WebSphere software

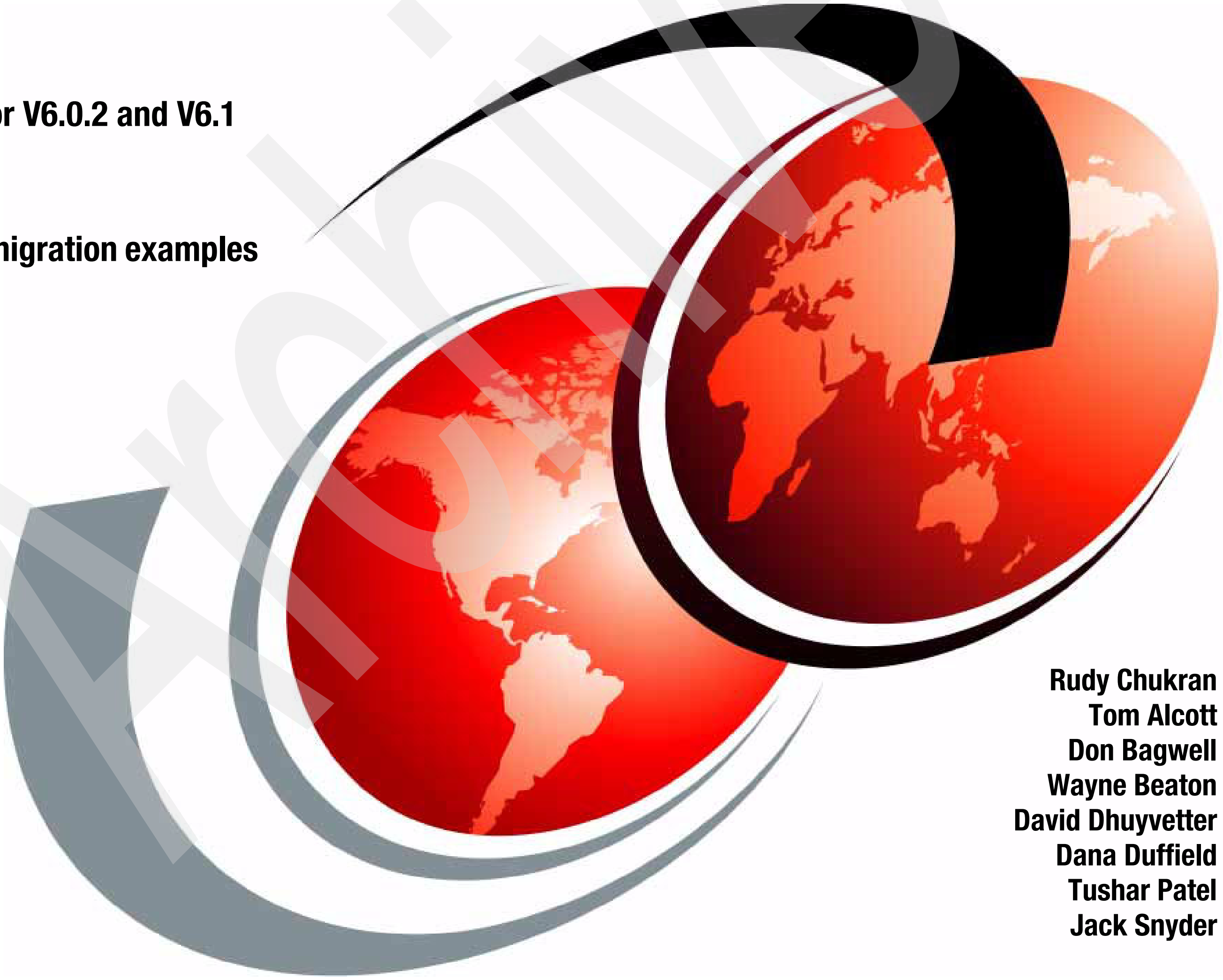
IBM

WebSphere Application Server V6 Migration Guide

Updated for the z/OS platform

Updated for V6.0.2 and V6.1

Practical migration examples



Rudy Chukran
Tom Alcott
Don Bagwell
Wayne Beaton
David Dhuyvetter
Dana Duffield
Tushar Patel
Jack Snyder

Redbooks

ibm.com/redbooks

International Technical Support Organization

WebSphere Application Server V6 Migration Guide

August 2006

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

Second Edition (August 2006)

This edition applies to Version 6 of IBM WebSphere Application Server.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which 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.

Trademarks

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

@server®
@server®
Redbooks (logo) ™
developerWorks®
iSeries™
pSeries®
z/OS®
zSeries®
AIX 5L™
AIX®
ClearCase®
Cloudscape™
Domino®
DB2 Connect™
DB2 Universal Database™
DB2®
Informix®
IBM®
Lotus®
MQSeries®
MVS™
OS/400®
PowerPC®
Rational®
Redbooks™
RACF®
Tivoli®
TXSeries®
VisualAge®
WebSphere®

The following terms are trademarks of other companies:

EJB, Java, Javadoc, JavaServer, JDBC, JDK, JMX, JSP, JVM, J2EE, J2SE, Solaris, Sun, Sun Java, Visual Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Intel, Itanium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

"The creation of this Migration Guide is the result of tremendous input from customers requesting guidance on migrating their WebSphere® development environment, including applications, integration, and test and production environments. This guide has been compiled using the collective experience of many experts who have supported thousands of installations and migrations."

Betsy Matthew
Vice President, AIM Technical Support and Customer Service

What is in this redbook?

This IBM® Redbook will assist in the migration of your WebSphere Application Server installation. The end-to-end migration path includes the migration of the development environment, the test/integration environment, and the production environment using software engineering methodologies. This guide presents best practices in migration strategy and planning, migration tools, and practical migration examples.

This guide was written specifically to address this set of products:

- Version 6 IBM Rational Application Developer for WebSphere Software and IBM Rational Web Developer for WebSphere Software for the Windows® and Intel® Linux® 32 bit operating system platforms
- Version 6 of the WebSphere Application Server, WebSphere Application Server Express, and WebSphere Application Server Network Deployment products. These products specifically target these operating system platforms, which we subsequently refer to collectively as the distributed platforms:
  - Microsoft® Windows for Intel 32 bit architecture
  - Microsoft Windows for Intel EM64T 64 bit and AMD 64 bit architecture
  - Linux for Intel 32 bit architecture
  - Linux for Intel EM64T 64 bit and AMD 64 bit architecture
  - Linux for IBM zSeries® architecture
  - IBM AIX® for PowerPC® architecture
  - Sun™ Solaris™ for Sparc architecture
  - Hewlet Packard HP-UX PA-Risc architecture
  - Hewlet Packard HP-UX for Itanium® 2 64 bit architecture
- WebSphere Application Server for z/OS® Version 6 for IBM zSeries architecture

This redbook includes the following sections:

- Part 1, "For application developers" on page 1

  This part is of most interest to developers who create and modify applications. Developers who develop for any of the WebSphere platforms can find pertinent material here.

  - Chapter 1, "Development tools overview" on page 3

    This chapter describes the new Rational Developer products and the key differences with the previous versions of the WebSphere Studio product line. Key user interface differences from past versions are discussed.

  - Chapter 2, “Application migration to developer tools” on page 13

    This chapter discusses workspace migration, project migration, and limitations of compatibility between versions of the development tools.

  - Chapter 3, “Changing application code” on page 17

    This chapter describes application coding issues for both Java™ applications and wsadmin scripts.

  - Chapter 4, “Development migration scenarios” on page 39

    This chapter describes specific steps showing how to migrate application projects from previous versions of WebSphere Studio to the current Rational® Developer product line.

- Part 2, “For system administrators” on page 73

  This part is of most interest to administrators whose must manage configurations of application servers and deploy applications to those servers.

  - Chapter 6, “Product overview” on page 75

    This chapter reviews the new content of the WebSphere product line that is important to migrating existing applications.

  - Chapter 7, “Runtime migration strategy” on page 85

    This chapter discusses global migration strategies and the planning that must occur before an application migration is executed.

  - Chapter 8, “Distributed runtime administration overview” on page 97

    This chapter discusses the major differences in how system administration is done on the distributed platforms.

  - Chapter 9, “z/OS runtime administration overview” on page 139

    This chapter discusses the major differences in how system administration is done on the z/OS operating platform.

  - Chapter 10, “Migration tasks” on page 151

    This chapter discusses the high-level tasks that an administrator must be aware of for various system configurations.

  - Chapter 11, “Script compatibility” on page 167

    This chapter discusses the compatibility issues with administrative scripts and the resources you need to rework scripts that are no longer compatible.

  - Chapter 12, “Distributed runtime application migration examples” on page 185

    This chapter gives you concrete examples of how a simple application is migrated in various server configurations on the distributed platforms.

  - Chapter 13, “z/OS runtime migration examples” on page 241

    This chapter gives you concrete examples of how a simple configuration is migrated in various server configurations on the z/OS platform.

  - Chapter 14, “V6.1 updates for distributed platforms” on page 303

    This chapter presents the new material that applies only to WebSphere V6.1 on any of the distributed platforms.

  - Chapter 15, “V6.1 updates for z/OS platforms” on page 325

    This chapter presents the new material that applies only to WebSphere V6.1 on the z/OS platform.

How to use this book

This book is an update of the migration guide that was published concurrently with the V6.0 release of WebSphere. As a whole, it applies to all releases of Version 6, which are V6.0 and V6.1 at the time of publication.

If you are migrating to WebSphere 6.0, chapters 1 through 13 are relevant, but you can skip chapters 14 and 15.

The entire book is pertinent if you are migrating to WebSphere V6.1. However, some of the details included in the first 13 chapters do not apply to V6.1. Chapters 14 and 15 provide information specific to this latest release. Where there are discrepancies between the first 13 chapters and the last two, the information in chapters 14 and 15 applies to WebSphere V6.1, and supersedes the material in previous chapters.

The team that wrote this redbook

The following people were members of the team that authored the book. They all work for IBM in the United States in various roles involving the WebSphere product line.

Rudy Chukran acted as project leader for this redbook. He is an Advisory Software Engineer with over 30 years of experience in the computer industry. His most recent role has been supporting customers for WebSphere early access programs. Prior to that, he worked as a software developer on several products, earned two IBM Patent awards, wrote several articles for technical publications, and authored the book *Accelerating AIX*. He holds a Bachelor of Science degree in Electrical Engineering from Carnegie-Mellon University.

Tom Alcott is a consulting I/T specialist in the United States. He has been a member of the World Wide WebSphere Technical Sales Support team since its inception in 1998. In this role, he spends most of his time trying to stay one page ahead of customers in the manual. Before working with WebSphere, he was a systems engineer for IBM's Transarc Lab, supporting TXSeries®. His background includes over 20 years of application design and development on both mainframe-based and distributed systems. He has written and presented extensively on a number of WebSphere runtime issues. Among his publications are *Migrating to WebSphere V5.0: An End-to-End Migration Guide*, *WebSphere Scalability: WLM and Clustering*, *WebSphere V3 Performance Tuning Guide* and, most recently, *IBM WebSphere: Deployment and Advanced Configuration*.

Don Bagwell has been with IBM for 22 years, working in such diverse areas as manufacturing logistics, software test, systems engineering, sales, and now WebSphere Application Server for z/OS support. He focuses on writing white papers and developing educational offerings. He reports to the Washington Systems Center in Gaithersburg, Maryland, but he lives in Tucson, Arizona.

Wayne Beaton is a Senior Software Consultant with Software Services for WebSphere, part of the IBM Software Group. His main focus is helping customers to migrate from previous versions and competitive products (including Visual Basic®) to WebSphere Application Server 5.x. He has co-authored two other IBM Redbooks™ on the topic of migration. Wayne's diverse role involves him in many other interesting areas including the WebSphere Skills Transfer programs and general consulting.

David Dhuyveter is a Senior Software Engineer with the WebSphere Enablement Team. With IBM since 2000, he has over 10 years of experience in distributed transactional systems. David has focused on application development and migration, and is co-author of the redbook *Migrating to WebSphere V5.0: An End-to-End Migration Guide*. David received his MS in Computer Science from California State Polytechnic University, Pomona in 1993.

