IBM WebSphere Application Server - Express Version
5.1

Migration Guide

Note!

Before using this information and the product it supports, be sure to read the general information under **"Notices" on page 61**.

Sixth Edition (December 2003)

Contents

Chapter 1. WebSphere Application Server - Express Migration Guide overview

In this version of IBM® WebSphere®Application Server - Express Version 5.1 , you can migrate code from:

- VisualAge for Java
- IBM WebSphere Application Single Server Edition Version 3.5
- IBM WebSphere Application Server - Express Version 5.0, 5.0.1, 5.0.2
- WebSphere Studio ″Classic″
- IBM WebSphere Studio Site Developer
- IBM WebSphere Studio Site Developer Version 5 Beta, Early Availability, or General Availability
- IBM WebSphere Studio Site Developer Version 5.0.1
- IBM WebSphere Studio Site Developer Version 5.1

WebSphere Application Server - Express 5.1 is comprised of WebSphere Application Server 5.1 and WebSphere Studio Site Developer 5.1.1. The first chapter below covers migration of the server feature of WebSphere Application Server - Express. The remainder of this Migration Guide is devoted to migrating code from different versions of WebSphere Studio Site Developer.

Important note regarding migrating the server:

Migrating your server configuration is only meaningful if you are administering the server using the Administrative Console -- typically in a production environment. In this mode of operation the server configuration and deployed applications are stored in the config directory of the server. The migration process migrates these configuration and application files for you. If, on the other hand, you are using WebSphere Studio Site Developer to configure and deploy applications to your remote server, there is no need to migrate the server configuration files. This is because the configuration and application files are all maintained in the Studio Site Developer workspace. The workspace will be migrated by Studio Site Developer. You can then define a new instance of a WebSphere Application Server - Express 5.1 server and continue configuring and deploying your applications from Studio Site Developer.

This guide is organized into the following chapters:

- Migrating your production server
- Migrating from IBM WebSphere Studio Site Developer Version 5.1
- Migrating from IBM WebSphere Studio Site DeveloperVersion 5 or Version 5.0.1
- Migrating IBM WebSphere Studio Site Developer from Version 4.0.x
- Windows Migrating from WebSphere Studio ″Classic″ to IBM WebSphere Studio Site Developer
- Migrating from VisualAge for Java to IBM WebSphere Studio Site Developer
- Migrating from VisualAge for Java Visual Composition Editor to Visual Editor for Java
- Build setup (library JARs, dependant project JARs, Ant builds)
- Migration examples

- Further reading

Information about using WebSphere Application Server - Express can be found in the *Getting Started* guide and the online help. Read the *Installation guide* prior to installing WebSphere Application Server - Express. After you have successfully installed WebSphere Application Server - Express, read the *Getting Started* guide and complete the *Getting Started* tutorials. The tutorials will introduce you to the workbench, Java™ development, and Web services. After you have completed the tutorials, read this guide to migrate your application resources into WebSphere Application Server - Express.

This guide is available in both HTML and Acrobat PDF versions, in the /readme directory. Both versions contain the identical information. You can open migrate.html in any Web browser. To open migrate.pdf, you must have installed the Acrobat Reader software, which you can download from www.adobe.com/products/acrobat/readstep2.html.

This *Migration Guide* uses Windows® conventions throughout. For example, ″WS_Installdir\″ in Windows is equivalent to ″WS_Installdir/″ in Linux.

For future updates to this guide, refer to www.ibm.com/websphere/developer/zones/studio/transition.html.

Chapter 2. Migrating your production server

Migration

Migrating is an activity in which you take advantage of existing materials. Migration tasks and tools help you upgrade the product and its prerequisites, reuse existing application components when feasible, and transfer administrative configurations from your past version to a current one.

Migration of WebSphere Application Server products is about leveraging the existing environment and applications and changing them to be compatible with the current product version.

Product migration functions are provided by the migration tools in IBM WebSphere Application Server - Express, Version 5.1. The migration tools support migrating from:

- IBM WebSphere Application Server - Express Version 5.0, 5.0.1, 5.0.2
- IBM WebSphere Application Single Server Edition Version 3.5

The product installation wizard will detect previous versions of IBM WebSphere Application Server - Express and will give you an option to run the migration tools during the installation of Version 5.1. To migrate from IBM WebSphere Application Server Version 3.5 you must run these tools directly.

Migration Redbook

Migrating to WebSphere V5: An End-to-End Migration Guide is available from the Redbooks Web site at http://www.ibm.com/redbooks. To locate the Redbook, search for the document number SG24-6910-00. The Redbook provides a broader coverage than this article, including more detailed planning information for application migration and WebSphere Studio Application Developer tooling and samples.

Version 3.5 migration: Moving to the J2EE model

V3.5.x users upgrading to V5 are moving to a platform that is based on the J2EE specifications. J2EE technology clearly separates development and the creation of applications from application administration, deployment and management. Migration from V3.5 involves changes in application structures, development, and deployment.

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 tune migrated applications, using development and deployment tools, to determine exactly how the applications function in the new environment.

The J2EE model enables you to develop applications independently from their final deployment environment. This task separation simplifies the process of promoting an application from initial development through production, or moving an

application from one server to another. The intent is to change only the application deployment parameters, and not the application code.

Migrating and coexisting

Before you begin, determine whether you have an existing version of WebSphere Application Server installed on the machine where you plan to install your Version 5.1 product. If you have a previous version, you must plan whether to copy the configuration and applications of the previous version to the new version, which is *migration*. Migration does not uninstall the previous version. The earlier release is still functional. If you run it at the same time as the Version 5.1 installation, the two versions are *coexisting*. In order to run both versions at the same time you will have to configure the ports such that they do not conflict. Note that the migration operation just migrates the port definitions as is, such that your port definitions are the same on both versions. WebSphere Application Server contains migration tools that provide all migration functionality. The installation wizard can call the migration tools, or you can call them manually at a later time.

In overview, the migration from IBM WebSphere Application Server - Express V5.0.x to V5.1 is fairly routine. You can use the installer to migrate and have little or no post-migration tuning to perform. Or you can use the migration tools manually to save the V5.0.0, V5.0.1, or V5.0.2 configuration data, uninstall V5.0.0, V5.0.1, or V5.0.2, install V5.1, and use the migration tools again to restore the configuration data.

In overview, the migration from IBM WebSphere Application Server V3.5 to IBM WebSphere Application Server - Express V5.1 involves significant changes in application structures, development, and deployment. The migration tools assist in this transition by migrating system configurations and creating J2EE artifacts, including mapping previous security settings to J2EE security roles. These security mappings let you access migrated assets during the transition. The migration tools create initial J2EE enterprise applications based on V3.5.x configurations. However, because of the significant changes in the application structures, carefully test and fine tune migrated applications using development and deployment tools.

Migration saves the following files in the `backup` directory.

For Version 3.5.x

- `bin/setupCmdLine.sh` (or `bin/setupCmdLine.bat` for Windows platforms)
- `classes` (not saved for iSeries)
- `hosts`
- `properties`
- `servlets`

For Version 5.0.x

- `classes` (not saved for iSeries)
- `config`
- `installableApps`
- `installedApps`
- `properties`

Migration tools

This topic introduces the migration tools that WebSphere Application Server provides. All migration tools ship in the /migration directory on the product CD-ROM. It is important to use the migration tools for the version of Application Server that you are installing. The tools change over time. The tools on the product CD-ROM provide the necessary function for migrating from a previous release of Application Server to the one on the product CD-ROM. The tools on the CD-ROM match the product on the CD-ROM. If you use migration tools from an earlier release of Application Server, you are likely to encounter a problem with the migration.

WASPreUpgrade.sh (and WASPreUpgrade.bat)
Saves the applications and configuration data from a previous installation of WebSphere Application Server to a backup directory. The WASPostUpgrade script restores the configuration data from the directory to the new installation. The installer calls the WASPreUpgrade.sh script during installation, if you select migration. You can also use the command to perform a manual migration, after installing the new version.

WASPostUpgrade.sh (and WASPostUpgrade.bat)
Restores the configuration data from a previous release. WASPostUpgrade reads the data from the backup directory where the WASPreUpgrade script stored the data. The installer calls the WASPostUpgrade.sh script during installation, if you select migration. You can also use the command to perform a manual migration, after installing the new version.

WASPreUpgrade command

The WASPreUpgrade.sh (or WASPreUpgrade.bat) script is a migration tool for migrating the configuration and applications of previous versions or releases to a Version 5.1 Application Server - Express.

The command file is located in the AppServer/bin subdirectory of the installation root after installation. It is also available directly from the CD in the migration subdirectory.

Syntax

```
WASPreUpgrade backupDirectory currentWASDirectory
              [adminNodeName]
              [-nameServiceHost host_name [-nameServicePort port_number ]]
              [-traceString trace_spec [-traceFile file_name ]]
```

Parameters

The first two arguments are required and positional.

Supported arguments include:

backupDirectory
Required positional name of the directory in which the WASPreUpgrade tool stores the saved configuration and files, and from which the WASPostUpgrade tool later reads the configuration and files. The WASPreUpgrade tool creates this directory if it does not already exist.

currentWASDirectory
Required positional name of the installation root for the current V3.5.x or

V5.0.x installation. This version can be either WebSphere Application Server Standard Edition, V3.5.x, WebSphere Application Server - Express V5.0.x.

adminNodeName
Optional, positional name of the node containing the administrative server for the currently installed product. The value of this argument must match the node name given in the topology tree on the `Topology` tab of the administrative console for the currently installed product. The WASPreUpgrade tool calls the XMLConfig tool using this parameter. This parameter is only required when upgrading from WebSphere Application Server Standard Edition, Version 3.5.x.

-nameServiceHost -nameServicePort
When specified, the WASPreUpgrade tool passes these optional parameters to the XMLConfig tool. Use these parameters to override the default host name and port number used by the XMLConfig tool.

-traceString -traceFile
Optional parameters to gather trace information for IBM Service personnel. Specify a trace_spec of `"*=all=enabled"` (with quotation marks) to gather all trace information.

Logging

The WASPreUpgrade tool displays status to the screen while it is running. It also saves a more extensive set of logging information in the *backup* directory. You can view the `WASPreUpgrade.log` file with a text editor.

WASPostUpgrade command

The `WASPostUpgrade.sh` (or `WASPostUpgrade.bat`) script is a migration tool for migrating the configuration and applications of previous versions or releases to a Version 5.1 Application Server - Express.

The command file is located in the `AppServer/bin` directory of the installation root.

The WASPostUpgrade tool installs all migrated applications into the *AppServer*`/installedApps` directory for the Version 5.1 installation. The tool includes applications from the backup directory that the WASPreUpgrade tool creates. The WASPreUpgrade tool copies the applications from the `installedApps` directory and other directories in the earlier version or release.

Syntax

```
WASPostUpgrade backupDirectory
               [-serverName server_name]
               [-webModuleAdditionalClasspath classpath]
               [-documentRootLimit number]
               [-substitute "key1=value1[;key2=value2;[...]]"]
               [-portBlock port_starting_number]
               [-backupConfig true | false]
               [-replacePorts true | false]
               [[-traceString trace_spec [-traceFile file_name]]
```

Parameters

The first argument is required. Supported arguments include:

serverName
Optional server instance name. Defaults to server1.

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.

-backupConfig
Optional parameter used to back up the existing configuration before the migration tools change the configuration. The default is true, to back up the configuration.

-documentRootLimit
Optional parameter to specify the number of files that the program copies from the document-root field of Web-application. It is only applicable to Version 3.5.x upgrades. If not specified, the default is 300.

-portBlock
Optional parameter used to specify the starting value to use when creating ports.

-substitute
Optional argument passed to the XMLConfig tool. Specify values for security variables to substitute (for example, `-substitute "NODE_NAME=admin_node;APP_SERVER=default_server"`).

In the input XML data file, each key appears as *key* for substitution. This argument substitutes any occurrence of `$NODE_NAME$` with *admin_node* and `$APP_SERVER$` with *default_server* in the input XML file.

If the substitution string contains semicolons, use `$semiColon$` to separate it from the ";" delimiter. On UNIX platforms, add an escape character to each dollar sign ($) within the substitution string (for example, `\$semiColon\$`).

This parameter is applicable for configurations saved from Advanced Edition, Version 3.5.x.

-traceString -traceFile
Optional parameters to gather trace information for IBM Service personnel. Specify a trace_spec of `"*=all=enabled"` (with quotation marks) to gather all trace information.

-webModuleAdditionalClasspath
Optional parameter to specify the path or the path and file names of specific directories or files that you do not want copied into the Web archive (WAR) file. Instead, the program adds the paths and files to the Web Module extension (`ibm-web-ext.xmi`) `additionalClassPath` attribute. This is only applicable when migrating a Version 3.5.x installation.

Logging

The WASPostUpgrade tool displays status to the screen while running. It also saves a more extensive set of logging information in the `logs` directory. You can view the `WASPostUpgrade.log` file with a text editor.

Configuration mapping during migration

This topic describes what changes during migration, which always involves a single machine, such as a development environment on a stand-alone machine.

Version 3.5 to Version 5.x 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 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 combines resources in a J2EE EAR file and deploys it in the Version 5 configuration.

Mapping details for V3.5.x to Version 5.x migration

- **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.

- **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 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.

- **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. The J2EE specification includes 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.

  The J2EE specification includes 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 the J2EE specification. The deployment descriptor of the Web application WAR file contain the security constraints and Web resource collections.

