instance, *starting_port_number* is the first of a block of 13 consecutive ports, and *4xApplicationServerName* is the name of your WebSphere Application Server Version 4.x.

For details on the syntax and parameters of the crtwasinst script, see Create a new instance in the *Administration* topic.

Start the Version 4 instance being migrated

Perform the following steps to start the Version 4 instance being migrated:

1. Enter the following command from an OS/400 command line:
   ```
   STRQSH
   ```
   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the strwasinst script with the following parameters:
   ```
   /QIBM/ProdData/WebASAes4/bin/strwasinst -instance 4xInstanceName
   ```
   where *4xInstanceName* is the name of the Version 4.x instance that is being migrated.

Save the Version 4 configuration

If you have servlets or JSPs in the Version 4.x **default_app** Web application, the WASPreUpgrade migration tool does not migrate them. If you wish to have these servlets and JSPs migrated, you must move them into a different Web Application before calling WASPreUpgrade. See Issues concerning the migration of JSPs and Servlets in the *Application Development* topic for more information.

Perform the following steps to save the Version 4 configuration:

1. Enter the following command from an OS/400 command line:
   ```
   STRQSH
   ```
   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPreUpgrade script with the following parameters:
   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   /backup/myBackupDirectory
   /QIBM/UserData/WebAsAEs4/4xInstanceName
   ```
   where:

   - */backup/myBackupDirectory*(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 WasPreUpgrade.

   - /QIBM/UserData/WebAsAEs4/*4.xInstanceName* (required parameter) is the fully qualified path of the Version 4.x administrative instance being migrated.

For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade script in the *Administration* topic.

Restore the Version 4 configuration into a Version 5 instance

1. Enter the following command from an OS/400 command line:

```
STRQSH
```
This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:
```
/QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
backupDirectoryName
[-instance V5_instance_name]
[-portBlock port_starting_number]
```
The first argument is required. Supported arguments include:

- **backupDirectory**
  Required 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. The WASPreUpgrade tool creates this directory if it does not already exist. This parameter is equivalent to the -W migrationInformationPanelBean.migrationBackupDir="/tmp/migrationbackup" parameter in the silent installation options response file.

- **-portblock**
  This is an optional parameter. The value portblock specifies the first number of a block of port numbers that your instance uses. Specify the first port in a group of unused ports on your iSeries server. You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used. This parameter is case sensitive.

  Notes:

  – Although this is an optional parameter, it is recommended that you always specify the parameter (or the specific port parameters, described below) if you do not want your instance's ports to conflict with the default instance's ports.

  – A WebSphere Application Server instance uses several ports for various functions. When you create a new instance, ports are assigned based on the following ordered conditions:

    - Specific port parameters
      If you specify values for specific port parameters, the script uses those values. Specific port parameters are -inthttp (Base application server), -admin, -jmsqueued (Base application server), -jmsdirect (Base application server), -jmssecure (Base application server), -soap, -orblistener (Network Deployment), -nameservice, -drsclient, and -celldiscovery (Network Deployment).

    - The -portblock parameter
      Services for which you have not specified a port number are assigned ports sequentially starting with the value of the -portblock parameter. If a script encounters a port that is specified by another parameter in the script, it skips that port number and continues with the next unused port.

    - Default values
      If -portblock is not specified, any services for which you have not specified a port parameter are assigned the default ports. See the specific port parameters for the default port numbers.

  – For the -portblock parameter, the script checks only a master index of all instances of WebSphere Application Server Version 5 and WebSphere Application Server Version 5, Network Deployment. The script is not able to detect port usage by other applications, including previous versions of WebSphere Application Server.

    For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance that receives the Version 4 configuration

Perform the following steps to start the Version 5 instance:

1. Enter this command from an OS/400 command line to start the QEJBAS5 subsystem if it is not already started:
```
STRSBS QEJBAS5/QEJBAS5
```

2. Enter the following command from an OS/400 command line:

```
STRQSH
```

This starts the Qshell environment so that you can run WebSphere Application Server scripts.

3. Run the startServer script with the following parameters:

```
/QIBM/ProdData/WebAS5/Base/bin/startServer
-instance 50InstanceName 4xApplicationServerName
```

where *50InstanceName* is the name of the Version 5 instance created in an earlier step, and *4xApplicationServerName* is the name of the Version 5 application server created in an earlier step.

Migrate to a WebSphere Application Server Version 5, Network Deployment new instance: Tools for migrating administrative configurations are provided for Versions 4 and later. This support enables either edition of Version 4.0.4 (or later) to be upgraded to Network Deployment.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:
1. Migrate Version 4 to a Version 5 Base instance (page 42).
2. Create a Network Deployment instance (page 42).
3. Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance (page 42).
4. Add migrated resources to the Network Deployment instance (page 43).
5. Start the Version 5 instance that receives the Version 4 configuration (page 43).

Migrate Version 4 to a Version 5 Base instance

Complete the instructions in "Migrate to a WebSphere Application Server Version 5 new instance" on page 39. When you have finished, continue with the steps in this topic.

Create a Network Deployment instance

Perform the following steps to create a Network Deployment instance:
1. Enter the following command from an OS/400 command line:
   ```
   STRQSH
   ```
   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the crtnewinst script with the following parameters:
   ```
   /QIBM/ProdData/WebAS5/ND/bin/crtwasinst
   -instance 50NDInstanceName
   -portblock starting_port_number
   ```
   where *50NDInstanceName* is the name of your WebSphere Application Server Version 5, Network Deployment instance, and *starting_port_number* is the first of a block of 13 consecutive ports.

   For details on the syntax and parameters of the **crtwasinst** script, see Create a new instance in the *Administration* topic.

Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance.

Perform the following steps to add the Base instance to the Network Deployment instance:
1. Enter the following command from an OS/400 command line:
   ```
   STRQSH
   ```
   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the addNode script with the following parameters:
   ```
   /QIBM/ProdData/WebAS5/Base/bin/addNode
   -instance baseInstance host soapPort -includeapps
   ```

where *host* is the host name of the system the Network Deployment instance is running on, and *soapPort* is the SOAP port for the Network Deployment instance.

If you have already used addNode to add the default instance to a Network Deployment instance, specify `-startingport` *portValue* where *portValue* is the first port in a block of 14 unused ports. For details on the syntax and parameters of the addNode script, see The addNode script in the *Administration* topic.

Add migrated resources to the Network Deployment instance

Perform the following steps to add migrated resources to the Network Deployment instance:

1. Enter the following command from an OS/400 command line:

   ```
   STRQSH
   ```

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/ND/bin/WASPostUpgrade
   backupDirectoryName
   [-instance 5_instance_name]
   [-cellName nd_cell_name]
   [-nodeName base_node_name]
   ```

   The first argument is required. Supported arguments include:

   * **backupDirectoryName**
     Required 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. The WASPreUpgrade tool creates this directory if it does not already exist. This parameter is equivalent to the -W migrationInformationPanelBean.migrationBackupDir="/tmp/migrationbackup" parameter in the silent installation options response file.

   * **-cellName**
     Optional parameter to specify the cell name for the program to update. If not specified, the program inspects the configuration for cell names. When one cell name exists, the program uses it. Otherwise, the program returns an error.

   * **-nodeName**
     Optional parameter to specify the node name for the program to update. If not specified, the program inspects the configuration for node names. When one node name exists, the program uses it. Otherwise, the program returns an error.

   For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance that receives the Version 4 configuration

Perform the following steps to start the Version 5 instance:

1. Enter the following command from an OS/400 command line:

   ```
   STRQSH
   ```

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the strwasinst script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/ND/bin/startServer
   -instance 50InstanceName server1
   ```

   where *50InstanceName* is the name of the Version 5 instance created in an earlier step, and *server1* is the name of the Version 5 application server created in an earlier step.

Migrate from WebSphere Application Server - Express Version 5.0.x

Migration of WebSphere Application Server - Express Version 5 to WebSphere Application Server Version 5 or WebSphere Application Server Version 5, Network Deployment requires both products to be at the

same level, and both products must be Version 5.0.2 or later. For example, when migrating to WebSphere Application Server Version 5.0.x (Base) from WebSphere Application Server - Express Version 5.0.x, both editions must be at Version 5.0.2.

See "Migration prerequisites" on page 15 to determine the currently installed product level of WebSphere Application Server.

Select the appropriate option to obtain instructions on how to migrate to another version of WebSphere Application Server:

"Migrate to a WebSphere Application Server Version 5 default instance"
This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server - Express Version 5.0.2 (or later) instance to a WebSphere Application Server Version 5 default instance.

"Migrate to a new WebSphere Application Server Version 5 instance" on page 45
This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server - Express Version 5.0.2 (or later) instance to a new WebSphere Application Server Version 5 instance.

"Migrate to a new WebSphere Application Server Version 5, Network Deployment instance" on page 48
This topic contains detailed information and step-by-step instructions for migrating a WebSphere Application Server - Express Version 5.0.2 (or later) instance to a new WebSphere Application Server Version 5, Network Deployment instance.

Migrate to a WebSphere Application Server Version 5 default instance

Tools for migrating administrative configurations are provided for WebSphere Application Server - Express Version 5 and later. This support enables WebSphere Application Server - Express Version 5.0.2 (or later) to be upgraded to WebSphere Application Server Version 5.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:
1. Save the WebSphere Application Server - Express Version 5 configuration (page 44).
2. Restore the WebSphere Application Server - Express Version 5 configuration into a Version 5 instance (page 45).
3. Start the Version 5 default instance (page 45).

Save the WebSphere Application Server - Express Version 5 configuration

Perform the following steps to save the WebSphere Application Server - Express Version 5 configuration:
1. Enter the following command from an OS/400 command line:
   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
2. Run the WasPreUpgrade script with the following parameters:
   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   myBackupDirectory
   /QIBM/UserData/WebASE/ASE5/*5.xExpressInstanceName*

   where:
   - *myBackupDirectory* (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 WasPreUpgrade.

- /QIBM/UserData/WebASE/ASE5/*5.xExpressInstanceName* (required parameter) is the fully
  qualified path of the Version 5.x Express instance being migrated.

For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade
script in the *Administration* topic.

Restore the WebSphere Application Server - Express Version 5 configuration into a Version 5 instance

Perform the following steps to restore the WebSphere Application Server - Express Version 5
configuration into a Version 5 instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   /QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
   myBackupDirectory

   where *myBackupDirectory* (required parameter) is the fully qualified path to the integrated file system
   directory where the the WasPreUpgrade migration tool had previously been used to save the
   WebSphere Application Server - Express Version 5 instance configuration.

   For a full explanation of the WasPostUpgrade migration tool and parameters, see The
   WASPostUpgrade script in the *Administration* topic.

Start the Version 5 default instance

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. If the QEJBAS5 subsystem has not been started, start the default instance by entering the following
   command on an OS/400 command line:

   STRSBS QEJBAS5/QEJBAS5

3. If the QEJBAS5 subsystem is already started, and the SERVER1 job is present, stop the server and
   start it again.

4. If the QEJBAS5 subsystem is already started, but the SERVER1 job is not present, start the server
   using the startServer script. Enter the following commands on an OS/400 command line:

   STRQSH

   /QIBM/ProdData/WebAS5/Base/bin/startServer

Migrate to a new WebSphere Application Server Version 5 instance

Tools for migrating administrative configurations are provided for WebSphere Application Server -
Express Version 5 and later. This support enables WebSphere Application Server - Express Version 5.0.2
(or later) to be upgraded to WebSphere Application Server Version 5.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:

1. Create a Version 5 instance to receive the WebSphere Application Server - Express Version 5
   configuration (page 46).

2. Save the WebSphere Application Server - Express Version 5 configuration (page 46).

3. Restore the WebSphere Application Server - Express Version 5 configuration into a Version 5 instance
   (page 46).

4. Start the Version 5 instance that receives the WebSphere Application Server - Express Version 5
   configuration (page 47).

Create a Version 5 instance to receive the WebSphere Application Server - Express Version 5 configuration

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the crtwasinst script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/crtwasinst
   -instance 50InstanceName
   -portblock starting_port_number
   -nodefaultapps
   -server 5xExpressApplicationServerName
   ```

   where *50InstanceName* is the name of your WebSphere Application Server Version 5 instance, *starting_port_number* is the first of a block of 13 consecutive ports, and *5xExpressApplicationServerName* is the name of your WebSphere Application Server - Express Version 5.0.x application server

   For details on the syntax and parameters of the crtwasinst script, see Create a new instance in the *Administration* topic.