Dana Duffield is a Senior Software Engineer and is the Migration and Interoperability Architect for WebSphere Application Server in IBM Rochester. He is a software engineer who has worked at the IBM Rochester Lab for over 22 years on various development projects. Dana holds a Bachelor of Science degree in Computer Science from the University of Illinois, Champagne-Urbana. He has significant experience in object-oriented and distributed system development in architecture, development, and leadership positions. Prior to working at IBM, he worked at Bell Laboratories in Naperville, Illinois.

Tushar Patel is a WebSphere Lab Advocate in the United States, a software advisory role to help support WebSphere customers in large complex deployments. He has over 12 years of technical experience in Information Technology in various capacities. He holds a BS degree in Software Engineering from University of Birmingham, England and an MBA from the University of Cambridge, England.

Jack Snyder is a Senior I/T Architect in IBM Pittsburgh. He has over 30 years of experience in computer systems and the engineering field of mining and computer systems. Jack holds B.S., M.S., and Ph.D. degrees in engineering. His areas of expertise include WebSphere Application Server, Java, and process control.

The following people assisted in providing information and review commentary for the book:

- Vijay Bhadriraju is an Advisory Software Engineer with over six years of experience in IBM. He is currently a developer of J2EE™ tools and the team lead for migration topics for the WebSphere Studio and Rational Developer line of products. Vijay contributealso reviewed this book
- Diane Chalmers is a Staff Software Engineer located in Rochester, Minnesota. She is the Migration focal point for WebSphere Application Server System Verification Test team. Diane has over five years of test experience.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

Part 1

For application developers

The chapters in this part are of interest to application developers who need to know about development environment and application programming issues.

1

Development tools overview

This chapter is an overview of IBM Rational Developer products that support J2EE development for the IBM WebSphere Application Server. It covers the minimum set of new functions that a developer must be aware of in order to migrate from previous versions to Version 6.

1.1 Overview of WebSphere development tools

The suite of IBM products related to developing J2EE applications on the market during the years 2001 through 2004 was marketed under the brand name *WebSphere Studio*. For V6, the brand name changes to *Rational Developer*. This suite of products is based on the Eclipse open source project. The foundation uses the Eclipse workbench as a framework and adds IBM unique tools and functions. Table 1-1 is a summary of the products covered in this book.

Table 1-1 WebSphere development tools products

Version 4	Version 5	Version 6	Targeted to
WebSphere Studio Application Developer	WebSphere Studio Application Developer	IBM Rational Application Developer for WebSphere Software	Full J2EE 1.4 development
WebSphere Studio Site Developer	WebSphere Studio Site Developer	IBM Rational Web Developer for WebSphere Software	J2EE Web module development subset

The development tools have evolved from Version 4 to Version 6; enhanced functions enable each product to do more than the previous version.

1.1.1 IBM Rational Web Developer for WebSphere Software

IBM Rational Web Developer for WebSphere Software has evolved from its predecessor products of WebSphere Studio Site Developer V4 and WebSphere Studio Site Developer V5. It is primarily aimed at Web developers and supports J2EE 1.4 standards with the exception that no EJB™ development tools are provided. The following significant features are contained in V6:

- WebSphere Application Server test environments
- Visual Java™ GUI builder
- Web Diagram Editor
- Site designer
- Page templates
- XML tools
- Web services tools
- Database tools
- Portal and portlet development
- Enterprise Generation Language development

1.1.2 IBM Rational Application Developer for WebSphere Software

IBM Rational Application Developer for WebSphere Software has evolved from its predecessor products of WebSphere Studio Application Developer Version 4 and WebSphere Studio Application Developer Version 5. IBM Rational Application Developer for WebSphere Software has broadened its appeal by adding more functions. IBM Rational Application Developer for WebSphere Software has all the functions of IBM Rational Web Developer for WebSphere Software with these additional features:

- J2EE EJB development
- UML visual editors
- Static and run time analysis
- Extended debugging and profiling

- Component test automation
- ClearCase LT for team integration
- Rational Unified Processing integration

1.1.3 WebSphere Application Server Toolkit

WebSphere Application Server Toolkit is a limited purpose tool that is used to assemble and repackage existing applications. It is not a full function development environment. WebSphere Application Server Toolkit has a subset of the functionality of IBM Rational Application Developer for WebSphere Software. This means that the user interfaces controlling those subset functions are identical to those in IBM Rational Application Developer for WebSphere Software. The following functions are included:

- Assembly operations
  - Import wizards to import jar and class files
  - Module validators
- Deployment operations
  - J2EE deployment descriptor editors
  - Editors for IBM deployment extensions and binding files
  - EJBdeploy invocation for generating EJB deployment artifacts
- Debug operations
  - Remote and local testing on WebSphere Application Server Version 6 servers
  - Profiling on WebSphere Application Server Version 6 servers

See the InfoCenter documentation for an overview of WebSphere Application Server Toolkit:

http://publib.boulder.ibm.com/infocenter/ws60help/topic/com.ibm.welcome.ast.doc/topics/astoverview.html

1.1.4 Supported operating systems

See Appendix A, “Prerequisite software” on page 335 for a complete list of supported operating systems.

1.1.5 Embedded WebSphere Application Server test environment

IBM Rational Application Developer for WebSphere Software and IBM Rational Web Developer for WebSphere Software deliver optionally installed embedded WebSphere test server environments. The following servers are delivered with Rational Developer products. Please note that WebSphere Application Server V4 is no longer offered as an embedded test environment.

- WebSphere Application Server V6.0
- WebSphere Application Server V5.1
- WebSphere Application Server V5.0
- WebSphere Application Server Express V5.1
- WebSphere Application Server Express V5.0

1.2 Important user interface changes

You should be aware of some of the user interface changes that have been made. Otherwise, you may have difficulty finding the functions you are already familiar with from previous versions.

New welcome page

When you first launch IBM Rational Application Developer for WebSphere Software, the new welcome page shown in Figure 1-1 is displayed.

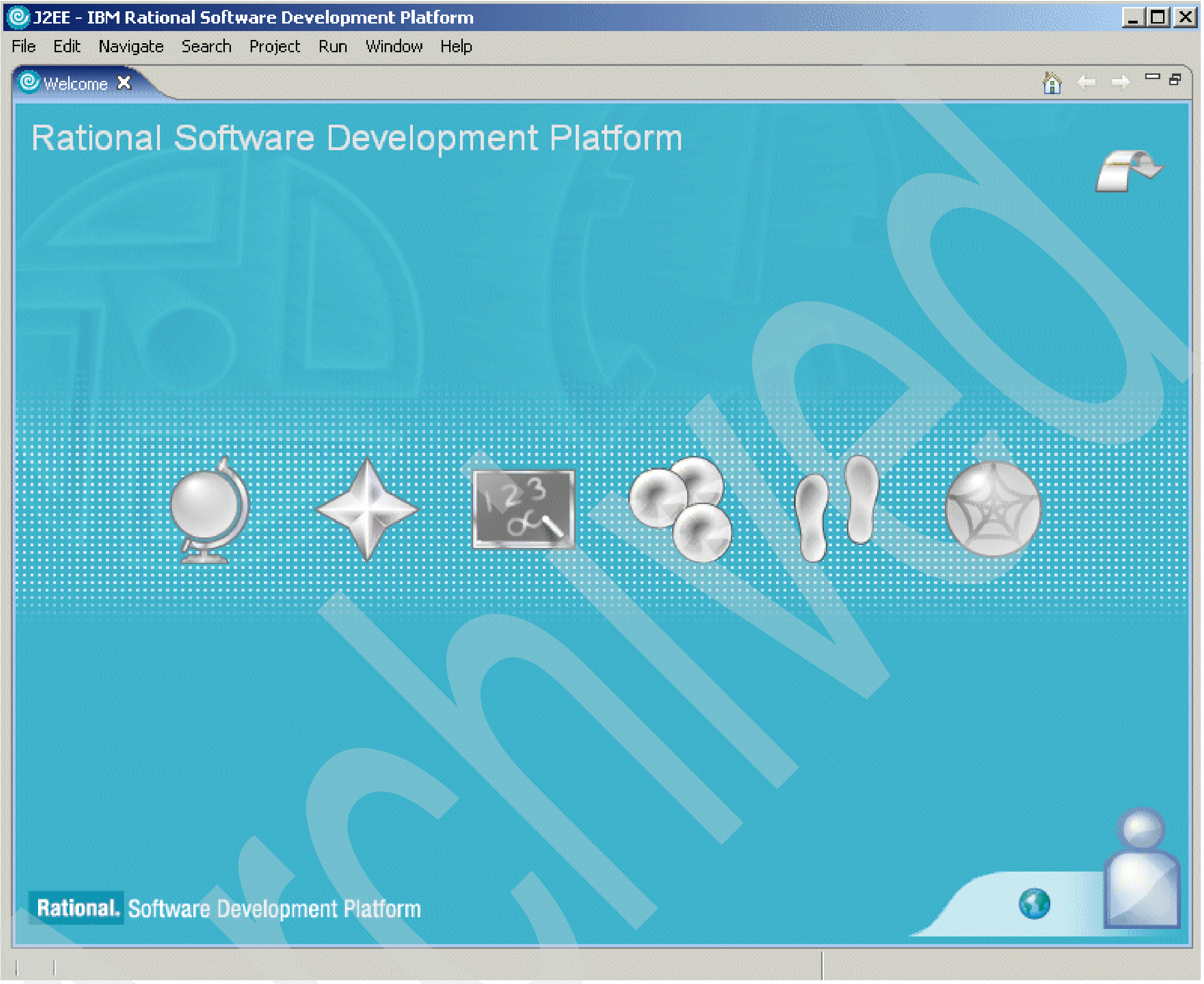

J2EE - IBM Rational Software Development Platform
File Edit Navigate Search Project Run Window Help
Welcome
Rational Software Development Platform
Rational. Software Development Platform

Figure 1-1 Rational Developer initial window

You can select any of the icons in the welcome page, and you can see documentation that gives you more information about the new features. If you are eager to get to work with IBM Rational Application Developer for WebSphere Software, then simply click the **x** symbol in the page tab next to the word *Welcome* to dismiss the page and see the product workbench desktop.

Perspective switcher location

The icon to open new perspectives and switch between open perspectives has changed its default location to the upper right corner of the workbench. Figure 1-2 on page 7 shows where the perspective switcher is found.

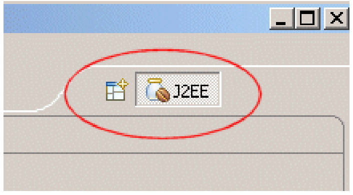
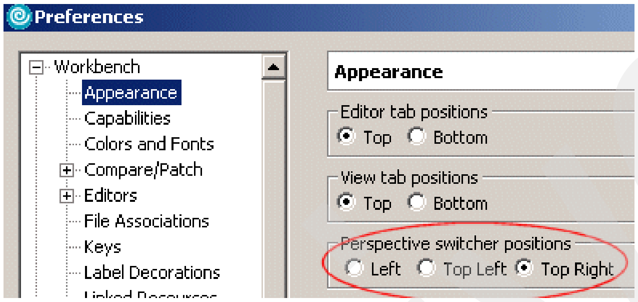


Figure 1-2 Perspective switcher

If you prefer the perspective switcher location of the previous versions, you can change the location making a change in the workbench appearance preference settings. Figure 1-3 shows how to change the position preference for the perspective switcher.