- **JSP levels**

  Version 5 runs JSP 1.0 and 1.1 objects as JSP 1.2 objects, which is the only supported level.

- **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 V3.5.x to V5.x**

  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.

- **Transports**

  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 HTTP transports, using the same port assignments. You must manually add VirtualHost alias entries for each port.

Migrating configuration data manually

You can migrate administrative configurations with the installation wizard or manually, as this task describes. If you decide to migrate manually, do not select the migration check box on the installation wizard migration panel.

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

You can perform incremental manual migration by calling the migration tools multiple times, each time specifying a different configuration file. There are various reasons for having multiple configuration files. Whatever the reason, migrating one configuration file at a time lets you test applications incrementally before continuing to the next configuration file.

Before using the migration tools, consult the V5.x Release Notes document to understand what fixes you must apply to earlier versions. Applying fixes to an earlier version might also apply fixes to files that have a role in the migration. Apply any fixes to ensure the most effective migration of configurations and applications possible.

Manual migration provides a more incremental migration approach than the complete migration that the installation wizard provides. IBM provides a set of migration tools for migrating administrative configurations to the WebSphere Application Server - Express product from any edition of V3.5.x or V5.0.x. The overall migration process is to back up the current configuration and necessary files with the WASPreUpgrade migration tool, uninstall the prior release, install the Version 5 product without selecting the automated migration option, and restore the configuration from the earlier release with the WASPostUpgrade migration tool.

Select any of these migration scenarios for information about how to migrate configuration data to a base WebSphere Application Server node:

- “Migrating V3.5.x to V5.1”
- “Migrating V3.5.x to a remote V5.1 machine” on page 12
- “Migrating V5.0.x to V5.1” on page 14
- “Migrating V5.0.x to a remote V5.1 machine” on page 15
- “Migrating from an unsupported operating system” on page 16

Migrating V3.5.x to V5.1

You can use the migration tools to migrate configuration data from Version 3.5 of WebSphere Application Server to Version 5.1 of WebSphere Application Server - Express.

Typically you would use the WASPreUpgrade and WASPostUpgrade migration tools from V5.1 of WebSphere Application Server to upgrade from V3.5 to V5.1 on the *same* machine. If your scenario includes migrating a V3.5 configuration on one

machine to WebSphere Application Server - Express V5.1 on another machine, use the alternate procedure described in “Migrating V3.5.x to a remote V5.1 machine” on page 12.

This topic describes using the V5.1 migration tools to migrate:

- WebSphere Application Server Single Server Edition, V3.5

The WASPreUpgrade tool saves the existing V3.5 configuration into a *migration-specific-backup* directory. The WASPostUpgrade tool uses this directory to add the old configuration settings to the new V5.1 environment.

Steps for this task

1. Obtain the V5.1 product CD-ROM.

   On this CD is the `migration/bin` directory. This directory contains a special environment that you can use to run the WASPreUpgrade tool without installing V5.1.
2. Save the current configuration using the WASPreUpgrade script from the `/migration/bin` directory of the V5.1 product CD-ROM.

   Save the configuration in the `migration-specific-backup` directory:

   ```
   WASPreUpgrade /usr/tmp/migration-specific-backup /usr/websphere/appserver yourNodeName
   ```

   Verify that the administrative server of the existing environment is running. The WASPreUpgrade tool provides status to the screen and to log files in the `migration-specific-backup` directory. ASCII log file names start with the text WASPreUpgrade and include a date timestamp.

   The WASPreUpgrade tool saves all files from the following directories in the existing V3.5.x configuration to the backup directory:

   For Version 3.5.x

   - `bin`
   - `classes`
   - `hosts`
   - `properties`
   - `servlets`

   The WASPreUpgrade tool saves selected files from the V3.5.x `/bin` directory. It also exports the existing Application Server configuration from the V3.5.x repository. The WASPreUpgrade tool calls XMLConfig to export the existing V3.5 repository to the `websphere_backup.xml` file in the `migration-specific-backup` directory.

   If errors occur while running the WASPreUpgrade tool, you might have to apply fixes to the V3.5 installation to successfully complete the export step. See the IBM Support page for the latest fixes that might be applicable. When viewing this information from the InfoCenter, click **Support** to link to the IBM Support page.
3. Install V5.1 of the WebSphere Application Server - Express Version product.

   Do not select the migration option, if it appears.

   After each use of WASPostUpgrade, verify V5 port settings in two files:

   - Verify the `BOOTSTRAP_ADDRESS` port assignment for `server1` in the `serverindex.xml` file

If the BOOTSTRAP_ADDRESS port of the earlier version is 900, migration maps this to 7809. If the BOOTSTRAP_ADDRESS port of the earlier version is not 900, migration maps the value to server1 in a Advanced Edition migration, or to the actual server name in an Advanced Single Server Edition migration.

- Verify the `HTTP Transport` port assignments in the `server.xml` file

  WASPostUpgrade processing adds the HTTP Transport ports from the earlier version to the Version 5 `server.xml` file. This means that server1 contains duplicate HTTP Transport port assignments, from both the coexistence panel and the previous version *Default Server*.

4. Migrate the previous configuration to the new installation with the WASPostUpgrade tool in the `AppServer/bin` directory of the V5.1 installation root directory.

   The WASPostUpgrade tool migrates V3.5.x configuration information created by the WASPreUpgrade tool to the V5.1 installation. Because the V5.1 product adheres to the J2EE programming model and V3.5.x does not, significant changes are required to apply the V3.5.x configuration to a V5.1 installation.

   The WASPostUpgrade tool does not migrate Samples or the administrative console application because there are already Samples and an administrative console application in V5.1.

   The WASPostUpgrade tool records detailed information specific to each enterprise bean it deploys, in the `WASPostUpgrade.log` file.
5. Stop the administrative server of the earlier version if it is running, before running the Version 5 node.
6. Configuring WebSphere Application Server after migration is a way of verifying the results of the migration tools. You can also use Configuration mapping during migration to verify the results of the migration. The topic has a detailed description of how the migration tools migrate objects, and what you should verify.

Migrating V3.5.x to a remote V5.1 machine

You can use the migration tools to perform a manual migration between two machines.

Typically you would use the WASPreUpgrade and WASPostUpgrade migration tools from V5.1 of WebSphere Application Server - Express to upgrade from V3.5 to V5.1 on the *same* machine.

However, there are some scenarios where you must migrate the V3.5 configuration on one machine to V5.1 on a different machine. One of these scenarios is installing new machines for your latest V5.1 environment but needing to migrate your existing V3.5 configuration on other machines.

This topic describes using the V5.1 migration tools to migrate:

- WebSphere Application Server Single Server Edition, V3.5

The WASPreUpgrade tool saves the existing V3.5 configuration into a *migration-specific-backup* directory. The WASPostUpgrade tool uses this directory to add the old configuration settings to the new V5.1 environment.

Steps for this task

1. Obtain the V5.1 product CD-ROM.

On this CD is the migration/bin directory. This directory contains a special environment that you can use to run the WASPreUpgrade tool without installing V5.1.

2. Save the current configuration using the WASPreUpgrade script from the /migration/bin directory of the V5.1 product CD-ROM, which you must mount to the V3.5 machine.

   Save the configuration in the migration-specific-backup directory on the V3.5 machine.

   ```
   WASPreUpgrade /opt/tmp/migration-specific-backup /opt/websphere/appserver adminNodeName
   ```

   Vverify that the administrative server of the existing environment is running.

   The WASPreUpgrade tool provides status to the screen and to log files in the /migration-specific-backup directory. ASCII log file names start with the text WASPreUpgrade and include a date timestamp.

   The WASPreUpgrade tool saves selected files from the V3.5.x/bin directory. It also exports the existing Application Server configuration from the V3.5.x repository. The WASPreUpgrade tool calls XMLConfig to export the existing V3.5 repository to the websphere_backup.xml file in the migration-specific-backup directory.

   If errors occur while running the WASPreUpgrade tool, you might have to apply fixes to the V3.5 installation to successfully complete the export step. See the IBM Support page for the latest fixes that might be applicable. When viewing this information from the InfoCenter, click **Support** to link to the IBM Support page.

3. Copy the /migration-specific-backup directory from the V3.5 machine to the V5.1 machine.

   Use ftp, shared storage, or some other mechanism to copy the file to the new machine.

   Perform the following steps on the machine with V5.1 of WebSphere Application Server - Express..

4. Copy the /migration-specific-backup/websphere_backup.xml or /migration-specific-backup/config/server-cfg.xml file and store it in a location of your choice to preserve the copy as an archive.

   You are copying the file because you edit the original file in the next step.

5. Edit the /migration-specific-backup/websphere_backup.xml or /migration-specific-backup/config/server-cfg.xml file to correct machine-dependent settings.

   Make these changes in the file:

   a. **Change the node name** in the /migration-specific-backup/websphere_backup.xml file. There is no node name in the /migration-specific-backup/config/server-cfg.xml file.

      If you are using the same node name for the V5.1 machine that you use for the original V3.5 configuration, do not change the name. Otherwise, you **must** change all occurrences of the V3.5 node name to the node name you are using for WebSphere Application Server V5.1. The node name occurs in many XML stanzas throughout the file. Failing to change all occurrences results in an incomplete migration of data.

   b. **Change the path names** in the /migration-specific-backup/websphere_backup.xml or the /migration-specific-backup/config/server-cfg.xml file.

The configuration file refers to path names in many XML stanzas throughout the file. Update any reference to a file outside of the V3.5 directory structure to the equivalent directory on the new machine, even if you must create an equivalent directory. The implication of configuring a matching environment means that you might have to copy the original directory to the V5.1 machine. Or you might have to install the appropriate software.

c. **Correct specification styles** for path names that are dependent on the operating system.

   You must correct path specifications if they differ from what works on the machine running V5.1. For example, if you are moving from V3.5 on a Windows platform to V5.1 on a Linux platform, change any Windows-specific path in the configuration file to use the Linux path style. Change `"c:\mystuff\somepath"` to `"/opt/mystuff/somepath"`.

d. **Change user IDs and passwords to match security requirements.**

   You might have to change user IDs and passwords if they are not identical to those in use on the V5.1 machine.

   To change an encoded password to a clear-text password, change `<password>{xor}LCoxayht</password>` to `<password>mypassword</password>`.

e. **Change other machine-specific information.**

   The configuration might refer to other software products or configurations that do not exist on the new machine. For example, the old machine might have a database. The V5.1 configuration should still point to the database on the old machine, possibly. Modify the datasource to point to database on the V3.5 machine.

6. Install V5.1 of the WebSphere Application Server without selecting the migration option.
7. Add the V3.5 configuration to the V5.1 configuration.

   Use the WASPostUpgrade tool in the `AppServer/bin` directory of the V5.1 installation root directory to add the V3.5 configuration to the V5.1 configuration.

   ```
   WASPostUpgrade /opt/tmp/migration-specific-backup
   ```

   The WASPostUpgrade tool records detailed information specific to each enterprise bean it deploys, in the `/migration-specific-backup/WASPostUpgrade.log` file.

Migrating V5.0.x to V5.1

You can use the V5.1 installation program to migrate from WebSphere Application Server - Express V5.0.x to V5.1.

Steps for this task:

1. Stop the V5.0.x Application Server.

   Use the stopServer.sh (or stopServer.bat) script from the `AppServer/bin` directory of the installation root:

   ```
   stopServer.sh server1
   ```

   You can migrate a V5.0.x node without stopping it. However,it is not necessary to have the node running to migrate its configuration. The migration tools can retrieve all the configuration data while the node is stopped. And you must stop the node before you can start the V5.1 node that you are installing. So, you can stop the node now.

2. Install the V5.1 product.

   Select the migration option, when it appears.
3. Verify the installation of the V5.1 Application Server.

   Use the First Steps tool when it appears at the end of installing the product, or run the installation verification test yourself, if the First Steps tool does not appear for some reason.

You can uninstall the V5.0.x server at your convenience.

Migrating V5.0.x to a remote V5.1 machine

You can use the migration tools to migrate between two machines.

Before you begin

Typically you would use the WASPreUpgrade and WASPostUpgrade migration tools from V5.1 of WebSphere Application Server to upgrade from either V5.0.x to V5.1 on the *same* machine.

However, there are some scenarios where you must migrate the V5.0.x configuration on one machine to V5.1 on a different machine. One of these scenarios is installing new machines for your latest V5.1 environment but needing to migrate your existing V5.0.x configuration on other machines.

This task describes how to use the V5.1 migration tools to perform the migration.

The WASPreUpgrade tool saves the existing V5.0.x configuration into a *migration-specific-backup* directory. The WASPostUpgrade tool uses this directory to add the old configuration settings to the new V5.1 environment.

Steps for this task

1. Obtain the V5.1 product CD-ROM.

   On this CD is the `migration/bin` directory. This directory contains a special environment that you can use to run the WASPreUpgrade tool without installing V5.1.
2. Save the current configuration using the WASPreUpgrade script from the `/migration/bin` directory of the V5.1 product CD-ROM, which you must mount to the V5.0.x machine.

   Save the configuration in the `migration-specific-backup` directory on the V5.0.x machine.

   ```
   WASPreUpgrade /opt/tmp/migration-specific-backup /opt/websphere/appserver
   ```

   The WASPreUpgrade tool provides status to the screen and to log files in the `/migration-specific-backup` directory. ASCII log file names start with the text "WASPreUpgrade" and include a date timestamp.
3. Copy the `/migration-specific-backup` directory from the V5.0.x machine to the V5.1 machine.