Save the WebSphere Application Server - Express Version 5 configuration

Perform the following steps to save the WebSphere Application Server - Express Version 5 configuration:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPreUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPreUpgrade
   myBackupDirectory
   /QIBM/UserData/WebASE/ASE5/5.xExpressInstanceName
   ```

   where:

   - *myBackupDirectory*(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 WasPreUpgrade.
   - /QIBM/UserData/WebASE/ASE5/*5.xExpressInstanceName* (required parameter) is the fully qualified path of the WebSphere Application Server - Express Version 5 instance being migrated.

   For a full explanation of the WasPreUpgrade migration tool and parameters, see The WASPreUpgrade script in the *Administration* topic.

Restore the WebSphere Application Server - Express Version 5 configuration into a Version 5 instance

Perform the following steps to restore the WebSphere Application Server - Express Version 5 configuration into a Version 5 instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/WASPostUpgrade
   myBackupDirectory
    [-instance 5_instance_name]
    [-portBlock port_starting_number]
   ```

   The first argument is required. Supported arguments include:

- **backupDirectoryName**
  where *myBackupDirectory* (required parameter) is the fully qualified path to the integrated file system directory where the the WasPreUpgrade migration tool had previously been used to save the WebSphere Application Server - Express Version 5 instance configuration.
- **-portBlock**
  This is an optional parameter. The value portblock specifies the first number of a block of port numbers that your instance uses. Specify the first port in a group of unused ports on your iSeries server. You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used.

  Notes:
  – Although this is an optional parameter, it is recommended that you always specify the parameter (or the specific port parameters, described below) if you do not want your instance's ports to conflict with the default instance's ports.
  – A WebSphere Application Server instance uses several ports for various functions. When you create a new instance, ports are assigned based on the following ordered conditions:
    - Specific port parameters
      If you specify values for specific port parameters, the script uses those values. Specific port parameters are -inthttp (Base application server), -admin, -jmsqueued (Base application server), -jmsdirect (Base application server), -jmssecure (Base application server), -soap, -orblistener (Network Deployment), -nameservice, -drsclient, and -celldiscovery (Network Deployment).
    - The -portBlock parameter
      Services for which you have not specified a port number are assigned ports sequentially starting with the value of the -portblock parameter. If a script encounters a port that is specified by another parameter in the script, it skips that port number and continues with the next unused port.
    - Default values
      If -portBlock is not specified, any services for which you have not specified a port parameter are assigned the default ports. See the specific port parameters for the default port numbers.

  For the -portBlock parameter, the script checks only a master index of all instances of WebSphere Application Server Version 5 and WebSphere Application Server Version 5, Network Deployment. The script is not able to detect port usage by other applications, including previous versions of WebSphere Application Server.

  For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance that receives the WebSphere Application Server - Express Version 5 configuration

Perform the following steps to start the Version 5 instance that receives the WebSphere Application Server - Express Version 5 configuration:

1. Start the QEJBAS5 subsystem if it is not already started. Enter this command from an OS/400 command line:
   ```
   STRSBS QEJBAS5/QEJBAS5
   ```
2. Enter the following command from an OS/400 command line:
   ```
   STRQSH
   ```
   This starts the Qshell environment so that you can run WebSphere Application Server scripts.
3. Run the startServer script with the following parameters:
   ```
   /QIBM/ProdData/WebAS5/Base/bin/startServer -instance 50InstanceName 5xExpressApplicationServerName
   ```
   where *50InstanceName* is the name of the Version 5 instance created in an earlier step, and *5xExpressApplicationServerName* is the name of the Version 5 application server created in an earlier step.

Migrate to a new WebSphere Application Server Version 5, Network Deployment instance

Tools for migrating administrative configurations are provided for WebSphere Application Server - Express Version 5 and later. This support enables either edition of WebSphere Application Server - Express Version 5.0.2 (or later) to be upgraded to Network Deployment.

Before performing the instructions on this page, do the steps in "Step 1: Plan your migration" on page 11.

A summary of the product migration process is as follows:

1. Migrate WebSphere Application Server - Express Version 5 to a Version 5 Base instance (page 48).
2. Create a Network Deployment instance (page 48).
3. Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance (page 48).
4. Add migrated resources to the Network Deployment instance (page 49).
5. Start the Version 5 instance that receives the WebSphere Application Server - Express Version 5 configuration (page 49).