Appearance

Editor tab positions
● Top ○ Bottom

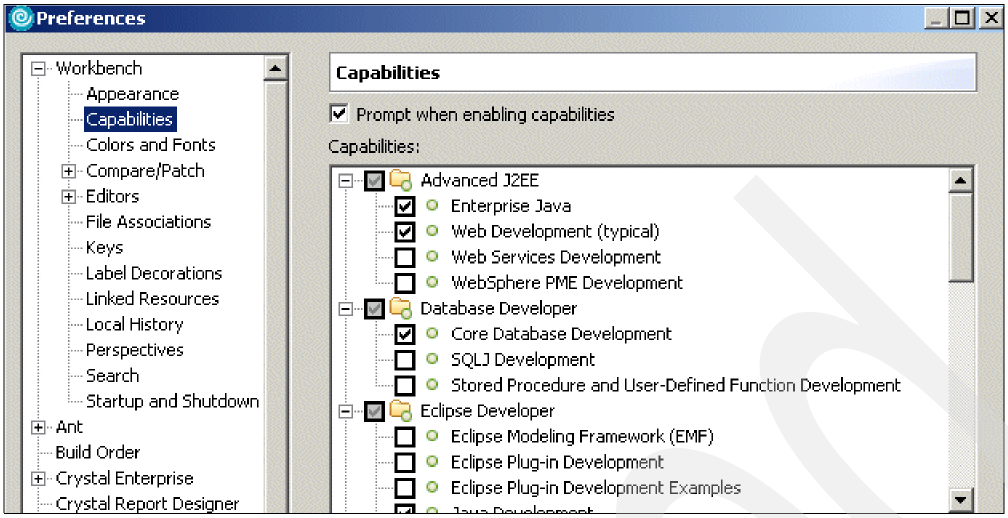
View tab positions
● Top ○ Bottom

Perspective switcher positions
○ Left ○ Top Left ● Top Right

Figure 1-3 Rational Developer preferences for perspective switcher

Enabling and disabling workbench capabilities

When you first launch IBM Rational Application Developer for WebSphere Software, you may not see all the capabilities available that you are expecting to see. V6 has adopted the concept of easy feature disabling and enabling. The default is to enable J2EE development, but to disable many of the other development features you are used to seeing. Disabling unneeded features can improve IBM Rational Application Developer for WebSphere Software startup time. Figure 1-4 on page 8 shows an example of the window that configures enabling features. Select **Window** → **Preferences**, and expand the workbench preferences to see what you can enable and disable. Before any customization, J2EE Enterprise Java and J2EE Web Development are enabled by default, but Web services development is disabled. Customize the capabilities preferences to enable those capabilities that you frequently use, and disable those capabilities that you seldom use.



Preferences

- Workbench
  - Appearance
  - Capabilities
  - Colors and Fonts
  - Compare/Patch
  - Editors
  - File Associations
  - Keys
  - Label Decorations
  - Linked Resources
  - Local History
  - Perspectives
  - Search
  - Startup and Shutdown
- Ant
- Build Order
- Crystal Enterprise
- Crystal Report Designer

Capabilities

☑ Prompt when enabling capabilities

Capabilities:

- ☑ Advanced J2EE
  - ☑ Enterprise Java
  - ☑ Web Development (typical)
  - □ Web Services Development
  - □ WebSphere PME Development
- ☑ Database Developer
  - ☑ Core Database Development
  - □ SQLJ Development
  - □ Stored Procedure and User-Defined Function Development
- ☑ Eclipse Developer
  - □ Eclipse Modeling Framework (EMF)
  - □ Eclipse Plug-in Development
  - □ Eclipse Plug-in Development Examples
  - ☑ Java Development

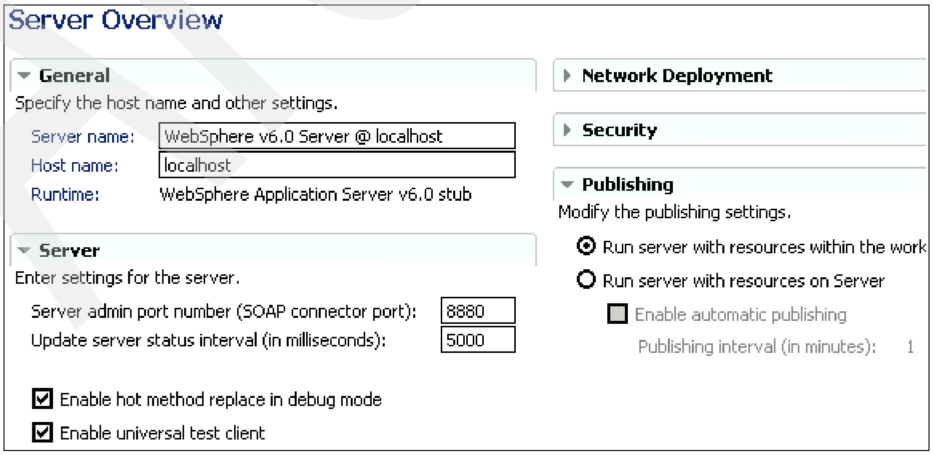
Figure 1-4 Rational Developer capabilities

Enhanced WebSphere Application Server test environment

The most significant change to the user interface that affects how you use the product is in the area of the WebSphere Application Server test environment.

Version 4 and Version 5 of WebSphere Studio include the concept of a server configuration editor. This editor changes the server settings and those settings are saved into the server configuration project. The V5 server configuration editor supersedes almost all the server settings that are changeable in the application server administrative console. However, there are certain settings that could only be changed from the administrative console. This means that inconsistencies with the user interface arose, and these inconsistencies could be confusing because some settings were changed with the configuration editor, while some settings were changed using the administrative console.

For V6, IBM Rational Application Developer for WebSphere Software relies on the administrative console for the great majority of its configuration. A server configuration as part of the server project still exists, but in greatly reduced form. When you open the Server configuration, you see a basic overview, which contains basic server launch information only. See an example of the server overview in Figure 1-5.



Server Overview

General

Specify the host name and other settings.

Server name: WebSphere v6.0 Server @ localhost

Host name: localhost

Runtime: WebSphere Application Server v6.0 stub

Server

Enter settings for the server.

Server admin port number (SOAP connector port): 8880

Update server status interval (in milliseconds): 5000

☑ Enable hot method replace in debug mode

☑ Enable universal test client

Network Deployment

Security

Publishing

Modify the publishing settings.

● Run server with resources within the work

○ Run server with resources on Server

□ Enable automatic publishing

Publishing interval (in minutes): 1

Figure 1-5 Rational Developer Server overview

You launch the administrative console in exactly the same way as in previous versions: Right-click the server and select **Run Administrative Console** as shown in Figure 1-6. This operation launches a browser window in which the administrative console for WebSphere Application Server is running.

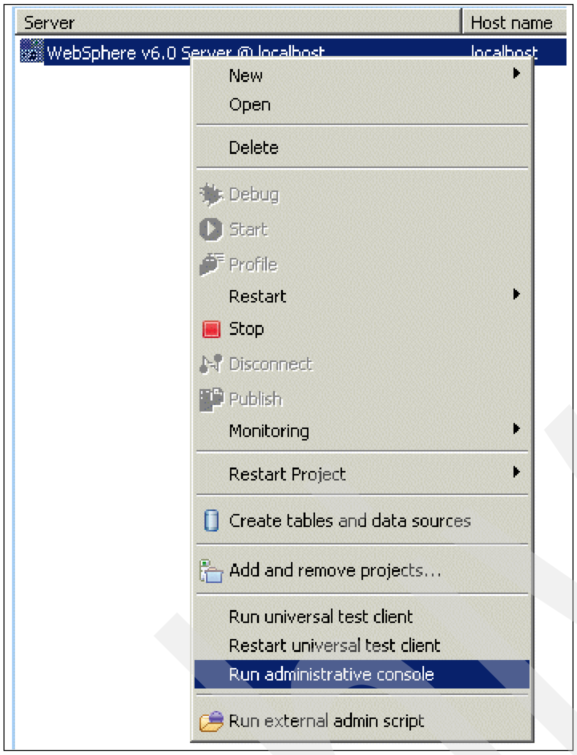

Server
Host name
WebSphere v6.0 Server @ localhost
localhost
New
Open
Delete
Debug
Start
Profile
Restart
Stop
Disconnect
Publish
Monitoring
Restart Project
Create tables and data sources
Add and remove projects...
Run universal test client
Restart universal test client
Run administrative console
Run external admin script

Figure 1-6 Rational Developer: run administrative console

In previous versions, you are required to install and configure a Remote Agent Controller (RAC) in order to test on external application servers, even if the application server is installed on the same system. V6 simplifies testing with locally installed servers by including the RAC functionality such that you can test on a local server without installing a RAC.

1.3 How to obtain more information

Following are some sources of further help and information.

Built-in help

Both IBM Rational Developer products have a built-in help facility. You access the help via the Help menu as shown in Figure 1-7 on page 10.

The Help browser looks like that shown in Figure 1-8 on page 10. You can perform keyword searches by typing in the search area. You can also browse articles in the navigation box on the left.

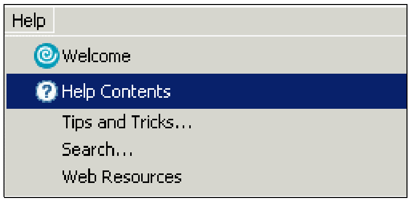
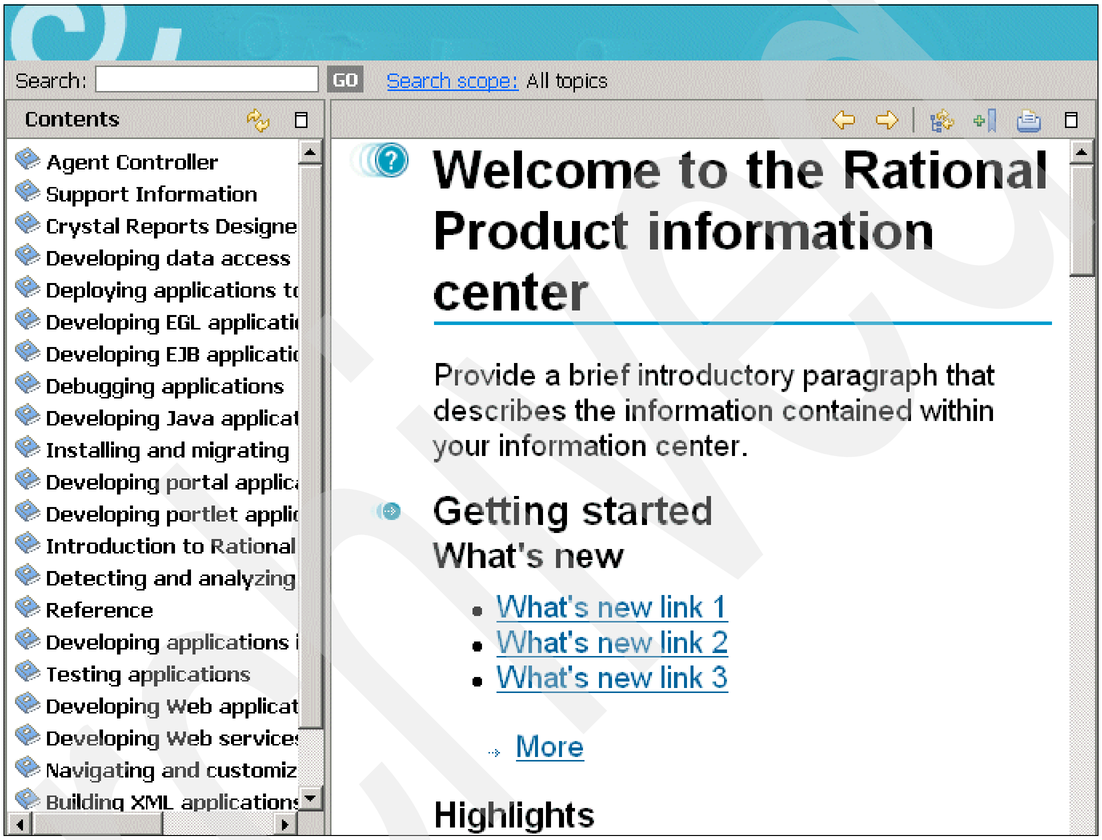


Figure 1-7 Rational Developer Help



Welcome to the Rational Product information center

Provide a brief introductory paragraph that describes the information contained within your information center.

Getting started

What's new

- What's new link 1
- What's new link 2
- What's new link 3

More

Highlights

Figure 1-8 Rational Developer Help browser

Web Resources

You can access help on the World Wide Web by locating the Web Resources page and selecting links that interest you. Figure 1-9 shows the process of accessing the Web Resources page via the Help menu. Selecting one of the icons on the resulting page (Figure 1-10 on page 11) opens a browser window to the corresponding information page on the Web.