   Use ftp, shared storage, or some other mechanism to copy the file to the new machine.
4. Install V5.1 of WebSphere Application Server without selecting the migration option.
5. Add the V5.0.x configuration to the V5.1 configuration.

Use the WASPostUpgrade tool in the AppServer/bin directory of the V5.1 installation root directory to add the V5.0.x configuration to the V5.1 configuration.

```
WASPostUpgrade /opt/tmp/migration-specific-backup
```

The WASPostUpgrade tool records detailed information specific to each enterprise bean it deploys, in the /migration-specific-backup/WASPostUpgrade.log file.

6. Modify the configuration using the WebSphere Application Server 5.1 administration interfaces.

   Make these changes:

   a. **Change user IDs and passwords to match security requirements.**

      You might have to change user IDs and passwords if they are not identical to those in use on the V5.0.x machine.

   b. **Change other machine-specific information.**

      The configuration might refer to other software products or configurations that do not exist on the new machine. For example, the old machine might have a database. Modify the datasource to point to the database on the old machine.

7. You can uninstall the V5.0.x server at your convenience.

Migrating from an unsupported operating system

You can migrate an earlier version of WebSphere Application Server Version 3.5.x or Version 5.0.x release that is running on an operating system that Version 5.1 does not support.

Steps for this task

1. Start up the WebSphere Application Server Version 3.5.x or Version 5.0.x Administrative Server.
2. Run the **WASPreUpgrade** command line migration tool.

   There are two options. You can run the command from the migration\bin (or migration/bin) directory in the *platform_root* of the Version 5.1 CD-ROM. Or, you can copy the files in the directory on the CD-ROM to a directory you create on your hard drive.

   Identify the Version 3.5.x or 5.0.x release, and identify a backup directory where the command stores configuration files and migrating applications from the earlier version. See the *WASPreUpgrade* topic for command syntax.

   a. Run the command from the migration\bin (or migration/bin) directory in the *platform_root* of the Version 5.1 CD-ROM.

      Identify the backup directory and the location of the configuration files.

      ```
      CD_drive:\WASPreUpgrade backupDirectory filepath\WebSphere\AppServer yourNodeName
      ```

      If this works, go to Step 4. If this does not work for some reason, perform steps 2B through 2F.

   b. Make a **migration** directory on your hard drive.

   c. Copy the WASPreUpgrade.bat (or WASPreUpgrade.sh) and the setupCmdLine.bat (or setupCmdLine.sh) files from the migration\bin\ (or migration/bin/) directory in the *platform_root* of the Version 5.1 CD-ROM, to the directory you created on your hard drive.

   d. Edit the setupCmdLine.bat (or setupCmdLine.sh) file in your new directory.

Change the following variables:

- **WAS_HOME** to point to the fully qualified path to the migration directory you created
- **JAVA_HOME** to point to the fully qualified path to your IBM Developer Kit or the Java directory

e. Ensure that the executable bit is on for the `setupCmdLine.sh` and `WASPreUpgrade.sh` files in the `migration/bin` directory in the *`UNIX-based_platform_root`* of the Version 5.1 CD-ROM, if you are backing up a UNIX-based installation.

f. Run the command from the `migration` directory you created.

Identify the backup directory and the location of the configuration files.

`yourMigrationDirectory``\WASPreUpgrade` *`backupDirectory filepath`*`\WebSphere\AppServer` *`yourNodeN`*

3. Shut down the WebSphere Application Server Version 3.5.x or Version 5.0.x.
4. Tar or zip the backup directory and FTP it to another system.
5. Install the new operating system, keeping the same host name.

   If possible, keep the system name and passwords the same as the old system. Place any database files related to applications you are migrating in the same path as the previous system. In general, try to keep paths the same. However, do not install Version 5.1 in the same directory as the previous version. If you do change paths and names, you can edit the XML configuration files to reflect the changes. Make such changes before running the **WASPostUpgrade** command below.
6. FTP the backup directory from the other system and unzip it.
7. Install WebSphere Application Server- Express, Version 5.1. Do not select the migration option, if it appears.
8. Run the **WASPostUpgrade** command line migration tool, from the `bin` directory of the Version 5.1 install_root.

   Specify the backup directory (and any non-standard configuration file name in the directory) that the **WASPreUpgrade** command created. See the *WASPostUpgrade* topic for proper command syntax.

   `install_root``\bin\WASPostUpgrade  WAS_HOME\migration`

Chapter 3. Migrating from IBM WebSphere Studio Site Developer Version 5.1

This chapter covers the following migration topics:

- Migrating J2EE projects to use Server Targeting support
- Wizard generation requires a Java package for JDK 1.4

Migrating J2EE projects to use Server Targeting support

In IBM WebSphere Studio Site Developer Version 5.1.1 there is a new Server Targeting function added. This function by default is disabled and you need to enable the function in the J2EE preferences page by selecting **Window > Preferences > J2EE**. Functional details about the Server Targeting function can be found in the IBM WebSphere Studio Site Developer product documentation. When the function is enabled you have the option to target a particular application server. This feature has been implemented to support JDK 1.4, which is the JRE for WebSphere Application Server Version 5.1 that is shipped with IBM WebSphere Studio Site Developer Version 5.1.1. J2EE projects that take advantage of the Server Targeting support are not backward compatible with earlier versions of IBM WebSphere Studio Site Developer, hence *cannot* be shared with users working on earlier versions of IBM WebSphere Studio Site Developer (for example, IBM WebSphere Studio Site Developer Version 5.1, IBM WebSphere Studio Site Developer Version 5.0.1). IBM WebSphere Studio Site Developerprovides a way to get backward compatibility with this feature enabled and is described in "Backward compatibility with Server Targeting support enabled" on page 19. The reason for this incompatibility is that the Server Targeting function changes the .classpath file on a J2EE project and the new .classpath file entries *cannot* be recognized by earlier versions of WebSphere Application Server - Express.

Backward compatibility with Server Targeting support enabled

With the Server Targeting support enabled, J2EE projects targeted to a server can be reverted back to not use the Server Targeting support by modifying the target server to a **No target server specified** option available in the Modify Target Server dialog. The Modify Target Server dialog is launched from the pop-up menu (**Target Server > Modify**) on a J2EE project in the Resource Navigator or the J2EE Perspective view. The target server can also be modified to a No target server specified from the J2EE properties page (**Properties > J2EE**) for EAR, EJB, Application Client and Connector projects. For a Web project the target server setting is in the Web properties page (**Properties > WEB**). Functional details about the Server Target modify function can be found in the IBM WebSphere Studio Site Developer documentation. When the **No target server specified** option is used the .classpath file is reverted back to the style that is compatible with earlier versions of IBM WebSphere Studio Site Developer and the .server is removed from the project.

Note: Only Server Targeted J2EE projects can be deployed on WebSphere Application Server Version 5.1 and take advantage of JDK 1.4 support.

Wizard generation requires a Java package for JDK 1.4

Due to a change in the JDK 1.4, the user must specify a Java package when using the Database Web Pages and Java Bean Web Pages wizards to generate pages to run on IBM WebSphere Studio Site Developer Version 5.1.1. This problem occurs if the View Bean template is used for the Java Bean Web Pages wizard or the IBM Database Access Java Beans-Master Details Pattern. This also applies to projects that contain pages and .java files previously generated with these wizards that did not specify a package during creation. For code that was previously generated, move the .java files to a package. Then update the .jsp files, update the import statements and class information. In the web.xml file of the project, update the servlet-class entry.

Chapter 4. Migrating from IBM WebSphere Studio Site Developer Version 5 or Version 5.0.1

This chapter covers the following migration topics:

- WebSphere Studio Workbench (WSWB) in Version 5, Version 5.0.1, and Version 5.1
- Using WebSphere Studio Version 5.1.1 with Version 5.0 workspace
- Migrating Java projects from Version 5 or Version 5.0.1
- Sharing projects between Version 5 or Version 5.0.1 and Version 5.1 using a source code management (SCM) system
- Migrating Web projects
- Converting Web projects to Struts 1.1
- Migration function added to J2EE project migration
- Changes to Web services tools
- Changes made in Profiling tools
- Known compatibility problems of the Template wizard

WebSphere Studio Workbench in Version 5, Version 5.0.1, and Version 5.1

IBM WebSphere Studio Site Developer Version 5.1.1 is based on the new Eclipse base WebSphere Studio Workbench (WSWB) 2.1.2. There are some differences between versions 2.1.2 and 2.0.3 or 2.0.2. To get a detailed information about the differences, refer the readme file located in the *WS_Installdir*\eclipse\readme directory (where *WS_Installdir* is the path where you have installed IBM WebSphere Studio Site Developer .

IBM WebSphere Studio Site Developer Version 5.0 was based on the Eclipse base WSWB 2.0.2 and IBM WebSphere Studio Site Developer Version 5.0.1 was based on the Eclipse base WSWB 2.0.3. There are no major differences between versions 2.0.2 and 2.0.3. The IBM WebSphere Studio Site Developer Version 5.0.1 release was an Update Manager fix pack that installed on top of IBM WebSphere Studio Site Developer Version 5.0.

Using IBM WebSphere Studio Site Developer Version 5.1.1 with Version 5.0 workspace

When IBM WebSphere Studio Site Developer Version 5.1.1 is started for the first time using an existing IBM WebSphere Studio Site Developer Version 5.0 workspace a dialog box appears showing you one way to migrate from Version 5.0. Click **OK** to migrate the Version 5.0 workspace, or click **Cancel** to stop IBM WebSphere Studio Site Developer from starting.

When the workspace is migrated, you can still use the workspace with Version 5.0 as the metadata of the new project features are ignored and can be read by Version 5.0. You cannot make any changes in Version 5.0 to the projects in the workspace that would effect the metadata or overwrite the metadata of the new project feature of projects.

Migrating Java projects from Version 5 or Version 5.0.1

Migrating Java projects from Version 5 or Version 5.0.1 is very straightforward and automatic. Once projects are loaded into Version 5.1.1 workspace, no metadata changes occur in the .classpath files or the .project files unless you try to use any of the new features available in Version 5.1.1.

Sharing projects between Version 5 or Version 5.0.1 and Version 5.1.1 using a source code management (SCM) system

Special care is required when a project in a team repository is being loaded and operated on by developers using IBM WebSphere Studio Site Developer Version 5 and IBM WebSphere Studio Site Developer Version 5.1.1. The general problem is that the existence, contents, and interpretation of metadata files in the workspaces may be specific to a particular feature or plug-in version, and differ between versions. The workspace compatibility guarantees only cover cases where all developers upgrade their IBM WebSphere Studio Site Developer workspaces in lock step. In those cases there should be no problem with shared metadata. However, when some developers are working in IBM WebSphere Studio Site Developer Version 5.1.1 while others are working in IBM WebSphere Studio Site Developer Version 5, there are no such guarantees. This section provides advice for what to do and what not to do.

The typical failure mode is noticed by the IBM WebSphere Studio Site Developer Version 5.1.1 user. Version 5.1.1 metadata is lost when a Version 5 user saves changes and then commits the updated metadata files to the repository. Here are some things that can go wrong:

- A user working in IBM WebSphere Studio Site Developer Version 5.1.1 creates or modifies a project in a way that results in changes to a shared metadata file that rely on 5.1.1-specific information. The user then commits the updated project files, including the shared metadata file, to the shared repository.
- Another user working in IBM WebSphere Studio Site Developer Version 5 shares this project from the same repository. The 5.1.1-specific information in the shared metadata file is not understood by IBM WebSphere Studio Site Developer Version 5, and is generally discarded or ignored without warning. The user modifies the project in a way that results in changes to the shared metadata file, causing the shared metadata file to be rewritten without any of the 5.1.1-specific information. The user commits the updated project files, including the shared metadata file, to the shared repository. The user is generally unaware that shared information has just been lost as a result of their actions.
- A user working in IBM WebSphere Studio Site Developer Version 5.1.1 picks up the changes to a project from the shared repository, including the updated shared metadata file. The user may be unaware that they have just taken a retrograde step until later when things start to malfunction.

Here are a list of things to watch out for when the project is to be shared between users of IBM WebSphere Studio Site Developer Version 5.1.1 and Version 5 or Version 5.0.1:

- **Linked folders and files**

  This support was added in IBM WebSphere Studio Site Developer Version 5.1.1. Information about linked resources is recorded in the .project file.

  Recommendation: Do not use. Better still, disable the linked resources using the **Workbench > Linked Resources** preference page.
- **External tool (Ant) builders**

Information about external tool builder is recorded in the .project file. The format of the information changed between Version 5 and Version 5.1.1. Builders created or changed in Version 5.1.1 use the new format, which is not understood by a Version 5 workspace. Builders created in Version 5 use the old format and continue to work in Version 5.1.1.

Recommendation: Always create or edit external tools builders from a Version 5 workspace.

- **Optional exclusion patterns on Java source entries on the build class path**

  This support was added. This information is recorded in the .classpath file.

  Recommendation: Do not specify exclusion patterns. Better still, disable the exclusion patterns using the **Java > Compiler > Build Path** preference page.

- **Output folders associated with Java source entries on the build class path**

  This support was added. This information is recorded in the .classpath file.

  Recommendation: Do not specify anything other than the default (project-wide) output folder. Better still, disable the multiple output locations using the **Java > Compiler > Build Path** preference page.