Migrate WebSphere Application Server - Express Version 5 to a Version 5 Base instance

Complete the instructions in "Migrate to a new WebSphere Application Server Version 5 instance" on page 45. When you have finished, continue with the steps in this topic.

Create Network Deployment instance

Perform the following steps to create Network Deployment instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the crtnewinst script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/ND/bin/crtwasinst
   -instance 50NDInstanceName
   -portblock starting_port_number
   ```

   where *50NDInstanceName* is the name of your WebSphere Application Server Version 5, Network Deployment instance, and *starting_port_number* is the first of a block of 13 consecutive ports.

   For details on the syntax and parameters of the crtwasinst script, see Create a new instance in the *Administration* topic.

Add the WebSphere Application Server Version 5 ("Base") instance to the Network Deployment instance

Perform the following steps to add the Base instance to the Network Deployment instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the addNode script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/addNode
   -instance baseInstance host soapPort -includeapps
   ```

   where *host* is the host name of the system the Network Deployment instance is running on, and *soapPort* is the SOAP port for the Network Deployment instance. If you have already used addNode to add the default instance to a Network Deployment instance, specify -startingport *portValue*, where *portValue* is the first port in a block of 14 unused ports.

For details on the syntax and parameters of the addNode script, see The addNode script in the *Administration* topic.