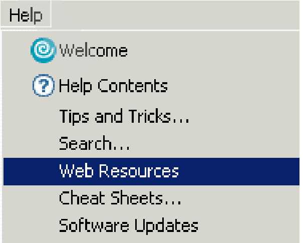
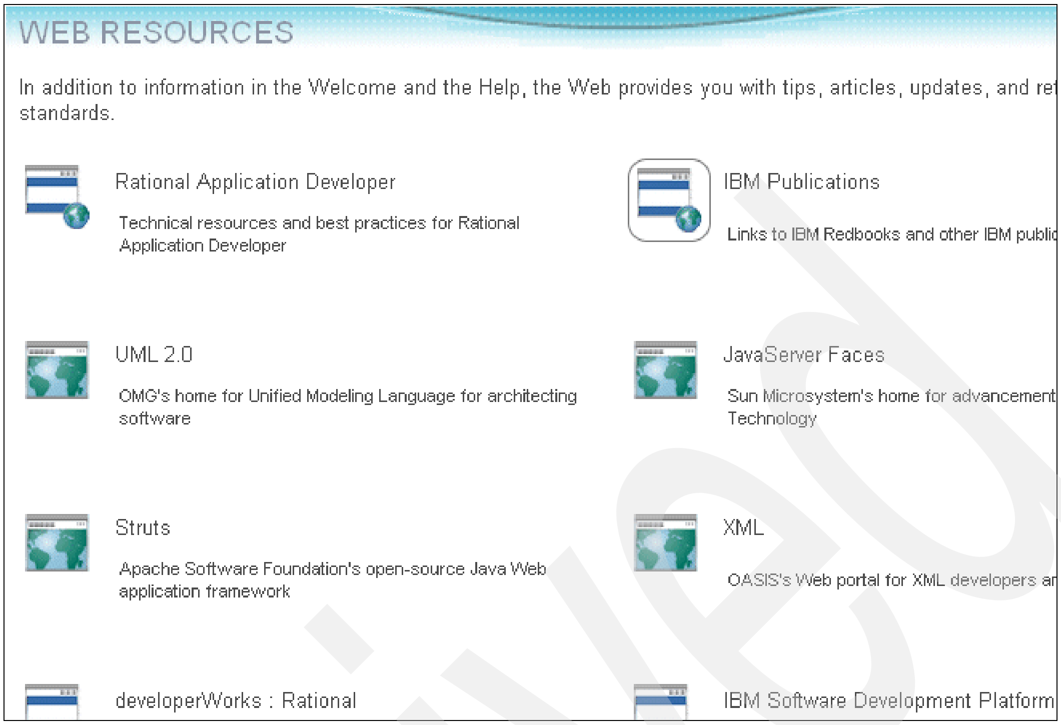


Figure 1-9 Rational Developer: starting Web Resources



WEB RESOURCES

In addition to information in the Welcome and the Help, the Web provides you with tips, articles, updates, and ref standards.

Rational Application Developer

Technical resources and best practices for Rational Application Developer

IBM Publications

Links to IBM Redbooks and other IBM public

UML 2.0

OMG's home for Unified Modeling Language for architecting software

JavaServer Faces

Sun Microsystem's home for advancement Technology

Struts

Apache Software Foundation's open-source Java Web application framework

XML

OASIS's Web portal for XML developers ar

developerWorks : Rational

IBM Software Development Platform

Figure 1-10 Rational Developer Web Resources links

2

Application migration to developer tools

This chapter discusses methods that developers can use to migrate their application source code into the latest development tools.

2.1 General application upgrade issues

The general issues that a developer must take into account when migrating an application to a new version of a development platform are the following:

- Adapting a general development process to a new product user interface
- Changing source code due to mandatory programming interface changes
- Moving source code from an archive into a development tool

User interface changes are discussed in 1.2, “Important user interface changes” on page 5 and Chapter 4, “Development migration scenarios” on page 39.

Source code changes are discussed in detail in Chapter 3, “Changing application code” on page 17.

Moving source code into IBM Rational Application Developer for WebSphere Software and IBM Rational Web Developer for WebSphere Software is discussed further in 4.1.1, “Example: Migrating Plants By WebSphere using the export/import method” on page 40.

2.2 Importing source code

Source code can be contained in these types of locations:

- A set of projects contained in a workspace from a previous version of WebSphere Studio
- A source code repository product
  - CVS
  - ClearCase LT
- An J2EE EAR file

2.2.1 Migrating workspaces from previous versions

A V5.1.x (any of 5.1.0, 5.1.1, or 5.1.2) workspace from WebSphere Studio Application Developer opened in IBM Rational Application Developer for WebSphere Software can be automatically migrated. You can open a Version 5.1.x workspace at initial startup time, or you can open a workspace by selecting **Switch Workspace** from the File menu. The 5.1.x workspace is converted in place transparently by an hidden conversion process. Upon conversion, the workspace can no longer be opened by WebSphere Studio Application Developer.

Server targeting

Server targeting is a feature introduced in WebSphere Studio Application Developer V5.1.1. Server targeting is an optional feature in the J2EE preferences in V5.1.1 and V5.1.2 and can be enabled or disabled by the user. In IBM Rational Application Developer for WebSphere Software, this feature is not optional and server targeting is therefore enabled by default.

For more details about the server targeting feature, refer to the online help.

Sharing projects with earlier versions

V5.1.x workspaces migrated to IBM Rational Application Developer for WebSphere Software cannot be shared with earlier versions of WebSphere Studio Application Developer. That is, no version, 5.0 or earlier, is able to understand the workspace contents and therefore none can open the workspace.

However, V5.1.x projects that are imported from a source code repository or another developer into V6.0 are compatible for sharing with V5.1.x so long as no new features in V6.0 are used in the project. Sharing projects between V5 and V6 requires that you export the project with the Project Interchange option. The output of the export operation is a ZIP file that contains the project and the project metadata. You then import the same ZIP file into WebSphere Studio Application Developer using the Project Interchange option. For more information, see the online help facility and look for the topic *Sharing Projects using Project Interchange*. For information about how to launch the online help facility, see 1.3, “How to obtain more information” on page 9.

Known problem when migrating a workspace

You may see error messages displayed and logged when a V5.1.x workspace is opened in Rational Application Developer V6.0, stating that the V6.0 workbench is unable to restore the perspectives and views that existed in the V5.1.x workspace. This is a known problem with workspace migration.

An example of the error messages is the following:

```
Problems occurred restoring workbench.
Unable to restore perspective: Workspace - Resource.
Unable to create editor: Web Browser.
Could not create view: com.ibm.etools.j2ee.ui.view.J2EENavigator
Could not create view: com.ibm.etools.webtools.WebView
Could not create view: com.ibm.sed.library.libraryView
Could not create view: com.ibm.etools.server.ui.ServersView
Could not create view: org.eclipse.debug.ui.ConsoleView
```

The error messages have no impact on the migration of the workspace. After closing the errors dialogs, all the projects in the workspace are fully operational.

Removing backward compatibility

Compatibility with earlier versions of WebSphere Studio Application Developer can be totally removed from a project created in IBM Rational Application Developer for WebSphere Software or a project migrated from an earlier version of WebSphere Studio. This is done only if the user determines the project should no longer be either interoperable or backward compatible with WebSphere Studio Application Developer V5.1.x.

When a V5.1.x project is opened in the Rational Application Developer V6.0 workspace, or new J2EE 1.2 or 1.3 specification projects are created in V6.0, a .compatibility file is automatically created in the project directory. The .compatibility file is used to track the timestamps of project resources when these resources are migrated. The .compatibility file should therefore not be edited or deleted by users.

Backward compatibility can be removed if it is no longer required by the user. To remove backward compatibility:

1. Right-click an Enterprise Application project; select the **Remove Compatibility** menu option from the pop-up.

   A dialog launches, asking for confirmation to remove the backward compatibility of the Enterprise Project and all the modules and utility projects nested under the project.

2. Click **Yes** to continue with the Remove Compatibility operation.

Once the Remove Compatibility operation is run, the Enterprise Application and all the Module and Utility projects nested under the Enterprise Applications project are no longer backward compatible with WebSphere Studio Application Developer 5.1.x.

JavaServer Faces

Projects developed on WebSphere Studio Application Developer V5.1 that use JavaServer Faces (JSF) in either Web projects or client applications should migrate the JSF components to the most current levels found in IBM Rational Application Developer for WebSphere Software. For more details, see the online help topics *Migrating JavaServer Faces resources in a Web project* and *Migrating JavaServer Faces resources with Faces Client Components*.

Debugger changes

The debugging facilities in IBM Rational Application Developer for WebSphere Software have changed from V5. You may need to change settings manually. For more details, see the online help topic *Debugger changes*.

WebSphere Data Objects

The WebSphere Data Objects (WDO) APIs introduced in WebSphere Studio Application Developer V5.1 have been replace by Service Data Objects (SDO). Projects developed in WebSphere Studio Application Developer V5.1 that use WDO are automatically migrated to use SDO when the server is retargeted to V6. For more details, see the online help topic *WDO to SDO migration*.

For more information, see the *IBM Rational Application Developer for WebSphere Software Migration Guide*. The Migration Guide is found in these locations:

- First installation media disk as the file migrate.html in the IBM Rational Application Developer for WebSphere Software installation directory.
- *Installing and Migrating* book in the IBM Rational Application Developer for WebSphere Software online help. See 1.3, “How to obtain more information” on page 9 for information about launching the online help.

3

Changing application code

This chapter discusses application programming interfaces that have become obsolete.

3.1 J2EE compatibility

WebSphere Application Server Version 6 implements the J2EE 1.4 specification. Each succeeding specification level supports the preceding levels. As long as your application complies with these J2EE specifications, your applications should continue to work on WebSphere Application Server Version 6 unmodified. This makes the process of migration of applications written for J2EE 1.3 and J2EE 1.2 easier because it allows you to postpone the application code migration. You can run existing J2EE 1.3 and 1.2 applications in WebSphere Application Server Version 6 alongside J2EE 1.4 applications until it becomes necessary to move up to the next level. When migration is required, the process of migration to J2EE 1.4 can be achieved incrementally. In addition to forward compatibility of J2EE specifications, WebSphere programming extensions are also forward compatible.

You need to consider both inconsistencies and deprecations. Inconsistencies require changes in the code to ensure proper functioning of the migrated code. Deprecations imply that the API or model change will not be supported in future products; this is an optional but highly recommended change. Deprecations occur both for the J2EE specifications and the WebSphere extensions.

If an application does not use any of the deprecated or inconsistent APIs, then the application usually only needs to be redeployed. Redeployment is accomplished either by using automatic migration utilities or by manual installation. See Chapter 8, “Distributed runtime administration overview” on page 97 for an overview of automatic migration utilities. See Chapter 12, “Distributed runtime application migration examples” on page 185 for examples of how to use the automatic migration utilities and how to perform manual installation.

3.1.1 Incremental migration

Application code migration is not an all or nothing proposition. Forward J2EE compatibility allows you to migrate to a newer WebSphere runtime without any code changes. Changing application requirements may force migrating the code to higher version levels of J2EE at some later date. Once it becomes clear that migration of the code is required, it is possible to migrate only those applications or application components that need to use the newer features. The rest of the application components can remain unchanged. Figure 3-1 shows a conceptual example of migrating applications incrementally.

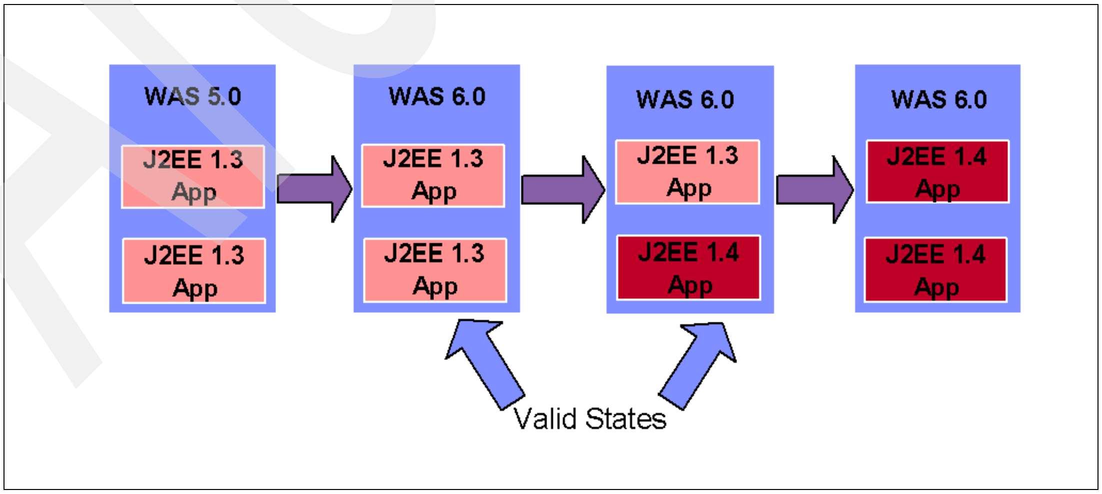


Figure 3-1 Incremental migration from J2EE 1.3 to J2EE 1.4

Incremental migration is also possible within an application. A J2EE 1.4 Enterprise Application Archive (EAR) file can contain J2EE 1.2 or J2EE 1.3 modules, as shown. J2EE is so flexible that EJB 2.1 modules can define EJB 2.0 or EJB 1.1 beans. If necessary, you can upgrade a selected set of your beans to take advantage of specific features. You can also leave the Web module at the J2EE 1.3 level, but migrate the EJB module to J2EE 1.4. This is an incremental way of migrating applications until they conform entirely to the new J2EE standard. Figure 3-2 shows a conceptual example of incrementally migrating modules within an application.