- **Source attachment root path associated with Java library entries on the build class path**

  When attaching a source ZIP file to a JAR library on the Java build path, the source root path prefix is inferred automatically. This has changed from Version 5, where it could be explicitly set using the user interface and explicitly recorded in the .classpath file. Consequently, a Java project created in a 5.1.1 workspace might not find the attached source.

  Use Version 5 to specify the source attachment root path. There is additional source attachment flexibility provided in Version 5.1.1. You can provide a folder instead of a JAR or ZIP file as a source attachment, and you can attach source to a class file folder; this functionality is not available in Version 5 (where the Version 5.1.1 information is ignored).

- **PDE class path containers for dependent plug-ins**

  PDE using class path containers was added. Class path containers are recorded in the .classpath file. If PDE class path containers are used, then a Version 5 workspace will have unresolved class path entries and, therefore, most Java capabilities (including compilation, search, run, debug) will not produce the expected results.

  Recommendation: Ensure that the setting on the **Plug-in Development > Java Build Path Control** preference page for using class path containers is disabled before creating any new plug-in (or fragment) projects.

Migrating Web projects

The folder names are **Java Source** and **Web Content**. The default folder names for new Web projects are configurable through a preference page (**Window > Preferences > Web Tools > New Project**). The default names are now **JavaSource** and **WebContent**. These default names will be used for new Web projects only. Web projects created in versions prior to this release will continue to function using the old names. The same is true for Static Web projects.

If you opt to change the source folders names for 4.0.x and 5.0 projects in Version 5.1.1, use the **Rename** pop-up menu action in the Navigator view. The **Rename** pop-up menu action renames the folder names and fixes the Java build path for the 4.0.x and 5.0.x Web projects.

For the **JavaSource** folder, the **Rename** pop-up menu action works in the Project Navigator view and the Packages view. For the **WebContent** folder, the **Rename** pop-up menu action works in the Resource Navigator view and the Project Navigator view.

If a Web project of a version prior to this release is saved into an SCM repository and then loaded into this release, it will retain the old structure with the **source** and **webApplication** folders. Either structure will build correctly.

Note: If the users opt to rename **Java Source** and **Web Content** folder names, then they have to manually update any automated build scripts they have to change them to use the new folder names.

Converting Web projects to Struts 1.1

The Struts tools run-time has stepped up from Version 1.1 Beta 2 in Version 5 to Version 1.1. In IBM WebSphere Studio Site Developer Version 5 (General Availability) when you create a Web project you have the option to add Struts support to your project. You can choose either Struts 1.0.2 or Struts 1.1 Beta 2. In IBM WebSphere Studio Site Developer Version 5.1.1, the latter choice is replaced by Struts 1.1. If you created Struts 1.1 Beta 2 Web projects in IBM WebSphere Studio Site Developer Version 5, you might wish to convert it to Struts 1.1 but this is not required as Struts 1.1 Beta 2 is still supported. If you do have Struts 1.1 Beta 2 Web projects that you wish to convert to Struts 1.1, you will need to do the following:

1. Load your Struts 1.1 Beta 2 projects into a IBM WebSphere Studio Site Developer Version 5.1.1 workspace.
2. Create a new Struts 1.1 Web project named **Struts11**. This provides convenient access to the Struts 1.1 artifacts we will need while we are converting our real projects. You can delete this project when you are done.
3. For each Struts 1.1 Beta 2 project that you want to convert to Struts 1.1, do the following:
   a. Delete the following .jar files from your project's Web Content/WEB-INF/lib directory: commons-*.jar and struts.jar.
   b. Copy the following .jar files from Struts11/WebContent/WEB-INF/lib directory to your project's Web Content/WEB-INF/lib directory: commons-*.jar and struts.jar.
   c. Delete the following .tld files from your project's Web Content/WEB-INF directory: struts-*.tld.
   d. Copy the following .tld files from Struts11/WebContent/WEB-INF directory to your project's Web Content/WEB-INF directory: struts-*.tld.

All the above information is also applicable if you are moving a Struts 1.1 Beta 3 Web project in IBM WebSphere Studio Site Developer Version 5.0.1 to Struts 1.1.

Changes to Web services tools

The Web services tools has added two new IBM WebSphere Application Server Version 5.0.2 run-time protocols that run only on WebSphere Application Server Version 5.0.2. There should be no mandatory migration since IBM WebSphere Studio Site Developer Version 5.1.1 and WebSphere Application Server Version 5.0.2 will support both the old and the new run-time protocols. IBM WebSphere Studio Site Developer will generate and deploy three run-time protocols of Web service artifacts: The old ″IBM SOAP″ run-time protocol that runs on WebSphere Application Server Version 4.x and Version 5.x; the new ″IBM WebSphere 5.0.2

runtime″ run-time protocol that runs only on WebSphere Application Server Version 5.0.2; and the new ″Apache Axis 1.0″ run-time protocol that runs only on WebSphere Application Server Version 5.0.2.

Users should be able to reuse their Version 5 projects and Web services related EAR and WAR files with no changes in Version 5.1.1. In order for them to convert their Web services and clients to the new IBM WebSphere 5.0.2 run-time protocol and take advantage of the JSR 101, 109, WS-I and WS-Security standards, they will have to regenerate through the Web services wizard. The Web services explorer will automatically continue to read the user's favorites although the physical data file will be moved automatically to a new location.

Changes made in Profiling tools

When you migrate a workspace from Version 5, you will receive a ″Problems occurred restoring workbench″ pop-up error message. This message appears if the Profiling perspective is open at the time of migration and if the Heap or Instance Statistic views were visible in the Profiling perspective. This is because the Heap view and the Instance Statistic view that existed in Version 5 have been removed. This message also appears if the Profiling perspective is open in the workspace at the time of migration. The error message can safely be ignored by clicking **OK**.

Known compatibility problems of the Template Wizard

In order to use a customized template that was created in Version 5, you should load the customized template, reconnect it to the database and save it. Next time you reload the saved customized template the connection is verified.

Our generated J2EE 1.2 artifacts created in this release may not be read by IBM WebSphere Studio Site Developer Version 4.0.3 and run on Version 4.0.3 test environments. As the Template wizard was not delivered with Version 4.0.3, we do not maintain backwards compatibility to that version.

Template applications generated in this release can run on Version 5 if in the Web project preferences the Java source folder is named to ″Java Source″ and the Web content folder is named to ″Web Content″.

Chapter 5. Migrating from IBM WebSphere Studio Site Developer Version 4.0.x

This chapter covers migrating from IBM WebSphere Studio Site Developer Version 4.0.x to Version 5.

There are two supported methods that you can use to migrate your projects from IBM WebSphere Studio Site Developer Version 4.0.x to Version 5. Each of these methods is described in greater detail, below:

- Using a software configuration management (SCM) system such as Concurrent Versioning System (CVS) or Rational® ClearCase®. This is the recommended method.
- Exporting your projects from Version 4.0.x and then importing them to this edition. This method migrates everything except project build path information.
- Using your existing Version 4.0.x workspace. This is not supported for reasons explained later.

Note that migrating from Version 4 to Version 5 does *not* automatically change the project J2EE level since Version 5 can still build and deploy to WebSphere Application Server Version 4. All J2EE project types, including Web projects, can be migrated using the J2EE Migration wizard available in IBM WebSphere Studio Site Developer. To access the J2EE Migration wizard, right-click on a J2EE-type project, and then select **Migrate > J2EE Migration Wizard**.

Differences between IBM WebSphere Studio Site Developer Version 4.0.x and Version 5

The following is a partial list of enhancements since Version 4.0.x:

- IBM WebSphere Studio Site Developer Version 5 can generate code for either WebSphere Application Server Version 4.0 or Version 5, and includes both WebSphere Application Server Version 4.0.4 and Version 5 Test Environments.
- The enterprise applications archive's (EARs) J2EE level has changed from 1.2 to 1.3 for WebSphere Application Server Version 5 projects.
  - J2EE 1.2 EARs will run on either WebSphere Application Server Version 4.0.x or WebSphere Application Server Version 5.
- The Enterprise Java Beans (EJB) specification level has changed from 1.1 to 2.0 for WebSphere Application Server Version 5 EARs.
  - are still supported, and may be part of either WebSphere Application Server Version 4.0.x J2EE 1.2 EARs or Version 5 J2EE 1.3 EARs.
- The Web applications (WARs) J2EE level has changed from 1.2 to 1.3 for WebSphere Application Server Version 5 projects.
  - The JSP level has changed from 1.1 to 1.2 and the Servlet level has changed from 2.2 to 2.3
  - Dynamic 1.2 Web projects (WARs) are still supported, and may be part of either WebSphere Application Server Version 4.0.x J2EE 1.2 EARs or Version 5 J2EE 1.3 EARs.
- In Version 5, you can create static Web projects as well as dynamic Web projects. In a static Web project you will only be able to create content served by a traditional HTTP server (HTML, JavaScript™, images, text and so on).

- The underlying workbench, which is based on the Eclipse open-source project, has changed from Version 1.0 to Version 2.0.
  - There is a new and much improved Java builder
  - There is a new and much improved version control management (VCM) interface for software configuration management (SCM) vendors

WebSphere Application Server changes and Servlet/JSP conversion tools

The WebSphere InfoCenter [www.ibm.com/software/webservers/appserv/doc/v40/aes/infocenter/index.html] has the following information:

- Differences between WebSphere Application Server Version 3.5 and 4.0 [www.ibm.com/software/webservers/appserv/doc/v40/aes/infocenter/was/03.html].
- For information on migrating to WebSphere Application Server Version 5.0 from Version 3.5 or Version 4.0, go the WebSphere InfoCenter at www.ibm.com/software/webservers/appserv/infocenter.html and follow the steps below to the navigate to the migration page.
  1. In the first link **Application Server for distributed operating systems** select a language of your choice.
  2. Select the link **All topics by activity** that is located in the upper left hand corner.
  3. Select the link **Migrating**.

Migrating to WebSphere V5.0 An End-to-End Migration Guide is good resource for information on migrating from Version 3.5 and Version 4.0 to Version 5 [www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246910.pdf].

The WebSphere Application Server downloads page [www14.software.ibm.com/webapp/download/product.jsp?s=p&id=TDUN-49EVRT&type=s&dt=DIAGNOSTIC+TOOL] has tools to help convert your application:

- MigrateWC takes a .91 or 1.0 JSP and converts it to a 1.1 JSP. It also takes a 2.1 Servlet and converts it to a 2.2 Servlet.
- XMLconvert converts XML configuration files from Release 3.02x or Release 3.5x to Release 4.0 format.

Internal changes from Version 4.0.3

Circular project dependencies will not build by default

If your projects have circular dependencies, Version 5 reports a build error. You can go into **Window > Preferences > Java > Compiler**, select the **Build Path** tab, and clear the **Abort building on build path errors** check box. Note that this will no longer cause the build to stop, but there will still be one or more build 'circular dependency' errors shown on the Task view (even when the build otherwise is successful). In this case, you may change these errors to warnings by selecting the **Other** tab, and then changing the preference in the **Circular Dependencies** drop-down.

Version 5 Web projects are source location compatible with Version 4.0.3

In IBM WebSphere Studio Site Developer Version 5, there are internal project structure changes from Version 4.0.3. A Version 5 J2EE 1.2 Web WAR, when exported with Java source, will import into IBM WebSphere Studio Site Developer Version 4 and the source code folder will be automatically converted into the right name and will build fine. The Web project still executes correctly on WebSphere Application Server Version 4 similarly when a Version 4 project is imported into Version 5, because the source code folder is automatically converted to the correct name. For more information about folder name changes, refer to "IBM WebSphere Studio Site Developer Web project structures" on page 29

Note: The above is not true if the Web projects are shared between the Version 5 and Version 4 through a software configuration management (SCM) system. The Version 4 projects need to be migrated to Version 5 project structure and cannot be loaded back into Version 4 from a SCM system once migrated.

IBM WebSphere Studio Site Developer Web project structures

The internal Web project structure in IBM WebSphere Studio Site Developer Version 5 is different than it was for IBM WebSphere Studio Site Developer Version 4.0.x. This difference is not related to J2EE 1.2 versus J2EE 1.3, but rather it is a tool usability change.

In Version 4, Web projects were dynamic Web projects by default and they appeared in the Navigator view with a **source** folder and a **webApplication** folder. In Version 5, if you create a dynamic Web project, then it will appear with a **Java Source** folder instead of a **source** folder and a **Web Content** folder instead of a **webApplication** folder.

However, if a Version 4 Web project is saved into an SCM repository and then loaded into Version 5, it will retain the old structure with the **source** and **webApplication** folders. Either structure will build correctly in Version 5.

Static versus dynamic Web projects

In Version 5, you can create static as well as dynamic Web projects.

Static Web projects contain only static resources like HTML, Java Scripts, images, text and so on, and no dynamic content in them. Static Web projects can run and be served by a traditional HTTP Web server and do not need a Web Application Server.

Dynamic Web projects contain dynamic J2EE resources such as servlets, JSPs, filters, and associated metadata, in addition to static resources. When you create dynamic Web projects, you can include cascading style sheets and JSP tag libraries, so that you can begin development with a richer set of project resources. Dynamic Web projects are always imbedded in Enterprise Application projects and run only on Web Application Servers.

HTML and JSP distinctions

- In Version 4.0.x, HTML files and JSP files were treated identically by Page Designer. For example, you could have JSP tags in an HTML file. This is no longer true: in this release, there is a distinction between JSP and HTML files, so you can no longer have JSP tags in an HTML file.