Add migrated resources to the Network Deployment instance

Perform the following steps to add migrated resources to the Network Deployment instance:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the WasPostUpgrade script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/ND/bin/WASPostUpgrade
   backupDirectoryName
    [-instance V5_instance_name]
    [-cellName nd_cell_name]
    [-nodeName base_node_name]
   ```

   The first argument is required. Supported arguments include:

   - **backupDirectoryName**
     Required 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.

   - **-cellName**
     Optional parameter to specify the cell name for the program to update. If not specified, the program inspects the configuration for cell names. When one cell name exists, the program uses it. Otherwise, the program returns an error.

   - **-nodeName**
     Optional parameter to specify the node name for the program to update. If not specified, the program inspects the configuration for node names. When one node name exists, the program uses it. Otherwise, the program returns an error.

   For a full explanation of the WasPostUpgrade migration tool and parameters, see The WASPostUpgrade script in the *Administration* topic.

Start the Version 5 instance that receives the WebSphere Application Server - Express Version 5 configuration

Perform the following steps to start the Version 5 instance that receives the WebSphere Application Server - Express Version 5 configuration:

1. Enter the following command from an OS/400 command line:

   STRQSH

   This starts the Qshell environment so that you can run WebSphere Application Server scripts.

2. Run the strwasinst script with the following parameters:

   ```
   /QIBM/ProdData/WebAS5/ND/bin/startServer -instance 50InstanceName server1
   ```

   where *50InstanceName* is the name of the Version 5 instance created in an earlier step, and *server1* is the name of the Version 5 application server created in an earlier step.

Step 4: Complete the migration

This topic describes the manual steps that you might need to perform to complete the migration process. These are steps that are not performed automatically by the WebSphere Application Server migration tools.

Change and install a new EJB dbschema database name
If the EJB dbschema name was hard coded to database EJB in your Version 3.5 application, you must change and install a new EJB dbschema database. For more information, see Change and install a new EJB dbschema database name in the *Application development* topic.

Change the HTTP transport after migration

If the -portBlock parameter was not specified in the WASPostUpgrade script, you need to use the Version 5 administrative console to adjust the ports manually after migration has completed."

1. Use the Version 5 administrative console to change the HTTP transport port:

   a. Navigate to **Servers —> Application Servers —>** *my_server* where *my_server* is the name of the application server that was migrated.

   b. Click **Web Container**.

   c. Click **HTTP transports**.

   d. Click on a host.

   e. Adjust the **Port** property by entering a port not currently in use on your iSeries server. Record both the old and new port numbers; they are needed for step 2.

   f. Click **OK**.

   g. Repeat for each host.

2. Adjust your Virtual hosts to correspond to your HTTP transport.

   a. Click **Environment —> Virtual Hosts**.

   b. Select your virtual host.

   c. Click **Host Aliases**.

   d. Click **New**.

   e. Select one of the ports that were changed in step 1.

   f. Enter the port that this was changed to in step 1.

   g. Click **OK**.

   h. Repeat for each port that was changed in step 1.

3. Click **Save.**

4. Regenerate your plugin configuration.

   a. Navigate to **Environment —> Update Web Server Plugin**.

   b. Click **OK**.

Migrate the default file and directory authorities

The default for files and directories created by a WebSphere Application Server Version 4 application server is to have *PUBLIC authority set to *RX. The default for files and directories created by a WebSphere Application Server Version 5 application server is to have *PUBLIC authority set to *EXCLUDE. This default was changed in WebSphere Application Server Version 5 for security reasons. Two system properties can be added to change this WebSphere Application Server Version 5 default to the default that was used in WebSphere Application Server Version 4. The following instructions describe how to add these two system properties to an application server:

1. Open the administrative console.

2. Expand **Servers** and click **Application Servers**.

3. Click the name of the application server for which you would like to add the system properties.

4. Click **Process Definition**.

5. Click **Java Virtual Machine**.

6. Click **Custom Properties**.

7. Click **New**.

8. Specify os400.file.create.auth for the property name and RX and the property value.

9. Click **Apply**.

10. Repeat steps 7-9 for the os400.dir.create.auth property name and RX property value.

11. Save your changes.

12. Restart the application server for the changes to take effect.

Additionally, the grtwasaut script located in the /QIBM/ProdData/WebAS5/base/bin directory can be used to explicitly grant authority to the appropriate users and authorization lists for directories and files in an instance. This is an alternative to the approach described above in that grtwasaut allows fine grained file and directory permissions to be defined. See The grtwasaut script in the *Administration* topic for more information on using the grtwasaut script to change the file and directory authorities.

Migrate the web server plug-in
If the WebSphere Application Server Version 3.5.x plug-in for your web server uses Open Servlet Engine (OSE) transport, you must switch to HTTP transport when migrating to WebSphere Application Server Version 5.

The following instructions are specific to the Web server being supported and assume that you can successfully migrate existing Web applications:

Plug-in migration has been tested with the following Web server products:
- IBM HTTP Server (original) for iSeries (V4R5)
- IBM HTTP Server (powered by Apache) for iSeries (V5R1)
- Lotus Domino HTTP Server

Use the following steps to migrate the plug-in configuration:
1. **Configure an HTTP server instance**
   There are two options to chose from:
   - Create a new HTTP server instance to be used by the Websphere Version 5 instance. This method allows both Websphere Version 3.5.x or 4.0.x and Version 5 instances to continue operating correctly.
   - Update the HTTP server instance configuration for the Websphere Version 3.5.x or 4.0.x instance that is being migrated. This method changes the HTTP instance configuration to work with the Websphere Version 5 instance and makes the Websphere Version 3.5.x or 4.0.x instance no longer usable.
2. **Configure the virtual host** for the Version 5 instance.
   This step ensures that both the host and HTTP transport port number exist in the virtual host list.
3. **Regenerate the plug-in configuration file, plugin-cfg.xml.**
   This step 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 5 instance, 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 5 instance, see J2EE security (page 8) on the Migration configuration mapping page. For more information on editing the plug-in, see Step 5: Configure HTTPS transport for the Web container of the Configure SSL for WebSphere Application Server in the *Security* topic.

Change the ConnectionIOTimeOut properties for the Web container

Peformance changes to the WebSphere HTTP plug-ins may result in **InterruptedIOTimeout** exceptions while reading large requests, such as receiving file uploads to a servlet. Increase the **ConnectionIOTimeout** value in the Web Container of your application as follows:
1. In the topology tree, expand **Servers** and click **Application Servers**.
2. On the **Application Servers** page, click the name of the server that you want to modify.
3. On the server's detail page, click **Web Container**.
4. On the **Web Container** page, click **HTTP Transports**.
5. On the **HTTP Transports** page, click the transport that you want to modify.
6. Click **Custom properties**.
7. Click **New** and add a property named **ConnectionIOTimeout**. Set the property to the maximum time a servlet or JSP waits for a client to transmit request data.

8. After you add a virtual host alias, you must restart the application server. See Start and stop application servers in the *Administration* topic for more information.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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 Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY  10594-1785
U.S.A.
```

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: WebSphere
AS/400
e (logo)
IBM
iSeries
Operating System/400
OS/400
400

Lotus, Freelance, and WordPro are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

IBM ®