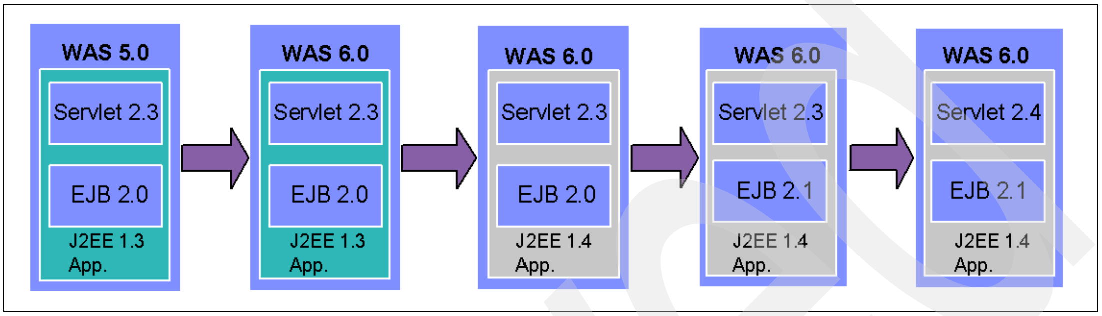


Figure 3-2 Incremental migration mixing modules of different levels

3.1.2 When to redesign

An application that conforms to J2EE specifications remains forward compatible; therefore, it can be redeployed on a newer version of the WebSphere Application Server without any code changes. However, there are cases when the application needs to be recompiled on a newer JDK™, or needs to be repackaged with a new structure. In some cases, where the code does not strictly conform to the specification or latest best practice paradigms, the code may need to be redesigned as part of the migration effort to make use of these newer functions and paradigms. See the article *Top 10 J2EE Best Practices* at this address:

http://www.ibm.com/developerworks/websphere/techjournal/0405_brown/0405_brown.html

3.2 Code incompatibilities and deprecations

When migrating an application from one version of WebSphere Application Server to the next, one needs to be aware of these issues:

- Deprecated models and API
- Incompatibilities
- New features and enhancements

Deprecation is the classification of a function or interface as obsolete. The function is still functional, but it is intended to become an incompatibility in the future. A deprecated class or interface is supported for two full product releases or three full years, whichever is longer. The removal window is measured from the point at which the deprecation is announced.

For the purpose of this chapter, we do not discuss the migration of code to exploit the new features with the exception of areas where there has been a significant model change, such as Web Services Integration and the new messaging bus. Instead, we concentrate on making existing J2EE code compatible with both J2EE 1.4 and WebSphere Application Server Version 6 runtimes, and look for areas of incompatibility and deprecation that one must examine as part of the migration process. In most cases, even if the application code is compatible with J2EE 1.4 (source code compatibility), it is advisable to recompile the code using the JDK from WebSphere Application Server Version 6, and use the deployment tools provided to package and redeploy the code to the WebSphere Application Server Version 6 runtime environment to ensure any packaging, deployment, or compile issues are ruled out.

In the Java environment, incompatibilities could be at the binary level or the source level. It is rare to see binary level incompatibility in applications that are being migrated, so it is not generally expected that they would require recompilation using the newer JDK that is supplied with WebSphere Application Server Version 6 or with the IBM Rational Application Developer for WebSphere Software. Source code level incompatibility is one area that one would definitely need to remedy; however, deprecated functions would still run.

For a complete list of all the incompatibilities, including binary incompatibilities, refer to the Sun J2EE documentation:

http://java.sun.com/j2se/1.4/compatibility.html

In considering the incompatibilities and deprecations, we look at these major areas:

- J2EE differences
- J2SE/JDK differences
- Application Server runtime differences
- Programming Model Extension differences
- Third Party Library differences

3.2.1 J2EE 1.3 versus J2EE 1.4 differences

There are two main areas of model change in J2EE 1.4. The first and most important area is Web Services integration, with the introduction of JAX-RPC and SAAJ APIs. This provides the basic and standards-based Web services support that is both portable and interoperable. This new specification also describes the packaging and deployment requirements for J2EE applications that provide and use these Web services. The second important area of change is a new messaging bus architecture supported by MDBs in EJB 1.1, JMS 1.1. This change affects both application APIs and how JMS providers are configured. WebSphere Application Server also provides a new embedded JMS messaging engine as an alternative to the WebSphere MQ that is included in WebSphere Application Server Version 5.

These changes, taken together, represent model change in a J2EE world that requires an architectural change in the applications being migrated. In this section, we discuss in detail how applications can be redesigned for Web services and new message bus architecture in WebSphere Application Server Version 6. For full details about what has changed in J2EE 1.4, refer to the Sun documentation:

http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf

Table 3-1 compares the J2EE application prerequisites for WebSphere Application Server versions V3.5, V4.0, V5.0 and V6.0 and their respective J2EE levels.

Table 3-1 J2EE specification levels supported by the WebSphere versions

	WAS V3.5 (Pre-J2EE)	**WAS V4.0 J2EE 1.2**	**WAS V5.0 J2EE 1.3**	**WAS V6.0 J2EE 1.4**
J2EE		1.2	1.3	1.4
EJB	1.0	1.1	1.1/2.0	2.1[b]
Servlet	2.1/2.2[a]	2.2	2.3[b]	2.4[b]
JSP™	.91/1.0/1.1[a]	1.1	1.2[b]	2.0
JMS		1.0	1.02	1.1

	WAS V3.5 (Pre-J2EE)	**WAS V4.0 J2EE 1.2**	**WAS V5.0 J2EE 1.3**	**WAS V6.0 J2EE 1.4**
JTA	1.01	1.01	1.01	1.01
JDBC™	1.1/2.0	2.0[c]	2.0[c]	3.0
JAF		1.0	1.0	1.0
RMI/IIOP	1.0	1.0	1.0	1.0
Java Mail		1.1	1.2	1.3
JNDI	1.2	1.2	1.2	1.2
JAXP			1.1	1.2
Connector		1.03	1.0	1.5
JAAS			1.0	1.0
Web Services				1.1
JAX-RPC				1.1
SAAJ				1.2
JAXR				1.0
J2EE management				1.0
JMX™				1.2
J2EE Deployment				1.1
JACC				1.0

Notes:

[a] Servlet 2.2/ JSP 1.1 introduced in V3.5.2

[b] Superset of preceding API

[c] Plus extension

Most of the changes from J2EE 1.3 to J2EE 1.4 are additive, and hence most code, including deprecated code, continues to work properly. However, it is always a good programming practice to deal with all the deprecations in the code at the particular specification level in order to avoid possible future problems. Deprecated code is sometimes necessary in cases where the same application needs to continue to run on the previous J2EE specification levels as well as on the newer J2EE level.

We have intentionally narrowed the discussion here to J2EE 1.3 to limit scope. Migrations from previous versions of WebSphere runtime have been discussed in a previously released migration redbook, *Migration to WebSphere V5.0 An End-to-End Migration Guide,* SG24-6910. This allows developers to migrate their applications to J2EE 1.3 as a stepping stone to the J2EE 1.4 level discussed here.

J2EE 1.4 deprecation

Applications written to J2EE APIs that are deprecated in J2EE 1.4 do work and may continue to work in future versions of J2EE. Our recommendation is to remove deprecated APIs when it is either strictly required due to obsolescence or when it is convenient to do so. It is important to check in each release to see what has actually been deprecated rather than assume the default deprecation policy.

Complete details of all the J2EE 1.4 deprecations can be found in the Sun documentation at:

`http://java.sun.com/j2ee/1.4/docs/api/deprecated-list.html`

Web services migration

The biggest change in the J2EE 1.4 platform is the strong integration of Web services. J2EE components can now interact with Web services and implement them in a standard, portable fashion. For example, a standard way to look up, invoke, and implement Web services is to use either stateless session beans or plain Java classes mapped to a servlet.

In versions prior to Version 5.0.2, WebSphere Application Server uses Apache SOAP-engine-based Web services. Apache SOAP is not compliant with the Web services interoperability standards and completely lacks Web services security.

WebSphere Application Server V5.0.2 and WebSphere Application Server V5.1 improve on the earlier versions to give us an early preview of the standards-based Web services in J2EE 1.4. Key to these are JAX-RPC 1.0 (JSR-101) which is a new standard API for programming Web services in Java, as well as the accompanying JSR 109, the new deployment model for Java Web services. These versions include an enhanced engine supporting Web services over both HTTP and JMS. Web services security and extensions have also been added to these releases. However, this functionality is based on Draft 13 April 2003 of the Web services security specification, and there may be areas of functionality that differ from the final draft that was the basis for J2EE 1.4 Web services in WebSphere Application Server Version 6.

Table 3-2 WebSphere Web services standards support

WebSphere version	Apache SOAP supported?	JSR 101 - JSR 109 supported?
V4	Yes	No
V5	Yes	No
V5.0.2, V5.1	Yes	Yes
V6	Yes	Yes

Note that Apache SOAP API and its deployment model have been deprecated in WebSphere Application Server Version 6. You should re-implement with J2EE 1.4 Web services, which were introduced in WebSphere Application Server V5.0.2. Applications developers can use the migration wizard in IBM Rational Application Developer for WebSphere Software to migrate these earlier Web services components to the new J2EE 1.4 specification.

Using the Migration Wizard for Web services

The Web services artifacts migrated using the J2EE Migration Wizard are:

- Web services descriptor
- Web services client descriptor
- JAX-RPC mapping descriptor

Migrating Web service deployment descriptors

Any Web services deployment descriptors contained in J2EE 1.3 projects migrated to J2EE 1.4 specification level are migrated from JSR-109 V1.0 for J2EE 1.3 to J2EE 1.4. Users should note the change in the representation of qualified names of Web services deployment descriptors in J2EE 1.4.

Web service deployment descriptors, as defined by JSR-109 V1.0, consist of webservices.xml, webservicesclient.xml, and all JAX-RPC mapping deployment descriptors that are referenced by webservices.xml and webservicesclient.xml.

As with other J2EE deployment descriptors, migration modifies the structure of the information contained in the descriptors to be J2EE 1.4 compliant. One structural change that is particular to the Web service deployment descriptors is the change to the way qualified names are represented. In JSR-109 V1.0, qualified names are represented using a sequence of two elements, <namespaceURI> and <localpart>, which contain the namespace URI and the local part of the name, respectively. Qualified names in J2EE 1.4 are based on the XMLSchema QName type, which uses XML namespaces.

Web services descriptor (webservices.xml)

The webservices.xml deployment descriptor is present in Web projects and EJB projects that contain J2EE Web services. Both the <wsdl-port> element and the <soap-header> element contain qualified names and their contents are migrated to the J2EE 1.4 format.

For example, if <wsdl-port> is represented as follows before migration:

```
<wsdl-port>
  <namespaceURI>http://addressbook.webservice</namespaceURI>
  <localpart>AddressBook</localpart>
</wsdl-port>
```

after migration, <wsdl-port> appears as:

```
<wsdl-port xmlns:pfx="http://addressbook.webservice">pfx:AddressBook</wsdl-port>
```

Note that `pfx` is used as the namespace prefix for all migrated qualified names.

Web services client descriptor (webservicesclient.xml)

The webservicesclient.xml deployment descriptor is present in J2EE 1.3 Web projects, EJB projects, and Application Client projects that contain J2EE Web service clients. During migration from J2EE 1.3 to 1.4, the contents of webservicesclient.xml are migrated and moved to the deployment descriptor for the project. The process that occurs is as follows:

- For Web projects, all <service-ref> elements in webserivcesclient.xml are moved under the <web-app> element in web.xml.
- For Application Client projects, all <service-ref> elements in webservicesclient.xml are moved under the <application-client> element in application-client.xml.
- For EJB projects, all <service-ref> elements within a <component-scoped-refs> in the webserviceclient.xml are moved under the corresponding <enterprise-bean> in ejb-jar.xml.