- The previous distinction affects encoding of non-English JSP files. In versions prior to this release, HTML encoding rules were used, even for JSP files, to determine the encoding named in a file. That is, the content type attribute of the meta tag was looked at (`<META http-equiv="Content-Type" content="text/html; charset=UTF-8">`). In this version, this was changed to use JSP encoding rules to determine the encoding named in a JSP file. That is, the page directive of the JSP file is looked at (`<%@page contentType="text/html;charset=UTF-8"%>`). For HTML files, encoding is unchanged from previous versions.

Migrating projects using a software configuration management (SCM) system

Migrating projects using CVS or Rational ClearCase

This is the recommended way to move workspaces from Version 4.0.x to IBM WebSphere Studio Site Developer Version 5. This is the only method that migrates all of your information, including project build path information.

1. As a backup precaution, save all your Version 4 projects into your SCM repository. Then commit (release) any pending changes.
2. If you wish to work on both Version 4 and Version 5 of IBM WebSphere Studio Site Developer, save your work again into a new Version 5 branch (stream). This is the branch that you will use when working with Version 5.
3. Install the Version 5.
4. Close IBM WebSphere Studio Site Developer Version 4 and start IBM WebSphere Studio Site Developer Version 5.

   Tip: In Version 4, the workspace directory was located in the installation directory, by default. In Version 5, this default has changed to a directory called **workspace** in the My Documents directory. If you wish to override the location where your work is stored, use the `-data` option on the command when you start the workbench.

   Note: Do not use `-data` to point to an existing Version 4 workspace since that is a different unsupported approach to migration. (For more information, refer to the "Migrating projects using an existing Version 4.0.x workspace" on page 32.)
5. Disable **Windows > Preferences > Workbench > Perform build automatically on resource modification** (to avoid build errors as individual dependent projects are loaded).
6. **For CVS:** Load all of the projects that you want to work with from the SCM repository into IBM WebSphere Studio Site Developer Version 5.

   For ClearCase: Use a clean Version 5 workspace, then for each project to be loaded, select **File > Import > Existing WebSphere Studio 4.x ClearCase Project into Workspace**.
7. Restore your desired setting for **Windows > Preferences > Workbench > Perform build automatically on resource modification**.
8. Change the **source** folder name from **source** to **Java Source** and **webApplication** folder to **Web Content** for the Web projects if a full build is needed. Otherwise, the old folder structure is retained and the web projects will *not* be fully rebuilt.
9. Do a full rebuild (**Project > Rebuild all**), and save the resulting projects back into your repository in your new Version 5 stream. (Do *not* mix these resources with your ongoing Version 4 stream.)

Note: These projects are now Version 5 projects and *cannot* be used by IBM WebSphere Studio Site Developer Version 4.0.x.

Post migration considerations:

- In IBM WebSphere Studio Site Developer Version 4.0 the files in a project were stored in binary format in CVS. In binary format carriage returns/line feed in files are *not* translated for use in other platforms and *cannot* be recognized in UNIX®/Linux platform environments. If you are working in mixed platform environments, for example, the DOS/Windows and UNIX/Linux platforms, you might wish to now mark source files as text (using **Team > CVS > Change ASCII/Binary Property**) and resave them in CVS.
- Version 4 Web projects from a CVS repository require **Window > Preferences > Team > CVS > Prune empty directories** setting to be disabled (the default is that it is enabled). If it is *not* disabled, and you load a Web project with an empty **source** folder (like the MyHomePage Web example), then you will get the following errors at check in:

  ```
  The project was not built since it is involved in a cycle or has classpath problems.
  Missing required source folder: /MyHomePageExample403/source.
  ```

- For Web projects saved and loaded from a ClearCase repository, you need to have a file checked out before you can open it in the editor. If it is not checked out you receive errors `error activating this view` (Null pointer exception in logs for Page Designer). With an xml editor, editing a web.xml file you need to have web.xml, ibm-web-bnd.xmi, and ibm-web-ext.xmi checked out. (There are indications that you need these files to be checked out on the status line, which states that they are read only, but they are easily missed.)
- If your projects have circular dependencies, Version 5 reports a build error. You can go into **Window > Preferences > Java > Compiler**, select the **Build Path** tab, and clear the **Abort building on build path errors** check box.
- The .vcm_meta (or .cc_meta) files could now be deleted from the Version 5 project because they are not used by Version 5 (it uses a new .project file instead) and because you are using a new repository branch (stream) for these Version 5 projects. Note that these files are still needed in the ongoing Version 4 branch (stream).

Post-Migration removal of EAR and Server Configuration absolute path references

Version 4 EAR IBM application extension files and server configuration files contained absolute path references. After you have migrated them into Version 5, you need to open them with their editor (which will automatically change their old absolute path references into new relative references).

1. For each EAR project, in a Navigator View, right-click **META-INF/application.xml > Open with > Deployment Descriptor Editor**.
   a. A dialog window pops up with the message:

      ```
      The IBM extensions file contains deprecated absolute paths.
      This can be auto-corrected and should be saved. This will
      remove the paths from the file, and only needs to be done once.
      Would you like to autocorrect?
      ```

   b. Click **Yes**.
   c. Save and then close the editor window.

Note: Alternatively, you can use the J2EE Migration wizard to migrate the project structure only for an EAR project. To access the J2EE Migration wizard, right-click on the EAR project and then select **Migrate > J2EE Migration Wizard**.

2. For each Server configuration, in a Server Perspective, Server Configuration View, right-click on the server, and then select **Open**.
   a. You will get a similar autocorrect dialog.
   b. Click **Yes**.
   c. Save and then close the editor window.

Migrating projects using other SCMs

There are other SCM vendors who provide SCM plug-ins for IBM WebSphere Studio Site Developer. You may browse the list of vendors at www.ibm.com/software/ad/studioappdev/partners/scm.html. As part of their *Ready for IBM WebSphere Studio software* [www.developer.ibm.com/websphere/ready.html] validation, all SCM vendors who provided a Version 4 plug-in will have ensured that the preceding migration steps (save from Version 4 to SCM repository, load from repository into Version 5) also work for their systems.

Migrating by exporting and importing your projects

1. In IBM WebSphere Studio Site Developer Version 4.0.x, export your projects to a WAR file, an EAR file, or a JAR file (**File > Export**).
2. In IBM WebSphere Studio Site Developer Version 5, import your WAR file, an EAR file, or a JAR file (**File > Import**).

Note: This is not a full migration since no project build path information is maintained.

Migrating projects using an existing Version 4.0.x workspace

This approach is partially supported, and will result in an incomplete migration. User interface settings, debug settings, and most preferences are all lost. Project names, project source files, and project Java build path (class path) are retained, but nothing else is guaranteed. This approach should only be used if no supported SCM system is being used and if it is critical to retain project build path information, which is lost when you import projects that were exported from Version 4. You can use the existing Version 4.0.x workspace by doing the following:

1. Commit (release) any pending changes to the repository.
2. Close all perspectives and shutdown IBM WebSphere Studio Site Developer Version 4.
3. Back up the contents of *workspace_directory*, where *workspace_directory* is the fully qualified directory name that contains the Version 4.0.x workspace. By default, the Version 4.0.x workspace subdirectory is located in the same directory where the product is installed. You will need this backup if you ever want to work with IBM WebSphere Studio Site Developer Version 4.0.x again. Once you have pointed to a Version 4.0.x workspace from a Version 5 IDE, you can no longer go back to using that workspace in IBM WebSphere Studio Site Developer Version 4.0.x.
4. Install IBM WebSphere Studio Site Developer Version 5.
5. When you start IBM WebSphere Studio Site Developer Version 5 with a Version 4.0.x workspace from a command prompt (that is, use the `-data`

option to specify a fully qualified path to the Version 4.0.x workspace directory on the command), that will cause an upgrade of the .metadata information.

6. When prompted to confirm that you wish to convert to the new user interface format, click **OK**.
7. Before doing any rebuilds or validating any projects that are in the workspace, select all of the projects in the Navigator view within the Resource perspective and then select **Refresh** from the pop-up menu. This will ensure that all files are synchronized with their appropriate metadata.
8. Open any closed projects (see known problems below).
9. Verify your class path variables (see known problems below).
10. Some builders and validators have been added, removed, or modified in this Version 5. To ensure that the correct errors and warnings are shown, you must rebuild all the projects by selecting **Project > Rebuild All**, and then select **Run Validation** for each Java project.
11. Some user preferences might be maintained, but many others will not be. Check your preference settings in Version 5 to be sure that they are as you want them.

Post-Migration removal of EAR and Server Configuration absolute path references

The post-migration instructions described in "Post-Migration removal of EAR and Server Configuration absolute path references" on page 31 also apply here.

Known problems and limitations

The following problems may occur if you attempt to migrate by opening a Version 4.0 workspace in IBM WebSphere Studio Site Developer Version 5.

Incorrect value in the JRE_LIB class path variable

To reset your JRE_LIB class path variable to a valid location, follow these steps. *Do this even if the value seems correct* when you first open the Preferences window.

1. Select **Window > Preferences > Java > Installed JREs**.
2. In the list, select the check box for the default JRE location that you wish your JRE_LIB set to.
3. Choose **Edit**, and then click **OK** to close the Edit JRE dialog box.

If you do not do this, the value for JRE_LIB might be incorrect, causing many build errors in Java files.

As a general check, verify the value of all your other class path variables.

For previously SCM shared projects, the Team menu contains Share Project

Team support has changed significantly between Eclipse 1.0 and 2.0. The method of sharing projects with the repository has changed as well.

- If you right-click on the project and then select the **Team > Share Project** option, a wizard will guide you through the migration process. When you are finished, your project will be shared and the Synchronize view will open. You will see conflicting changes on every file. This is due to changes in the way sync information is stored between Eclipse 1.0 and 2.0.
- If you do not have any outgoing changes (which you should not have if you committed all your outgoing changes before upgrading as recommended above),

then you can simply select the project in the Synchronize view and select **Override and Update** which will load the current contents from the server.

- If you do have outgoing changes, you can pull down the triangle menu in the Synchronize view and select **Compare File Contents**. After some work, the Synchronize view will show you only the files which are actually different. You can then use the Synchronize view to resolve these conflicts.

Projects created outside the workspace directory

By default, projects are created in the workspace directory. If you overrode the default to create projects elsewhere, you must open all of your projects before closing the workbench. This will allow the .project file for that project to be written in the proper location. Failure to open a closed project whose directory is outside of the workspace will result in a project that masks the actual project, with only a .project file existing within it.

JSP breakpoints must be reset

You will need to remove any JSP breakpoints that you have, and reset them within the migrated Version 5 workspace.

Migrating relational data in 4.0.3 Web projects

To migrate relational data from IBM WebSphere Studio Site Developer 4.0.3 projects:

1. From a IBM WebSphere Studio Site Developer 4.0.3 workspace, generate DDL files for each available database.
2. Remove the database from the Web Project source/databases folder (via Data Definition view).
3. Open the 4.0.3 workspace with IBM WebSphere Studio Site Developer Version 5.
4. Migrate the Web projects for which you wish to restore relational data.
5. Click **File > Import > File System**, and specify the DDL files from your 4.0.3 workspace.
6. In the Data Definition view of the Data Perspective, select **Run against Local**, and specify the target Web project.

The relational data artifacts will be restored.

WSDL errors after importing a Web services file from 4.0.x

If you have imported a Web services file from 4.0.x, you may receive the following error messages:

```
Error The part 'result' has an invalid value 'anyElement'
defined for its type. Type declarations must refer to
valid values defined in a schema.

Error The part 'return' has an invalid
value 'findPatientResult' defined for its element.
Element declarations must refer to valid values
defined in a schema.

Error The part 'response' has an invalid
value 'findPatientResponse' defined for its element.
Element declarations must refer to valid values
defined in a schema.
```

The workaround is:

1. Delete the WSDL files.

2. Regenerate your Web services by rerunning the Web Services wizard.

Migrating J2EE project structures and/or J2EE specification levels

To access the J2EE Migration wizard in Version 5, follow the steps below:

1. Select the project.
2. Right-click on it and then select **Migrate > J2EE Migration Wizard**. Follow the steps in the wizard to guide you through migration.
3. If your project is under source control, then save the restructured project in your SCM.

Chapter 6. Migrating from WebSphere Studio ″Classic″ to IBM WebSphere Studio Site Developer

This chapter documents how to migrate from WebSphere Studio Version 4.0 (both Advanced and Professional Edition) to the IBM WebSphere Studio Site Developer . Migrating from WebSphere Studio ″Classic″ Version 4.0 to IBM WebSphere Studio Site Developer Version 5.0 involves the following steps:

1. Create a new single-server stage for migration.
2. Create a Web configuration descriptor file.
3. Export a migration JAR file.
4. Import the migration JAR file into IBM WebSphere Studio Site Developer .
5. Set up your server and test your migrated application.

Note: The following instructions are for migrating from WebSphere Studio Version 4.0. If you want to migrate from an earlier version of WebSphere Studio, you should migrate to WebSphere Studio 4.0 first, then migrate to IBM WebSphere Studio Site Developer .

The advanced publishing feature (mapping files to publishing stages) and the Page Detailer feature (analysis of web pages) of WebSphere Studio ″Classic″ is not available in IBM WebSphere Studio Site Developer . Some other features from the Version 4.0.x CD media pack are also no longer available. For example, the Page Detailer feature for analysis of web pages, the HotMedia® feature for rich media types, the Voice XML editor (moved to WebSphere Everyplace™ Toolkit and Portal Toolkit), DataBaseWizard for pervasive devices.