Webservicesclient.xml is then deleted.

Both the <service-qname> element and the <soap-header> element contain qualified names, and their contents are migrated to the J2EE 1.4 format. For example, if <service-qname> is represented as follows before migration:

```
<service-qname>
  <namespaceURI>http://addressbook.webservice</namespaceURI>
  <localpart>AddressBookService</localpart>
</service-qname>
```

after migration, <service-qname> appears as:

```
<service-qname
xmlns:pfx="http://addressbook.webservice">pfx:AddressBookService</service-qname>
```

Note that `pfx` is used as the namespace prefix for all migrated qualified names.

JAX-RPC mapping descriptor

Both webservices.xml and webservicesclient.xml can reference one or more JAX-RPC mapping deployment descriptors. In webservices.xml, these references are contained in the <jaxrpc-mapping-file> element under each <webservice-description>. In webservicesclient.xml, these references are contained in the <jaxrpc-mapping-file> element under each <service-ref>. During migration from J2EE 1.3 to 1.4, all the JAX-RPC mapping deployment descriptors referenced in webservices.xml and webservicesclient.xml are migrated. Migration includes migrating all qualified names to the J2EE 1.4 format.

Migrating Web services security

Secure Web services are not migrated by the J2EE Migration Wizard when Web services are migrated from WebSphere Studio Application Developer Version 5.1 to IBM Rational Application Developer for WebSphere Software Version 6. The migration of Secure Web Services requires manual steps.

After the J2EE migration, the secure binding and extension files must be migrated manually to V6.0 as follows:

1. Double-click the **webservices.xml** to bring up the Web services editor.
2. Select the **Binding Configurations** tab to edit the binding file.
3. Add all the necessary binding configurations under the new sections “Request Consumer Binding Configuration Details” and “Response Generator Binding Configuration Details.”
4. Select the **Extension** tab to edit the extension file.
5. Add all the necessary extension configurations under the new sections “Request Consumer Service Configuration Details” and “Response Generator Service Configuration Details.”
6. Save and exit the editor.

Migrating to Servlet 2.4

In J2EE 1.3, Servlet 2.3 specification, the method HttpSessionListener sessionDestroyed was defined as notification that a session was invalidated. As of Servlet 2.4, this method is changed to notification that a session is about to be invalidated so that it notifies before the session invalidation. If the code assumed the previous behavior, it must be modified to match the new behavior.

The following methods are added in the ServletRequest interface in this version of the specification:

- public int getRemotePort() returns the Internet Protocol (IP) source port of the client or last proxy that sent the request.
- public java.lang.String getLocalName() returns the host name of the IP interface on which the request was received.
- public java.lang.String getLocalAddr() returns the IP address of the interface on which the request was received.
- public int getLocalPort() returns the IP port number of the interface on which the request was received.

Be aware that this addition causes source incompatibility in some cases, such as when a developer implements the ServletRequest interface. In this case, ensure that all the new methods are implemented correctly.

In Servlet 2.4, SingleThreadModel Interface is deprecated due to its use in a way that is potentially not thread safe. The best programming practice is to avoid the use of instance variables, session attributes, and static variables. If it is still necessary, we recommend that you use synchronized code blocks.

For further details about the J2EE Servlet specifications, download the specifications from:

http://java.sun.com/j2ee/1.4/download.html#platformspec

Migrating Web application artifacts

It is important to migrate the Web module artifacts along with the application code to ensure proper functioning. Artifacts of a Web deployment descriptor are migrated by the J2EE Migration Wizard when a J2EE 1.3 specification level Web project is migrated to the J2EE 1.4 specification level.

The following Web application artifacts are migrated.

Authentication constraints

The Description object did not exist in J2EE 1.3; the description was an attribute on the Authentication Constraint. J2EE 1.4 includes a Description object that has two attributes: language and value. So, the description is set to `value` in the Description object.

Security constraints

As in the prior item, the description was an attribute of SecurityConstraint. In 1.4, there is a new Description object with the attributes language and value. So, the description is set to `value` in the Description object.

Web application

The ContextParam object in J2EE 1.3 does not exist in J2EE 1.4. In 1.4, there is a common object called ParamValue. The Web root object in 1.4 contains a list of ParamValue objects, which are like the list of ContextParams on a Web 1.3 root object. So, to get to the contectParams in 1.4, you need to go through the ParamValue object.

The description string attribute of the ContextParam object in the J2EE 1.3 specification level has been moved to a Description object in ParamValue in J2EE 1.4.

The TagLib object in J2EE 1.3 has been moved to the JSPConfig object in J2EE 1.4. The JSPConfig object belonged to the Web root object in 1.3.

The InitParam object in J2EE 1.3 also does not exist in 1.4. In 1.4, there is a common object called ParamValue. The Servlet and Filter objects in 1.4 contain a list of ParamValue objects like the list of InitParam on a Web 1.3 root object. To get to the InitParam in 1.4, you need to go through the ParamValue objects associated with Servlet and Filter objects.

Migrating to JSP 2.0

JSP Specification 1.2 shipped with the J2EE 1.3 SDK. Where possible, the JSP 2.0 specification attempts to be fully backward compatible with the JSP 1.2 specification. In some cases, there are ambiguities in the JSP 1.2 specification that have been clarified in the JSP 2.0 specification. Because some JSP 1.2 containers behave differently, some applications that rely on container-specific behavior may need to be adjusted to work correctly in a JSP 2.0 environment.

The following is a list of known backward compatibility issues of which developers who use JSP technology should be aware:

- Tag Library validators that are not namespace-aware and that rely solely on the prefix parameter might not correctly validate some JSP 2.0 pages. This is because the XML view might contain tag library declarations in elements other than jsp:root, and might contain the same tag library declaration more than once, using different prefixes.

- The uri parameter should always be used by tag library validators instead. Existing JSP pages with existing tag libraries do not create any problems.
- In JSP specification versions previous to JSP 2.0, JSP pages in XML syntax and those in standard syntax determined their page encoding in the same fashion, by examining the pageEncoding or contentType attributes of their page directive, defaulting to ISO-8859-1 if neither is present. As of JSP Specification V2.0, the page encoding for JSP documents is determined as described in section 4.3.3 and appendix F.1 of the J2EE 1.4 XML specification, and the pageEncoding attribute of those pages is only checked to make sure it is consistent with the page encoding determined as per the XML specification. As a result of this change, JSP documents that rely on their page encoding to be determined from their pageEncoding attribute can no longer be decoded correctly.
- These JSP documents must be changed to include an appropriate XML encoding declaration. Additionally, in the JSP 1.2 Specification, page encodings are determined on a per translation unit basis, whereas in the JSP 2.0 Specification, page encodings are determined on a per file basis. For example, consider a JSP page A that specifies a page encoding. It statically includes JSP page B, which does not specify an encoding. Under JSP 1.2, B inherits the encoding from A. Under JSP 2.0, B picks up the default encoding and does not inherit the encoding from A.
- The JSP container uses the version of web.xml to determine the default behavior of various container features.
- The following is a list of items of which JSP developers should be aware when upgrading their web.xml from Servlet Version 2.3 Specification to Servlet Version 2.4 Specification:
  - EL expressions are ignored by default in applications created with JSP 1.2 technology. When upgrading a Web application to the JSP 2.0 Specification, EL expressions are interpreted by default. The escape sequence \$ can be used to escape EL expressions that should not be interpreted by the container.
  - The isELIgnored page directive attribute or the el-ignored configuration element can deactivate EL for entire translation units. Users of JSTL 1.0 need to either upgrade their taglib/ imports to the JSTL 1.1 URIs, or they need to use the _rt versions of the tags (for example, c_rt instead of c, or fmt_rt instead of fmt).

Migrating to EJB 2.1

The EJB 2.1 specification includes important enhancements to EJB-QL, a new timer service, and the extension of message-driven beans (MDBs) to support any type of messaging system, not just JMS.

A major change in EJB 2.1 is that MDBs have been extended to support any message type, not just JMS. This allows configuration of any proprietary messaging provider and incoming asynchronous messages handling by business logic written to the MDB programming model. This is done by adding a J2EE Connector resource adapter, provided by the vendor, that translates incoming messages into calls to the MDB. See “Migrating to JMS 1.1” on page 29 for more information about JMS 1.1.

Migrating EJBs

The J2EE Migration Wizard supports the migration of Enterprise Bean deployment descriptors from J2EE 1.3 specification level EJB resource to J2EE 1.4 specification level. Stateless session beans and message-driven beans are migrated to J2EE 1.4.

The J2EE Migration Wizard migrates stateless session beans that are defined as Service Endpoint Interfaces (SEIs) in an EJB project to the J2EE 1.4 specification level by creating new SEIs on the stateless session bean.

The J2EE 1.4 specification requires that an SEI be defined on a stateless session bean if the session bean is to be used as a Web Services endpoint. During the migration of an EJB JAR, all session beans in the EJB project get a new SEI that uses the name used in the webservices.xml descriptor of the EJB project.

Example 3-1 shows how the metadata of an EJB project looks at the J2EE 1.3 specification level. The <service-endpoint-interface> and <service-impl-bean> tags define stateless session bean “EchoEJB” as a service endpoint in the Web services descriptor at the J2EE 1.3 specification level prior to migration.

Example 3-1 J2EE 1.3 Web services project

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE webservices PUBLIC "-//IBM Corporation, Inc.//DTD J2EE Web services 1.0//EN"
"http://www.ibm.com/webservices/dtd/j2ee_web_services_1_0.dtd">
   <webservices id="WebServices_1084831328093">
      <webservice-description id="WebServiceDescription_1084831328093">
         <webservice-description-name>EchoEJBService</webservice-description-name>
         <wsdl-file>META-INF/wsdl/EchoEJB.wsdl</wsdl-file>
         <jaxrpc-mapping-file>META-INF/EchoEJB_mapping.xml</jaxrpc-mapping-file>
         <port-component id="PortComponent_1084831328103">
            <port-component-name>EchoEJB</port-component-name>
            <wsdl-port id="WSDLPort_1084831328103">
               <namespaceURI>http://test</namespaceURI>
               <localpart>EchoEJB</localpart>
            </wsdl-port>
            <service-endpoint-interface>test.EchoEJB</service-endpoint-interface>
            <service-impl-bean id="ServiceImplBean_1084831328103">
               <ejb-link>EchoEJB</ejb-link>
            </service-impl-bean>
         </port-component>
      </webservice-description>
</webservices>
```

Example 3-2 shows a J2EE 1.4 EJB Deployment Descriptor for the stateless session bean “EchoEJB” with service endpoint interface created by the migration wizard. The <service-endpoint> tag defines “EchoEJB” as a service endpoint in J2EE 1.4 specification level.

Example 3-2 J2EE 1.4 Web services project

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejb-jar>
<ejb-jar id="ejb-jar_ID" version="2.1" xmlns="http://java.sun.com/xml/ns/j2ee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/ejb-jar_2_1.xsd">
   <display-name>
   EchoEJBProject</display-name>
   <enterprise-beans>
      <session id="EchoEJB">
         <ejb-name>EchoEJB</ejb-name>
         <home>test.EchoEJBHome</home>
         <remote>test.EchoEJB</remote>
         <service-endpoint>test.EchoEJB</service-endpoint>
         <ejb-class>test.EchoEJBBean</ejb-class>
         <session-type>Stateless</session-type>
         <transaction-type>Container</transaction-type>
      </session>
   </enterprise-beans>
</ejb-jar>
```

Migrating message-driven beans

The J2EE Migration Wizard supports the migration of EJB 2.0 message-driven beans to EJB 2.1 Java Message Service (JMS) type message-driven beans.

Message-driven beans were introduced in EJB 2.0 to support the processing of asynchronous messages from a Java Message Service. The EJB 2.1 specification expands the definition of the message-driven bean so that it can support any messaging system, not just JMS.

The EJB 2.0 message-driven bean artifacts migrated are:

- acknowledgeMode
- messageSelector
- destinationType
- subscriptionDurablity

Some of the EJB 2.0 message-driven bean elements are replaced with activation-config properties. The property names and values used in the activation-config to describe the messaging service vary depending on the type of message service used. However, EJB 2.1 defines a set of fixed properties for JMS-based message-driven beans.

Example 3-3 shows the elements of a sample bean in EJB 2.0. Contrast this with Example 3-4, which shows how the elements appear in EJB 2.1.

Example 3-3 EJB 2.0 descriptor for message-driven bean

```
<message-driven id="Mdb20">
     <ejb-name>Mdb</ejb-name>
     <ejb-class>ejbs.MdbBean</ejb-class>
     <transaction-type>Bean</transaction-type>
     <message-selector>mdbMessage</message-selector>
     <acknowledge-mode>Auto-acknowledge</acknowledge-mode>
     <message-driven-destination>
      <destination-type>javax.jms.Topic</destination-type>
      <subscription-durability>Durable</subscription-durability>
      </message-driven-destination>
</message-driven>
```

Example 3-4 EJB 2.1 descriptor for message-driven bean

```
<message-driven id="Mdb21">
  <ejb-name>Foo/ejb-name>
  <ejb-class>ejbs.FooBean</ejb-class>
   <messaging-type>javax.jms.MessageListener</messaging-type>
   <transaction-type>Bean/transaction-type>
   <message-destination-type>javax.jms.Topic</message-destination-type>
    <activation-config>
    <activation-config-property>
      <activation-config-property-name>destinationType</activation-config-property-name>
      <activation-config-property-value>javax.jms.Topic</activation-config-property-value>
    </activation-config-property>
    <activation-config-property>
```

```
<activation-config-property-name>subscriptionDurability</activation-config-property-name>
        <activation-config-property-value>Durable</activation-config-property-value>
     </activation-config-property>
     <activation-config-property>
        <activation-config-property-name>acknowledgeMode</activation-config-property-name>

<activation-config-property-value>AutoAcknowledge</activation-config-property-value>
     </activation-config-property>
     <activation-config-property>
      <activation-config-property-name>messageSelector</activation-config-property-name>
      <activation-config-property-value>fooSelector</activation-config-property-value>
     </activation-config-property>
</activation-config>
</message-driven>
```

Migrating to JCA 1.5

The J2EE Migration Wizard migrates the artifacts of a JCA 1.0 descriptor to JCA 1.5. These changes are made:

- The following elements are moved from ResourceAdaptor object to OutboundResourceAdaptor object:
  - reauthenticationSupport
  - transactionSupport
- The following elements are moved from ResourceAdaptor object to the ConnectionDefinition object:
  - managedConnectionFactoryClass
  - connectionFactoryInterface
  - connectionFactoryImplClass
  - connectionInterface
  - connectionImplClass
- The outboundResourceAdaptor holds a list of ConnectionDefinitions. Therefore, the ConnectionDefinition is added to the ConnectionDefinitions held by OutboundResourceAdaptor.
- The OutboundResourceAdaptor is held by the ResourceAdaptor object.
- The AuthenticationMechanism has undergone some changes in JCA 1.5 where customAuthMechType is replaced by authenticationMechanism and authenticationType is replaced by authenticationMechanism.
- The description attribute is replaced with a description object where the description string is set to a value element in the description object for the following objects:
  - SecurityPermission
  - AuthenticationMechanism
  - ConfigurationProperties

Migrating to JMS 1.1

Although JMS 1.0 applications would continue to run in J2EE, there are some new features, requirements, and restrictions that come with JMS 1.1 and message-driven beans in J2EE 1.4 of which application developers have to be aware.

For a more in-depth look at the issues, the reader is referred to an article by David Currie on the IBM DeveloperWorks site at:

http://www-106.ibm.com/developerworks/java/library/j-getmess/

3.2.2 J2SE

The J2EE runtime consists of the Web container, the EJB container, the applet container, and the application client container. The containers are J2EE runtime environments that provide required services to the application components. For example, the application client container provides application clients with direct access to the J2EE required database through the Java API for connectivity with database systems, the JDBC API. Similar access to databases is provided to JSP pages and servlets by the Web container, and to enterprise beans by the EJB container.

WebSphere Application Server V5.1 is the first version to adopt J2SE™ 1.4. A number of changes to the Java language have taken place in J2SE 1.4. These changes may impact your application. For example, JDK 1.4 includes an XML implementation that may potentially conflict with XML libraries your application uses. In most cases, the changes have no impact.

Compatibility

For full details about all the J2SE 1.4 incompatibilities, refer to this Sun Java™ article:

http://java.sun.com/j2se/1.4/compatibility.html

Deprecation

For a full list of deprecated APIs for J2SE 1.4 platform, refer to this Sun Java article:

http://java.sun.com/j2se/1.4.2/docs/api/deprecated-list.html

JDBC 3.0

The JDBC 3.0 API, included as part of the J2SE 1.4 platform, introduces two new interfaces and adds several new methods to existing interfaces. Drivers and applications that use earlier versions of the JDBC API are binary compatible with the J2SE 1.4 platform and run without any problems. However, the changes made in the JDBC 3.0 API are not source-compatible. Drivers and applications that implement the JDBC interfaces must be updated to reflect the changes in order to build successfully. Chapter 6 of the JDBC 3.0 Specification gives a complete list of what must be done to be compliant with the JDBC 3.0 API, and therefore source-compatible with the J2SE 1.4 platform. You can find the JDBC 3.0 Specification at this location:

http://java.sun.com/products/jdbc/download.html

For applications using JDBC APIs prior to JDBC 2.0, these are some significant deprecations that should be noted:

- java.sql.CallableStatement.getBigDecimal(int, int)
- java.sql.Date(int, int, int)
- java.sql.Date.getHours()
- java.sql.Date.getMinutes()
- java.sql.Date.getSeconds()
- java.sql.Date.setHours(int)
- java.sql.Date.setMinutes(int)
- java.sql.Date.setSeconds(int)
- java.sql.DriverManager.getLogStream()
- java.sql.DriverManager.setLogStream(PrintStream)
- java.sql.PreparedStatement.setUnicodeStream(int, InputStream,int)
- java.sql.ResultSet.getBigDecimal(int, int)
- java.sql.ResultSet.getBigDecimal(String, int)
- java.sql.ResultSet.getUnicodeStream(int)
- java.sql.ResultSet.getUnicodeStream(String)
- java.sql.Time(int, int, int)

- java.sql.Time.getDate()
- java.sql.Time.getDay()
- java.sql.Time.getMonth()
- java.sql.Time.getYear()
- java.sql.Time.setDate(int)
- java.sql.Time.setMonth(int)
- java.sql.Time.setYear(int)
- java.sql.Timestamp(int, int, int, int, int, int, int)

3.2.3 WebSphere API incompatibilities

There is one incompatibility that may affect your applications.

Slash character on the getResource path

WebSphere Application Server Version 6 changed its behavior relative to how it treats the path on the getResource and getResourceAsStream methods when called from a servlet context.

The J2EE Servlet 2.3 specification defines these methods such that the resource path is required to begin with a slash character. The specification is summarized for you here:

```
getResource(String)
public java.net.URL getResource(java.lang.String path)
throws MalformedURLException
```

This returns a URL to the resource that is mapped to a specified path. The path must begin with a / and is interpreted as relative to the current context root.

```
getResourceAsStream(String)
public java.io.InputStream getResourceAsStream(java.lang.String path)
```

This returns the resource located at the named path as an InputStream object. The data in the InputStream can be of any type or length. The path must be specified according to the rules given in getResource.

WebSphere Application Server Version 5 does not enforce the specification requirement that the path begin with slash. It accepts a path regardless of whether the path begins with slash. However, WebSphere Application Server Version 6 is more stringent and requires that the path begin with slash. Failure to do so results in a MalformedURLException exception.

Even if your code does not use getResource or getResourceAsStream, your application may encounter this incompatibility because your application uses a third-party library that then uses one of these methods. One third-party library that is known to have this problem is the Apache Struts framework.

You can eliminate this problem without changing any code. WebSphere Application Server Version 6 allows you to establish a compatibility mode such that the behavior is identical to that of V5. You need to set the custom property prependSlashToResource for the Web container of the application server running the application. This setting is global to the entire server and affects every application on that server. See 8.5.3, “getResource path syntax” on page 124 for information about how to set the prependSlashToResource property.

3.2.4 WebSphere API deprecations

There are several WebSphere-specific APIs that have been deprecated over various releases. Table 3-3 on page 32 shows a summary of those deprecations.

Table 3-3 WebSphere-specific deprecations

Function	Release when deprecated
UDDI Registry V2 EJB interface	6.0
UDDI4J V2	6.0
UDDI Utility Tools APIs	6.0
Apache SOAP implementation	6.0
Web Service Security Draft 13 specification-level	6.0
Package com.ibm.websphere.servlet.filter	6.0
MIME filtering	6.0
WebSphere JRAS Extensions API	6.0
Package com.ibm.websphere.product.product	6.0
Data Access Beans	6.0
PMI Client	6.0
J2EE Connector Architecture	6.0
Method Level Access Intent	6.0
getStackTrace	5.1
JSP tsx tags	5.1
Common Connector Framework	5.1
Security API	5.1

For more information about deprecations, see the InfoCenter article: *Deprecated features in Version 6.0*, found at:

`http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/rmig_deprecationlist.html`

UDDI Registry V2 EJB interface

This interface is included in WebSphere Application Server Version 6 for compatibility with V5. There is no replacement API.

UDDI4J V2

A client library is provided to simplify constructing and sending UDDI V3 requests from Java. This is the IBM UDDI V3 Client for Java, provided in uddiv3client.jar. The UDDI4J APIs (uddi4jv2.jar) may still be used. Migrate to V3 UDDI APIs.

UDDI Utility Tools APIs

All of the low-level APIs such as BusinessStub and ServiceStub are deprecated. Migrate to the PromoterAPI in place of these low-level APIs. The PromoterAPI provides the same functionality at a higher level of abstraction.

Apache SOAP Security

Apache SOAP Security, which uses an XML digital signature, is deprecated. Migrate to the JSR-109 implementation of Web services and Web Service Security 1.0.

Web Services Security Draft 13 Specification support

Applications should be migrated to the supported WSS 1.0 standard. The draft-level support does not provide interoperability with some third-party vendors since the message level has been changed between the draft and the WSS 1.0 implementation.

WSS 1.0 is only supported in J2EE 1.4 applications; therefore, you need to migrate applications to J2EE 1.4 first. Migration is a manual step.

Package com.ibm.websphere.servlet.filter

These classes are included in this package:

- ChainedRequest
- ChainedResponse
- ChainerServlet
- ServletChain

Redesign your application to use javax.servlet.filter classes. Starting from the Servlet 2.3 specification, javax.servlet.filter classes give you the capability to intercept requests and examine responses. They also allow you to achieve chaining functionality, as well as embellishing or truncating responses.

MIME filtering

MIME filtering is deprecated. MIME filters were first introduced in WebSphere Application Server V3.5 as a way for servlets to embellish, truncate, and modify the responses generated by other servlets, based on the MIME types of the output content. Migrate to javax.servlet.filter classes.

WebSphere JRAS Extensions API

The WebSphere JRAS Extensions API is deprecated in this release. No further enhancements are planned for JRAS support. Migrate to java.util.logging package (JSR47).

Package com.ibm.websphere.product.product

All the methods and fields in com.ibm.websphere.product.product and com.ibm.websphere.product.buildInfo classes are deprecated. Hence, the following methods from com.ibm.websphere.product.WASProduct class (which involves com.ibm.websphere.product.product and com.ibm.websphere.product.buildInfo objects) are deprecated:

```
public product getProductByFilename(String basename)
public product getProductById(String id)
public boolean productPresent(String id)
public boolean addProduct(product aProduct)
public boolean removeProduct(product aProduct)
public Iterator getProducts()
public Iterator getProductNames()
public String loadVersionInfoAsXMLString(String filename)
public String getProductDirName()
public static String computeProductDirName()
```

Use the following supported methods from com.ibm.websphere.product.WASDirectory:

```
public WASProductInfo getWASProductInfo(String id)
public boolean isThisProductInstalled(String id)
public WASProductInfo getWASProductInfoInstances()
public String getWasLocation()
```

Also, instead of getting product information (name, version, build level, build date) from the old WASProduct API (com.ibm.websphere.product.WASProduct), you should now use the following methods in the WASDirectory class to get that information:

```
com.ibm.webpshere.product.WASDirectory.getName(String)
com.ibm.webpshere.product.WASDirectory.getVersion(String)
com.ibm.webpshere.product.WASDirectory.getBuildLevel(String)
com.ibm.webpshere.product.WASDirectory.getBuildDate(String)
```

Data Access Beans

Data Access Beans APIs, which are in databeans.jar, are deprecated. Migrate to Service Data Objects (SDO). See this InfoCenter article for more information about SDO: *Data access with Service DataObjects*, found at:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/cdat_datsdo.html

PMI Client

The PMI Client API, which was introduced in V4.0 to programmatically gather performance data from WebSphere Application Server, is deprecated.