You should be aware of the following limitations before you migrate any of your WebSphere Studio data:

- IBM WebSphere Studio Site Developer uses an XML-based SQL editor, so your .sql files cannot be used in it.
- Project publishing information and stage information cannot be migrated.
- WebSphere Studio server configuration information cannot be migrated.
- Version control information cannot be migrated.

During the migration process outlined below, WebSphere Studio creates a JAR file that contains all of your project files, publishable and source, for a single server. All the files visible in the Publishing view for the default server will be packaged into the JAR file. You can then import the JAR file into IBM WebSphere Studio Site Developer .

When you migrate existing projects, all the project publishing information and the stage information are lost during the migration. If your stage has multiple servers, only files published to the default server are kept. Therefore, for the purpose of migration, you will create a new stage that has only one server.

Creating a new single-server stage for migration

If you have more than one server in your current stage, create a new stage called Migration with only one server by following these steps:

1. Click **Project > Customize Publishing Stages**.
2. Type `Migration` in the **Stage name** field.
3. Click **Add**.
4. Click **OK**.
5. Click **Project > Publishing Stage** and select **Migration** from the list of available stages.
6. While in the publishing view, click **Insert > Server**.
7. Type a server name, such as `localhost`.
8. Changing the server or changing the publication stage does not propagate the servlet mapping information for WebSphere Application Server Version 4.0. Go to the Publishing view, and, for each servlet, click **Properties > Publishing > Servlet Mapping** and then copy the actual servlet mapping.

Creating a Web configuration descriptor file

1. While in the project file view, click **Project > Create Web Configuration Descriptor File**.
2. Select all required servlets.
3. Select all required Tag Library Descriptor (TLD) files.
4. Click **Create**.

The default Web configuration descriptor file name is *serverName_web.xml*, localhost_web.xml in this scenario. Unless you specified a different location, the .xml file is saved in the WEB-INF directory.

Exporting a migration JAR file

1. While in the project file view, select server **localhost** and click **Properties > Publishing > WebApp Web Path** and enter a web path (context root), such as `myWebPath`. This will also be used as the WebSphere Application Server - Express project name.
2. While in the project file view, select **Project > Create Migration file**.
3. Verify that **localhost** is the selected server.
4. Verify that **localhost_web.xml** is the selected Web configuration descriptor file.
5. Click **OK**.
6. The default JAR file name is `serverName.jar`, localhost.jar for this scenario. Rename the file if desired.
7. Save the JAR file.

Importing the migration JAR file into IBM WebSphere Studio Site Developer

1. Start IBM WebSphere Studio Site Developer .
2. Create a Web project (**File > New > Project > Web Project**).
3. In the **Project name** field, type the name of your Web project. This should be the same name you specified in step 1 of the preceding “Exporting a migration JAR file” on page 38 section.

4. Specify the name of a new or existing EAR project that will contain the new Web project for purposes of deployment.
5. In the **Context Root** field, type the Webapp Web Path name you specified when you created the migration JAR file in WebSphere Studio. Click **Finish**.
6. In the Navigator view, select the Web project you just created.
7. Import the JAR file.
   a. Click **File > Import**.
   b. Click **WAR file**. Click **Next**. You *must* import the JAR file using the WAR file option; otherwise it will not work properly.
   c. Enter the path to localhost.jar in the **WAR File** field or click **Browse** to search for it. (You can only browse for a .WAR name, not a .JAR name.)
   d. Select the existing Web project that you created. The **Context Root** field is automatically populated with the value you specified earlier.
   e. Click **Finish**. A dialog appears asking "`Resource WEB-INF/web.xml already exists. Would you like to overwrite it?`".
   f. Select **Yes** and IBM WebSphere Studio Site Developer unpacks *localhost.jar*.
8. You may have several unresolved references or missing import files. These will appear in the Tasks view. To fix this, you must change the Java build path for the Web project:
   a. Right-click the project and click **Properties > Java Build Path.**
   b. Click the **Libraries** tab. Click **Add External JARs**.
   c. Import any JARs that you need from the following directories:
      - WS_Installdir/runtimes/aes_v4/lib and
      - WS_Installdir/runtimes/base_v4/lib
9. In the Navigator view, right-click the project and select **Rebuild Project**.

Testing your migrated application on a test server

You are now ready to test your application. To test it on the default test server, follow these steps:

1. Right-click the EAR project.
2. Select **Run on Server**

To test your application on other server run-time environments, refer to the online help for the Server Tools feature.

Chapter 7. Migrating from VisualAge for Java to IBM WebSphere Studio Site Developer

This chapter documents how to migrate from VisualAge® for Java Professional Edition or VisualAge for Java Enterprise Edition to IBM WebSphere Studio Site Developer.

Note: The instructions given in this chapter are for migrating from VisualAge for Java Version 3.5.3 or 4.0 for Windows. If you want to migrate from an earlier version of VisualAge for Java to IBM WebSphere Studio Site Developer, you should first migrate from your earlier version of VisualAge for Java to Version 3.5.3 or 4.0 for Windows, before migrating to IBM WebSphere Studio Site Developer.

Note: Windows Instantiations, Inc., an IBM Business Partner, distributes a product, called CodePro Studio that provides productivity enhancements to VisualAge for Java and WebSphere Application Server - Express, including migration and co-existence facilities. To help VisualAge for Java customers begin their migration, Instantiations is offering a free, unlimited use VisualAge for Java to IBM WebSphere Studio Site Developer export facility as part of their time-limited evaluation copy of CodePro Studio. You can download the evaluation copy from www.instantiations.com/vaj-migrate. For further information on Instantiation's advanced migration and co-existence support including full bi-directional export/import of files, creation of export/import sets, project synchronization and task automation, please browse Instantiations, Inc. www.instantiations.com/codepro/ws.

Differences between VisualAge for Java and IBM WebSphere Studio Site Developer

The following is a partial list of changes from VisualAge for Java:

- The Enterprise Java Beans (EJB) specification level has changed from 1.0 to 1.1 (EJB 2.0 is also supported for applications that will be deployed to WebSphere Application Server Version 5).
- For Web applications, the JSP level remains at 1.1 (1.2 for WebSphere Application Server Version 5 applications).
- For Web applications, the Servlet level remains at 2.2 (2.3 for WebSphere Application Server Version 5 applications).
- The level of the Java 2 platform that is supported has changed from 1.2 to 1.3. (The compiler can target 1.4 code generation, but the WebSphere Application Server run-time environment is still 1.3.)
- The Visual Composition Editor has been replaced by the Visual Editor for Java.
- VisualAge for Java version control and the propriety source code repository have been replaced by support for software configuration management (SCM) plug-ins.
- The VisualAge for Java Tools API has been replaced by the WebSphere Studio Workbench plug-in architecture.
- The VisualAge for Java XML tools have been replaced by IBM WebSphere Studio Site Developer XML tools.

- The VisualAge for Java project concept has been replaced by multiple types of IBM WebSphere Studio Site Developer projects.

Migrating from VisualAge for Java

The following steps outline how to migrate from VisualAge for Java. Details on how to perform these steps are provided below:

1. Export your Java files and project resource files from VisualAge for Java.
2. Start IBM WebSphere Studio Site Developer and create new projects to contain your code.
3. Import your Java and project resource files into IBM WebSphere Studio Site Developer .
4. Use the web.xml editor to ensure that any servlets are correctly defined (Web project only).
5. Migrate your project and workspace settings.
6. Set up your server and test your migrated application(s).
7. Deploy your applications from IBM WebSphere Studio Site Developer to WebSphere Application Server.
8. Share the IBM WebSphere Studio Site Developer project settings between multiple developers (post-migration).

Exporting your Java files and project resource files from VisualAge for Java

There is no support for the bulk migration of versioned projects and resources from the VisualAge for Java repository. You can only migrate projects and resources that are in your VisualAge for Java workspace. If you want to migrate a versioned copy of a project or resource into IBM WebSphere Studio Site Developer, you must bring it into your VisualAge for Java workspace and then migrate it.

Note: If your project contains more than one kind of data (for example, enterprise beans and Java source code files), you should split up your data into different JARs based on their type.

Export your projects to a JAR file by following these steps:

1. If the projects that you want to export are not currently in your VisualAge for Java workspace, add them to the workspace now.
2. In the VisualAge for Java Workbench window, select your project, right-click, and then click **Export**.
3. Select the **JAR file** radio button and click **Next**.
4. Specify the name of the JAR file.
5. Select the **.java** check box to export your Java files and the **resources** check box to export your resource files.
6. Fill in the other fields as required. Refer to the VisualAge for Java online help for more information on how to perform this task.

Starting IBM WebSphere Studio Site Developer and creating new projects to contain your code

Start IBM WebSphere Studio Site Developer, then create the appropriate projects. The following is a set of general migration guidelines to help you decide which kind of IBM WebSphere Studio Site Developer project you should import your files into:

- If your code is part of a Web application, you should import the code into a Web project:
  - Import all Java files into the Web project **Java Source** directory (the proper hierarchy based on their package statements will automatically be created by IBM WebSphere Studio Site Developer)
  - Import all resource files into the Web project **Web Content** directory.
- If your code is straight Java, (for example, an application that will run stand-alone) you should import the code into a Java project.

Note: The preceding is only a general set of guidelines to help you decide which kind of IBM WebSphere Studio Site Developer projects you should use. We recommend that you read the IBM WebSphere Studio Site Developer online help and become familiar with the different kinds of WebSphere Application Server - Express projects before you create any projects or import any code.

Importing your Java and resource files into IBM WebSphere Studio Site Developer

1. Open IBM WebSphere Studio Site Developer and switch to the Resource perspective.
2. Click **File > Import > Zip file**. Click **Next**.
3. Browse to the appropriate JAR file.
4. Select the files you want to import and the project or folder you want to contain your files.

Note:

- The FileSystem import can also be used instead of the ZIP import, but ZIP import is more commonly used. You can use the same instructions even for a file import and point to a folder rather than a ZIP file in the File Import wizard.
- When you import your files into IBM WebSphere Studio Site Developer, you should ensure that they go in the appropriate directory. We recommend that you read the IBM WebSphere Studio Site Developer online help and become familiar with the different kinds of IBM WebSphere Studio Site Developer projects before importing your code. This will help you determine which folders should contain which kind of code.

Using the web.xml editor to ensure that servlets are correctly defined (Web project only)

If your application uses servlets, then you need to define the servlet-URL mappings in the web.xml file. Follow these steps:

1. In the Web perspective, open the web.xml file, which is located in the Web Content/WEB-INF subdirectory of your Web project.
2. Click the **Servlets** tab.
3. Click **Add**, and select the **Servlet** radio button.
4. Type the servlet name and click **OK**.
5. Click **Browse** to change the **Servlet class** value to the appropriate package name.
6. (Optional) The display name is a short name used to identify the servlet. In the **Display name** field, type a short name for the servlet.

7. A URL mapping defines a servlet and a URL pattern. Click the **Add** button located next to the **URL mappings** field, then type the name of the URL mapping.
8. Save the changes (**File > Save web.xml**) and close the web.xml file.

Migrating project and workspace settings

You must record the following VisualAge for Java settings and set them up in IBM WebSphere Studio Site Developer :

- Project class path
- Resource associations
- Code formatting
- WTE configuration
- Java files and project resource files

Project class path

In VisualAge for Java, you set your project class path in the Resources pages of the Options window (**Window > Options > Resources**). After you have migrated your projects into IBM WebSphere Studio Site Developer , you can set up your project's class path in the project's Properties window (Right-click the project and select **Properties > Java Build Path**. Click the **Libraries** tab.) You can also set class path variables in the Preferences window (**Window > Preferences > Java > Classpath Variables**.)

Resource associations

If you set up an association between a file type and an executable, you can open a file that sits outside the workbench from within it.

In VisualAge for Java, you set up your resource associations in the Options window (**Window > Options > Resources > Resource Associations**). After you have migrated your resource files to IBM WebSphere Studio Site Developer , you can set up your resource associations using the Preferences window (**Window > Preferences > Workbench > File Associations**).

Code formatter

In VisualAge for Java, you set up your code formatting options in the Formatter page of the Options window (**Window > Options > Coding > Formatter**). After you have migrated your code to IBM WebSphere Studio Site Developer, you can set up your code formatting in the Preferences window (**Window > Preferences > Java > Code Formatter**).

WTE configuration

In VisualAge for Java, your WebSphere Unit Test Environment and WebSphere Application Server run time settings are in various property files in the following directory: *VisualAgeInstalldir*\ide\project_resources\IBM WebSphere Test Environment\properties, where *VisualAgeInstalldir* is your product installation directory.

If, for example, you have enabled URL rewriting in the session.xml property file by changing the property to true as shown below, `<url-rewriting-enabled>true</url-rewriting-enabled>` you can configure this property in the IBM

WebSphere Studio Site Developer Version 4.0 Test Environment. (In the Server perspective, open the Server Configuration view, right-click the server you want to work with and click **Open**. Click the **Web** tab and select the **Enable URL rewrite** check box).

Java files and project resource files

The property file default.servlet_engine contains the <root-uri> context root of the VisualAge for Java web application(s). When creating a Web Project in IBM WebSphere Studio Site Developer the **Create a Web Project** dialog contains a **Context root** field for this data.

Web application settings in files such as *VisualAgeInstalldir*\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host\default_app\servlets\default_app.webapp that you have customized yourself should be migrated to *your_Web_project*\Web Content\WEB-INF\web.xml file in IBM WebSphere Studio Site Developer. For example, if you have changed servlet names and servlet paths in the default_app.webapp file, you would make the corresponding changes in your web.xml file.

Setting up your WebSphere V4 test environment and testing your migrated application(s)

If the application is a Java project, then you just use the normal IBM WebSphere Studio Site Developer **Run** or **Debug** support for Java projects to test it.

If the application uses WebSphere Application Server , then test it using the built-in WebSphere Application Server. This requires that a default test server be created and started. For a Web project, right-click on the main HTML page, and select **Run on Server** to launch the web browser.

For information on testing other types of projects, refer to the online help.

Deploying your applications from IBM WebSphere Studio Site Developer to remote WebSphere Application Server

If you are using the WebSphere Application Server as your run-time environment, deploy your application using the Server Tools feature of IBM WebSphere Studio Site Developer .

Sharing IBM WebSphere Studio Site Developer project settings between multiple developers (post-migration)

IBM WebSphere Studio Site Developer projects (and their associated settings) can be shared between developers. To do this, save a project into the IBM WebSphere Studio Site Developer software configuration management (SCM) server, then extract it onto another team member on the IBM WebSphere Studio Site Developer.

Team support in IBM WebSphere Studio Site Developer

For information on team support in IBM WebSphere Studio Site Developer Version 4.0, refer to www.ibm.com/websphere/developer/library/techarticles/0108_karasiuk/0108_karasiuk.html

There is also information in the Installation guide and online help about team support in IBM WebSphere Studio Site Developer.

Chapter 8. Migrating from VisualAge for Java Visual Composition Editor to Visual Editor for Java

This chapter provides instructions on how to migrate applications created in the Visual Composition Editor feature of VisualAge for Java to the Visual Editor for Java within WebSphere Application Server - Express:

- Saving enhanced design-time metadata from VisualAge for Java
- Completing the migration (importing into WebSphere Studio)

Saving enhanced design-time metadata from VisualAge for Java

This step is optional but is highly recommended (especially if your application has any connections) for the following reasons.

- VisualAge for Java did not save information about the placement of top-level beans (that is, beans that are not contained inside other beans) on the free-form surface. By contrast, WebSphere Studio always saves this information as a comment on the line of code that declares the bean and restores the bean to that position on the canvas every time you open the visual editor. You have the option to save this design-time information in VisualAge for Java before migration. If you do not save it, when you first open your .java files in the new Visual Editor for Java the editor calculates a default position for top-level beans (also known as free-form parts), which you can easily change by means of a drag and drop operation.
- Although the new Visual Editor for Java in WebSphere Application Server - Express does not support connection design, it is possible that equivalent function might be added in the future. (**Note:** This is not a product commitment; it is just preparation for a future possibility.)

To save the enhanced metadata information prior to migration:

1. Go to the VisualAge for Java Developer Domain [www.software.ibm.com/vad/data/document4293] and download the *IBM VCE Code Generation and Export Utility*.
2. Following the tool readme, add the tool into VisualAge for Java and then stop and restart VisualAge for Java.
3. Version the current application code into the VisualAge for Java repository (so you can return to this version in case of any ongoing VisualAge for Java development).
4. For each of your graphical applications within VisualAge for Java, select one or more of the graphical programs (typically XxxxxView), right-click, and then do the following:
   a. Click **VCE Code Generation/Export**, and leave selected the **Export to a directory after code regeneration** option.
   b. Click **Finish**.
   c. Leave **Directory** as the selected export destination, click **Next**.
   d. Select your target directory, clearing the **.class** option and selecting the **.java** option (since you want the source code), and then click **Finish**.
   e. Optionally, reload your VisualAge for Java code with its previous version (right-click and then select **Replace with > Previous edition**).

Completing the migration (importing into WebSphere Studio)

Your classes are now ready to be imported into WebSphere Application Server - Express. Refer to earlier in Chapter 7, “Migrating from VisualAge for Java to IBM WebSphere Studio Site Developer,” on page 41. Once your previous Visual Composition Editor source programs have been imported into WebSphere Application Server - Express, you can maintain them in the Visual Editor for Java.

Chapter 9. Build setup (library, JARs, dependent project JARs, Ant builds)

This chapter covers the following migration topics:

- Java library JARs, and third-party external JARs
- Optimizing multi-project builds using dependent Project JARs
- Automated production builds using Ant

Java library JARs and third-party external JARs

For detailed explanations of what is involved, see the article on *J2EE Class Loading Demystified* (www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer.html) (J2EE modules and class paths) and on developing J2EE utility JARs (www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer2.html) (Java JARs in J2EE modules). These provide excellent technical background and advice.

Recommended way to use a third-party JAR within a Web project

The recommended way to use a third-party JAR file within a Web project is to import it (keeping it as a JAR file) into the library folder of your Web project. This is the only J2EE-defined, portable way to use a JAR file, and will ensure that you do not have to make any changes when deploying to another server.

To use an external JAR file in a single Web project, follow the steps below. If you need to use the JAR file in multiple Web projects, follow the steps described instead in "Recommended way to use a third-party JAR for use with multiple Web projects" on page 49.

1. Select **File > Import > File System**. Click **Next**. You must select **File system**, not **Zip file**, in order to ensure that the JAR file is not expanded when it is imported.
2. Click **Browse** and locate the JAR file directory.
3. Import it into your *WebProject*/WebContent/WEB-INF/lib folder, where *WebProject* is the name of your Web project.
4. Click **Finish**. The JAR file will be automatically added to the Java build path, and no further changes are required at run time.

Recommended way to use a third-party JAR for use with multiple Web projects

The recommended way to use a third-party JAR file with two or more Web projects is to import it (keeping it as a JAR file) into your Enterprise Application (EAR) project. This is the only J2EE-defined, portable way to use a JAR file, and will ensure that you do not have to make any changes when deploying to another server.

To use an external JAR file with multiple Web projects, follow the steps below. If you only need to use the JAR file in a single Web project, follow the steps in the previous section.

1. Select **File > Import > File System**. Click **Next**. You must select **File system**, not **Zip file**, in order to ensure that the JAR file is not expanded when it is imported.
2. Click **Browse** and locate the JAR file directory.
3. Import the JAR file into the Enterprise Application project that contains the Web projects.
4. Click **Finish**. The JAR file will automatically be added to the Java build path, and no further changes are required at run time.
5. Follow the steps in the following next section to add the JAR to the module dependencies of the Web project.

Alternative way to use external JAR files (global build and server classpath)

You may also leave the JAR file outside of WebSphere Application Server - Express and add it to both the Java build path and the server instance's class path. This is not recommended, because your application will not be easily portable. When you move to a different server, you will always have to update the server's class path. As well, you need to ensure that your class files do not conflict with other versions of similar class files already on the server classpath (and needed by the server or its other applications). If you do decide to take this approach, take the following steps:

1. Add the external JAR file to the Java build class path of the project that requires the JAR file.
   a. Select the project, right-click it, and select **Properties** from its pop-up menu.
   b. Click **Java Build Path**.
   c. Click the **Libraries** tab.
   d. Click **Add External JARs**. Select the JAR file, and click **Open**.
   e. Click **OK**.
2. Add the external JAR file to the server instance's class path
   a. Open the Server Configuration view, and expand the **Server** folder.
   b. Select the server instance that the project is deployed on. Right-click it and click **Open**.
   c. Click the **Paths** tab.
   d. Within ws.ext.dirs, click **Add External JARs**. Select the JAR file, and click **Open**. Note that ws.ext.dirs is used for application JAR files, and the CLASSPATH is used for server JAR files.
   e. Close the server instance and save your changes.

Optimizing multi-project builds using Dependent Project JARs

WebSphere Application Server - Express's powerful autobuild feature can slow down build performance during complex, multiproject builds. There are a number of ways to control these autobuilds (dependent files, active and inactive projects, and projects in source or JAR format) but these options can get quite complicated. There is an article that explains the options and how to optimize your build performance. See the WebSphere Developer Domain article "Optimizing Multi-Project Builds Using dependent Project JARs in WebSphere Studio Application Developer" (www.ibm.com/websphere/developer/library/techarticles/0204_searle/searle.html).

Automated production builds using Ant

You can use Ant with WebSphere Application Server - Express to automate your production builds. There is a multi-part article that explains the following things:

- What is Ant
- How to run Ant both inside and outside WebSphere Application Server - Express
- How to use Ant for production builds of J2EE elements (EJBs, WARs, EARs, and so on)
- How to extend Ant with new WebSphere Application Server - Express build tasks

See the WebSphere Developer Domain article ″Using Ant with WebSphere Studio Application Developer″ (www.ibm.com/websphere/developer/library/techarticles/0203_searle/searle1.html).

Chapter 10. Migration examples

This chapter contains migration examples to help you learn more about migrating from VisualAge for Java and WebSphere Studio ″Classic″ to WebSphere Application Server - Express IBM WebSphere Studio Site Developer.

- VisualAge for Java JSP/servlet sample (LeapYear)
- WebSphere Studio ″Classic″ Web application sample (YourCo)

Example: VisualAge for Java JSP/servlet sample (LeapYear)

Description

This is the FindTheLeapYears sample provided with VisualAge for Java Version 4.0. Information about it can be found in the VisualAge for Java online help (**Samples > JSP/Servlet Development Environment**).

Migration overview

You will follow the steps below to migrate the sample from VisualAge for Java to IBM WebSphere Studio Site Developer. These steps are discussed in more detail below:

1. Export your Java and project resource files from VisualAge for Java.
2. Create a new IBM WebSphere Studio Site Developer Web project.
3. Import your Java and project resource files into the IBM WebSphere Studio Site Developer project.
4. Define any servlets and make any application restructuring changes.
5. Create a IBM WebSphere Studio Site Developer server project.
6. Test the migrated application.

Exporting your files from VisualAge for Java

1. Open VisualAge for Java.
2. Select the **IBM JSP Examples** project.
3. Right-click the project and select **Export**. Select the **Directory** radio button and click **Next**.
4. Type the name of the directory you want to export the files to.
5. Clear the **.class** check box. You do not need to export these files as you will rebuild the project in WebSphere Application Server - Express and re-create these files.
6. Select the **.java** check box and click **Details**. Select only the LeapYear files and click **OK**.
7. Select the **resource** check box and click **Details**.
8. Select **LeapYearInput.html** and **LeapYearResults.jsp**, which are located in the following directory: **IBM WebSphere Test Environment\hosts\default_host\default_app\web\JSP\sample3.**
9. Click **OK**.
10. Clear the **Create Manifest file** check box (you do not need to create a manifest file).
11. Click **Finish**.

12. Close VisualAge for Java.

Creating a new IBM WebSphere Studio Site Developer Web project

1. Start IBM WebSphere Studio Site Developer.
2. Create a new Web project (**File > New > Project > Web > WebProject**) called `LeapYear`.
3. Ensure that **Dynamic Web project** is selected, and then click **Next**.
4. Select **New**.
5. Change the **Enterprise Application project name** to `LeapYearEAR` and select **J2EE level 1.2**. You could put the Web project into any existing Enterprise Application (EAR) project, but for this example, you will put it in LeapYearEAR.
6. Leave `LeapYear` in the **Context root** field.
7. Click **Finish**.

Importing the Java and project resource files into the IBM WebSphere Studio Site Developer project

Import the Java source files into the LeapYear project source directory by following these steps:

1. In the Web perspective, expand LeapYear and select the **JavaSource** directory.
2. Click **File > Import > File system** and click **Next**. Browse to the directory you exported your files to and click **OK**.
3. You only want to import the Java source files into the **JavaSource** directory, so, in the Import dialog, expand your export directory and select only the **com** subdirectory (it contains the three Java source files).
4. Click **Finish**. This creates LeapYear\JavaSource\com\ibm\ivj\wte\samples\leapyear\LeapYearXXXX.java files. Java classes are automatically compiled into LeapYear\WebContent\WEB-INF\classes.

Import the resource files into the LeapYear project under the **WebContent** directory by following these steps:

1. In the current Web perspective, expand the LeapYear project and select the **WebContent** directory.
2. Select **File > Import > File system** and click **Next**. Browse to the directory you exported your files to, expand your export directory to the **sample3** subdirectory, then click **OK**.
3. You only want to import the resource files into the **WebContent** directory, so, in the Import dialog, select the **sample3** subdirectory, which contains the .jsp and .html files.
4. Click **Finish**. The files are imported into the **WebContent** directory.

Defining any servlets and make any restructured application changes

1. You now need to create a servlet. Select the LeapYear project and expand it (**Leap Year > WebContent > WEB-INF**) to the web.xml file. Open the web.xml file.
2. Click the **Servlets** tab at the bottom of the page.

3. Click **Add**.
4. Ensure that the **Servlet** radio button is selected.
5. Select the class **LeapYear**, then click **OK**.
6. Select **URL Mapping > Add**, then type `LeapYear`.
7. Save the changes (**File > Save web.xml**) and close the web.xml file.

You now need to make any application changes due to the slightly changed source/application structure:

1. There will be two errors listed in the Tasks view. One is in LeapYearInput.html and one is in LeapYearResults.jsp.
2. Open the LeapYearResults.jsp file. Replace /JSP/index.html with `LeapYearInput.html`.
3. Open the LeapYearInput.html file. Replace /servlet/com.ibm.ivj.wte.samples.leapyear.LeapYear with `LeapYear`.
4. Save your changes and close the LeapYearResults.jsp and LeapYearInput.html files.
5. To avoid a run-time error, open the LeapYear.java file, which is located in the JavaSource\com\ibm\ivj\wte\samples\leapyear subdirectory:
6. Go to line 118 and change getRequestDispatcher from "/JSP/Sample3/LeapYearResults.jsp" to `"LeapYearResults.jsp"`
7. Save your changes and close LeapYear.java.

At this point the sample has been migrated into IBM WebSphere Studio Site Developer. All that remains is to create a IBM WebSphere Studio Site Developer server project and test the sample in the WebSphere test environment.

Creating a IBM WebSphere Studio Site Developer server project

1. Click **File > New > Project > Server > Server Project**. Click **Next**. In the **Project name** field, type `newServer` and click **Finish**. You will automatically be switched to the Server perspective.
2. Right-click **newServer** and click **New > Server and Server Configuration**.
3. In the **Server name** field, type `WSTestEnv`. In the **Server instance type** field, select **WebSphere V4.0 Test Environment**. Click **Finish**.

Now, you need to specify your EAR project to the server configuration:

1. In the Server Configuration view, expand the server items and click **WSTestEnv**.
2. Right-click it and click **Add > LeapYearEAR**.

Testing the migrated LeapYear application

1. Select the LeapYearInput.html file.
2. Right-click the HTML file, and from its pop-up menu, click **Run on Server**.
3. Wait while the server starts. Watch the **Console** page (click the **Console** tab in the Servers view) until the message "Server Default Server open for e-business" appears.
4. When a browser opens, type `2001` in the **Start Year** field, then click **Submit**.
5. The Console view shows the message `LeapYear:init`. Wait until you see the list of leap years, then select **WSTestEnv** in the Servers view. Right-click it and click **Stop**.

Example: WebSphere Studio ″Classic″ Version 4.0 Web Application (YourCo)(Windows)

Description

You must work with WebSphere Studio ″Classic″ Version 4.0.x for this example.

The sample you are going to work with is the **YourCo** sample. To access this sample open the online help (**Help > WebSphere Studio 4.0 > How to > Work with the samples > Overview**). To load this sample, follow the instructions in **Opening the Studio Samples** (for WebSphere Application Server 4.0) and load YourCo.war.

Note: The migrated application will execute in IBM WebSphere Studio Site Developer, but IBM WebSphere Studio Site Developer does not currently provide all the web page design and development features of WebSphere Studio, Professional or Advanced Editions.

Before you begin

- Ensure that the YourCo sample application is loaded in WebSphere Studio ″Classic″.
- Stop any instances of the WebSphere Application Server (so it will not conflict with IBM WebSphere Studio Site Developer)

Migration steps

To migrate this sample from WebSphere Studio ″Classic″ to IBM WebSphere Studio Site Developer, you will follow the steps below. Each step is described in more detail below.

1. (Optional) Start WebSphere Studio ″Classic″ and create a new Migration stage.
2. Create a Web configuration descriptor file.
3. Create a migration WAR file (containing a WebPath).
4. Start IBM WebSphere Studio Site Developer and import the WAR file.
5. Create a IBM WebSphere Studio Site Developer server project.
6. Test the migrated application.

Starting WebSphere Studio ″Classic″ Version 4.0 and creating a new Migration stage

(Optional) Normally, you would create a new stage for a migration, but for the purposes of this example, you use the Test stage included with WebSphere Studio ″Classic″. Using the Test stage will save you from having to manually edit many servlet mappings in step 8.

For information on how to create a new stage for migration, refer to the ″Migrating from WebSphere Studio ″Classic″ to IBM WebSphere Studio Site Developer″.

Creating a Web configuration descriptor file

1. In the project file view, click **Project > Create Web Configuration Descriptor File**, and accept the default value `WEB-INF\localhost_web.xml`.
2. Select all required servlets (any files that are not named xxxxBean).
3. There are no Tag Library Descriptor (TLD) files for this sample.

4. Click **Create**.

Creating a migration file

1. While in the project file view, select server *localhost* and click **Properties > Publishing > WebApp Web Path** and type a web path (context root) "newStudioSample". (Setting a Web path will remain the recommended approach in the final IBM WebSphere Studio Site Developer product).
2. While in the project file view, select **Project > Create Migration file**.
3. Verify that **localhost** is the selected server.
4. Verify that **localhost_web.xml** is the selected Web configuration descriptor file.
5. Click **OK**.
6. The default JAR file name is `X:\Studio40\projects\YourCo\localhost.jar`, where `X` is your WebSphere Studio ″Classic″ installation directory.
7. Click **Save**.
8. Close WebSphere Studio ″Classic″.
9. Rename the localhost.jar file to localhost.war.

Starting IBM WebSphere Studio Site Developer and importing the WAR file

1. Start IBM WebSphere Studio Site Developer.
2. Click **File > Import > WAR file > Next**.

   Note: You *must* import the JAR file using the WAR file option, otherwise it will not work properly.
3. Type the path to localhost.war in the **WAR File** field or click **Browse** to search for it.
4. In the **Web Project** field, select **New**, and then type `newStudioSample`
5. In the **EAR project name** field, select **New**, and then type `newStudioSampleEAR`
6. Click **Finish**. IBM WebSphere Studio Site Developer will unpack localhost.war.
7. You will have many unresolved references or missing import files. They will appear in the Task view.

   ```
   a. com.ibm.db requires databeans.jar,
   b. com.ibm.webtools.runtime requires webtlsrn.jar,
   c. com.ibm.ejs.ns.jndi requires ns.jar,
   d. com.ibm.webshpere.advanced.cm.factory requires cm.jar,
   e. com.ibm.ejs.models.base.extensions.webappext.ServletExtensions requires
   ws-base-extensions.jar
   ```

   To fix this, you must modify the Java build path for the Web project.
   a. Right-click the project and click **Properties > Java Build Path.**
   b. Click the **Libraries** tab. Click **Add External JARs**.
   c. Import the following JAR files: **databeans.jar, webtlsrn.jar, ns.jar, cm.jar**, and **ws-base-extensions.jar** from this directory: MyInstall\runtimes\aes_v4\lib
   d. Twenty-four warnings will remain. You do not need to deal with them.
8. Right-click the **newStudioSample** project and click **Rebuild Project**.

At this point the sample has been migrated into IBM WebSphere Studio Site Developer. All that remains is to create a IBM WebSphere Studio Site Developer Server project and test the sample in the WebSphere Test Environment.

Creating a IBM WebSphere Studio Site Developer server project

1. Switch to the Server perspective.
2. Click **File > New > Project > Server > Server Project**. Click **Next**. In the **Project name** field, type `newServer` and click **Finish**.
3. In the Navigator view, right-click **newServer** and click **New > Server and Server Configuration**.
4. In the **Server name** field, type `WSTestEnv`. In the **Server instance type** field, select **WebSphere V4.0 > Test Environment**. Click **Finish**.

Now, you need to specify your EAR project to the server configuration:

1. In the Server Configuration view, click **Servers > WSTestEnv**.
2. Right-click it and click **Add > newStudioSampleEAR.**

Note: (Optional) Right-click newStudioSample project, select **Properties > Server Preference > Always run on the following server**, select **WSTestEnv**, then click **Apply > OK**. (This step is only necessary if you have other servers.)

Testing the migrated YourCo application

1. Select the **YourCoIntro.html** file, which is located in the following directory in your newStudioSample project: WebContent\StudioSamples
2. Right-click **YourCoIntro.html**, and from its pop-up menu, click **Run on Server**, and then select **WSTestEnv**.
3. Wait while the server starts. Watch the **Console** page (click the **Console** tab in the Servers view) until the message `Server Default Server open for e-business` appears.
4. If you have not already run this sample in WebSphere Studio ″Classic″, then you need to configure the database by clicking **Database Configuration.**
5. When a browser opens, scroll down and click **Run This Sample**.
6. Wait until the browser **Welcome** page appears, then click **Employee Center**.

   Note: The first time you run this application, you will receive the following errors in the Console page: `DataSource not found. Try to construct a new datasource name: jdbc/yourco DataSource not found. Try to construct a new datasource name: jdbc/studio`. These errors are self-correcting. You can ignore them.
7. When you are done, close the browser window and the Web Browser view, then in the **Server Control Panel** right-click **WSTestEnv** and click **Stop.**
8. (Optional) Close IBM WebSphere Studio Site Developer.

Chapter 11. Further reading

Up-to-date information on migration and other topics is available at www.ibm.com/websphere/developer/zones/studio/transition.html

The following publications and Web pages provide general information which you may find helpful when working with WebSphere Application Server - Express:

- *JSR-000053 Java Servlet 2.3 and JavaServer Pages 1.2 Specifications*: java.sun.com/aboutJava/communityprocess/first/jsr053/index.html
- The IBM WebSphere Application Server Version 4 InfoCenter: www.ibm.com/software/webservers/appserv/doc/v40/aes/infocenter/index.html
- *WebSphere Version 4 Application Development Handbook*: www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246134.pdf
- *Programming J2EE APIs with WebSphere Advanced*: www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246124.pdf
- *WebSphere Application Server Version 3.5 to 4.x - Migration Hints & Tips*: www7b.software.ibm.com/wsdd/library/techarticles/0208_wright/wright.html
- *WebSphere Studio Application Developer Service Portal*: www.ibm.com/software/ad/studioappdev/support/
- *WebSphere Studio Application Developer FAQ Frequently Asked Questions*: www.ibm.com/support/search.wss?rs=457&tc=SSBRLP&dc=D800
- *WebSphere Application Server Service Portal*: www.ibm.com/software/software/webservers/appserv/support.html
- *WebSphere Application Server FAQ Frequently Asked Questions*: www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&dc=D800

Further reading which may be of interest:

- An article on Using Ant with WebSphere Studio Application Developer (including J2EE project builds/exports): www.ibm.com/websphere/developer/library/techarticles/0203_searle/searle1.html
- An article on Optimizing complex builds in WebSphere Studio Application Developer: www.ibm.com/websphere/developer/library/techarticles/0204_searle/searle.html
- An article on J2EE Class Loading (J2EE modules and class paths) in WebSphere Studio Application Developer: www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer.html
- An article on developing J2EE utility JARs (Java JARs in J2EE modules) in WebSphere Studio Application Developer: www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer2.html
- An article on team support in WebSphere Studio Application Developer: www.ibm.com/websphere/developer/library/techarticles/ 0108_karasiuk/ 0108_karasiuk.html
- An article on Migrating Enterprise Access Builder Components from VisualAge for Java to WebSphere Studio Application Developer: www.ibm.com/websphere/developer/techjournal/0201_minocha/minocha.html
- An article on EJB application design using the Session Facade to talk to CMPs: www.ibm.com/websphere/developer/library/techarticles/ 0106_brown/sessionfacades.html

- An article on WebSphere Application Server Best Practices: www.ibm.com/software/webservers/appserv/ws_bestpractices.pdf
- An article on WebSphere Best Practices zone: www.ibm.com/websphere/developer/zones/bp/
- WebSphere Developer Domain main Web page: www.ibm.com/websphere/developer
- WebSphere Developer Domain Technical Articles: www.ibm.com/websphere/developer/techjournal/
- Information about the WebSphere Studio family, and the features and directions of WebSphere Studio Application Developer: www.ibm.com/websphere/developer/zones/studio/transition.html
- All about the IBM WebSphere Studio Family of Development Tools: www.ibm.com/websphere/developer/library/techarticles/0108_studio/studio_beta.html
- External Application Developer newsgroup: news://news.software.ibm.com/ibm.software.websphere.studio.application-site-developer
- External workBench (Eclipse) newsgroup: news://news.software.ibm.com/ibm.software.websphere.studio.workbench
- External WebSphere Application Server newsgroup: news://news.software.ibm.com/ibm.software.websphere.application-server
- An article on deploying a J2EE Application from WebSphere Studio Application Developer to WebSphere Application Server: www.software.ibm.com/vad.nsf/Data/Document3584
- Application Developer Software Configuration Management (Source Code Management) vendors: www.ibm.com/software/ad/studioappdev/partners/scm.html
- Migrating applications to Application Developer from competitors development tools: www.ibm.com/websphere/developer/zones/studio/migration.html
- Migrating VisualCafé WebLogic applications to Application Developer (still deploying to WebLogic): www.ibm.com/websphere/developer/library/techarticles/0209_searle/searle1.html
- Eclipse.org: www.eclipse.org
- WebSphere Developer Domain Plugin Central: www.ibm.com/websphere/developer/downloads/plugin/
- Eclipse workbench plug-ins (not part of Eclipse.org): www.eclipse-workbench.com/jsp/plugins.jsp
- Eclipse plug-ins (not part of Eclipse.org): www.eclipse-plugins.2y.net/eclipse/plugins.jsp

Notices

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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
North Castle Drive
Armonk, NY 10504-1785
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

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

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 OR CONDITIONS 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.

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:

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

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 document 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.

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.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may 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 illustrates 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 2000, 2003. All rights reserved.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, or other countries, or both:

- AIX
- CICS
- Cloudscape
- DB2
- DB2 Extenders
- DB2 Universal Database
- e-business
- IBM
- iSeries
- OS/390
- S/390
- VisualAge
- WebSphere
- z/OS

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

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

UNIX is a registered trademark of The Open Group.

Other company, product or service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.