The Java Management Extension (JMX) interface, which is part of the J2EE specification, is the recommended way to gather WebSphere Application Server performance data. PMI data can be gathered from the J2EE-managed object MBeans, or from the WebSphere PMI Perf MBean. While the J2EE MBeans provide performance data about a specific component, the Perf MBean acts as a gateway to the WebSphere Application Server PMI service, and provides access to the performance data for all the components.

J2EE Connector Architecture

The following in the J2EE Connector Architecture runtime are deprecated:

- com.ibm.ws.management.descriptor.xml.ConnectionFactory.xml. getPoolContents method is replaced by showPoolContents. getAllPoolContents method is replaced by whoAllPoolContents.
- com.ibm.websphere.j2c.ConnectionManager interface is replaced by J2EE Connector Architecture 1.5 LazyAssociatableConnectionManager interface.
- com.ibm.websphere.j2c.ConnectionEventListener interface is replaced by J2EE Connector Architecture 1.5 LazyEnlistableConnectionManager interface.

Method-level access intent

Container-managed persistence (CMP) entity beans configured with method-level access intent may run into data access problems, such as a deadlock. Therefore, the method-level access intent is deprecated. Reconfigure CMP entity beans to use bean-level access intent.

com.ibm.websphere.servlet.error.ServletErrorReport.getStackTrace

This is not really a deprecation. It is really an incompatibility caused by a method signature collision with a method defined in J2SE 1.4. The changed class is com.ibm.websphere.servlet.error.ServletErrorReport. The return signature for getStackTrace is changed because java.lang.Throwable now defines the same method with a different return signature.

Old method signature:

```
public String getStackTrace();
// returns a String representation of the
exception stack
```

New method signature (JDK 1.4, WebSphere Application Server Version 5.1):

```
public StackTraceElement[] getStackTrace();
// returns an array of stack trace
elements
```

A replacement method that has the same signature as the old method is provided:

```
public String getStackTraceAsString();
// returns a String representation
of the Exception Stack
```

JSP tsx tags

Support for the following tsx tags in the JSP engine is deprecated:

- repeat
- dbconnect
- dbquery
- getProperty
- userid
- passwd
- dbmodify

Instead of using the tsx tags, you should use equivalent tags from the Java Server Pages Standard Tag Library (JSTL). JSTL is supported in WebSphere Application Server Version 6, and the tag library is shipped with the product. Use Table 3-4 as a guideline for converting tsx tags to JSTL tags.

Table 3-4 JSP tsx tags

tsx tag	JSTL tag
tsx:repeat	c:forEach
tsx:dbconnect	sql:setDataSource
tsx:dbquery	sql:query
tsx:getProperty	use standard EL syntax, for example, `c:out value="${book.title}"`, where `book` is the current index in the result set
tsx:userid	use the `user` attribute of the setDataSource tag
tsx:passwd	use the `password` attribute of the setDataSource tag
tsx:dbmodify	sql:update

Common Connector Framework

The following jar files are deprecated:

- ccf.jar
- ccf2.jar
- recjava.jar
- eablib.jar

The J2EE Connector Architecture APIs should be used instead of the Common Connector Framework.

Security APIs

These APIs have been deprecated:

- com.ibm.websphere.security.auth.WSPrincipal.getCredential should be replaced by either com.ibm.websphere.security.auth.WSSubject.getRunAsSubject or com.ibm.websphere.security.auth.WSSubject.getCallerSubject.
- com.ibm.websphere.security.auth.WSSecurityContext should be replaced by JAAS APIs.
- com.ibm.websphere.security.auth.WSSecurityContextException should be replaced by JAAS APIs.
- com.ibm.websphere.security.auth.WSSecurityContextResult should be replaced by JAAS APIs.

3.2.5 Programming Model Extensions

WebSphere Application Server Version 6 makes the WebSphere Programming Model Extensions available at more basic levels of the product line. The strategy for V4 and V5 makes these extensions available only in upper tier products such as WebSphere Application Server Enterprise and WebSphere Business Integration Server Foundation. The strategy for Version 6 makes most available in the most basic product tier, namely in WebSphere Application Server Express. The migration tools in IBM Rational Application Developer for WebSphere Software copy and migrate applications using extension features supported in Version 6. If the extension is not supported, the application is not copied over to Version 6.

Table 3-5 shows that the WebSphere products that contain the Programming Model Extensions have changed over time.

Table 3-5 IBM Programming Extensions mapped to containing products

Extension	**WebSphere V4**	**WebSphere V5**	**WebSphere V5.1**	**WebSphere V6**
Business Rule Beans	Enterprise	Enterprise	WBISF	WBISF
Process Choreography	Enterprise	Enterprise	WBISF	WBISF
Extended Messaging	Enterprise	Enterprise	WBISF	WBISF
CORBA C++ SDK	Enterprise	Enterprise	WBISF	Not Supported
Application Profiling	Enterprise	Enterprise	WBISF	Express / Base / ND
Work Area	Enterprise	Enterprise	WBISF	Express / Base / ND
Distributed Map		Enterprise	WBISF	Express / Base / ND
Startup Beans		Enterprise	WBISF	Express / Base / ND
Activity Session		Enterprise	WBISF	Express / Base / ND
Internationalization		Enterprise	WBISF	Express / Base / ND
Asynchronous Beans		Enterprise	WBISF	Express / Base / ND
Object Pool		Enterprise	WBISF	Express / Base / ND
Dynamic Query		Enterprise	WBISF	Express / Base / ND
Last Participant		Enterprise	WBISF	Express / Base / ND
Extended JTA		Enterprise	WBISF	Express / Base / ND

Extension	WebSphere V4	WebSphere V5	WebSphere V5.1	WebSphere V6
Backup Clusters		Enterprise	WBISF	ND
WSWG Filters		Enterprise	WBISF	ND

Extensions no longer supported

C++ CORBA applications are no longer supported in WebSphere Application Server Version 6. Re-implement with JAVA ORB API.

3.2.6 Third-party libraries

WebSphere Application Server treats third-party libraries the same as your application code. Libraries composed of Java code should continue to work with little modification on WebSphere Application Server Version 6. However, many libraries do have dependencies on particular versions of the application server. To complicate matters, most library vendors do not provide source code or permission to change their code, which can sometimes leave you at their mercy for support.

Parser compatibilities

Inclusion of the JAXP 1.2 specification means that J2EE 1.4 application servers now have XSLT engines, SAX 2.0, and DOM level 2 parsers included. You can write your XML processing code once and swap in third-party parsers or XSLT engines as necessary, without changes to the source code. This is a step forward from J2EE 1.3, which included only JAXP 1.0, which did not support XSLT or the latest DOM/SAX technologies. If your code relies on inclusion of your own parsers, you may need to change the application code as well as the class loader settings to take advantage of system parsers in WebSphere Application Server Version 6.

There are really two cases to consider when migrating from previous versions of WebSphere Application Server. The first case is where an application has an intentional dependency on the system XML parsing classes (org.apache.xerces and org.apache.xalan). You need to be aware that Xerces classes and version levels that were included with previous WebSphere Application Server versions cannot be relied upon. Previous versions of WebSphere Application Server included these for convenience, since they were not specified in J2SE, J2EE, or IBM programming extensions API.

In WebSphere Application Server Version 5.1 and Version 6, all classes became part of J2SE 1.4 standard, therefore the lib/xerces.jar no longer exists. WebSphere Application Server Version 5.1 and Version 6 components rely on the J2SE XML classes supplied with the runtime. Make sure that your migrated applications code and classpath conform to these newer version levels of the parser API.

The second case is where the application developer has packaged their own XML classes in the EAR, and due to class loader settings, ends up picking up the system classes. In this case, you may need to change the class loader settings to ensure you pick up the parser classes in the EAR. The class loader settings and defaults are the same between V5.x and V6.x, so if the settings were correctly set, they would remain correctly set, assuming that the default setting was also moved across during migration of the runtime configuration.

See 8.5.1, “Class loader” on page 119 for more information about the class loader settings changes.

4

Development migration scenarios

This chapter provides real examples of migrating an application to IBM Rational Application Developer for WebSphere Software with step-by-step instructions.

4.1 Migrating from WebSphere Studio Application Developer Version 5.1 to IBM Rational Application Developer for WebSphere Software Version 6

The migration examples in this section all start with the Plants By WebSphere application installed in WebSphere Studio Application Developer Version 5.1. This is a sample application that can be downloaded from the DeveloperWorks Web site at this location:

http://www-106.ibm.com/developerworks/websphere/library/samples/WASV501/plantsby.html

In addition to importing and configuring the Plants By WebSphere application into the WebSphere Studio Application Developer workspace, you need to create certain required tables in a Cloudscape™ database called PLANTSDB. See the documentation that comes with the sample for instructions on creating the database.

For all the examples in this section, IBM Rational Application Developer for WebSphere Software Version 6 can be installed on the same system as WebSphere Studio Application Developer, or on a different system.

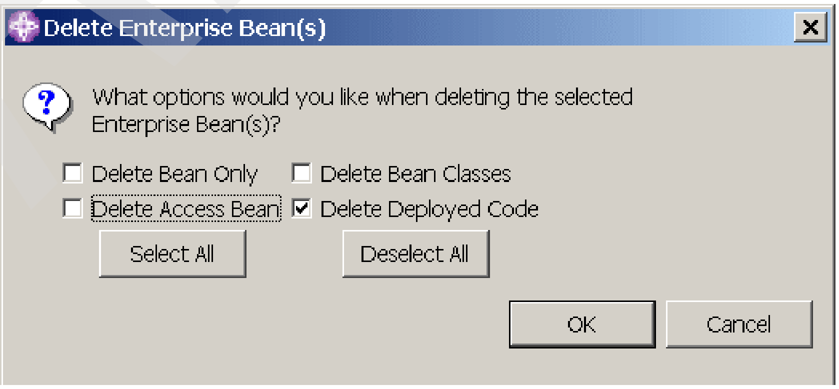
4.1.1 Example: Migrating Plants By WebSphere using the export/import method

This example shows you how to migrate a project by exporting your existing project into an EAR file and then importing that same EAR file into a new workspace.

Preparing the Enterprise Applications for export from WebSphere Studio Application Developer

Before exporting the Enterprise Applications from WebSphere Studio Application Developer, it is a good idea to delete the artifacts that were generated by that tool during EJB deployment by using the following steps:

1. Start WebSphere Studio Application Developer and open the workspace containing the PlantsByWebSphere application.
2. Switch to the J2EE hierarchy. Expand **EJB Modules** → **PlantsByWebSphereEJB** and expand both **Session Beans** and **Entity Beans**.
3. Highlight the six session beans and six entity beans.
4. Right-click the selection and choose **Delete**.



What options would you like when deleting the selected Enterprise Bean(s)?

☐ Delete Bean Only ☐ Delete Bean Classes
☐ Delete Access Bean ☑ Delete Deployed Code

Figure 4-1 Deleting existing EJBs

5. Deselect **Delete Bean Only**, **Delete Bean classes**, and **Delete Access Bean**. This should leave just **Delete Deployed code** checked.
6. Click **OK**.
7. Expand the **SupplierEJB** project and expand **Session Beans**.
8. Highlight the one session bean and delete the deployed code as with the previous project.
9. Switch to the **J2EE Navigator** view and expand **PlantsByWebSphereEJB** → **ejbModule**.
10. Delete all packages except com.ibm.websphere.samples.plantsbywebsphereejb.
11. Expand **SupplierEJB** → **ejbModule**.
12. Delete all packages except com.ibm.websphere.samples.supplierejb.

Exporting the Enterprise Applications from WebSphere Studio Application Developer

1. In the J2EE Navigator view, right-click the **Plants** project and select **Export**.
2. Select **EAR file** and click **Next**.
3. Type `C:\Plants.ear` for the Destination.
4. Check the box next to **Export source files**.
5. Click **Advanced** and check the box to **Include project build paths and meta-data files**. Click **OK**.
6. Click **Finish**.
7. Export the PlantsSupplier project into C:\PlantsSupplier.ear using the same procedure as used to export the previous project.
8. Close WebSphere Studio Application Developer.

Importing the Enterprise Applications into IBM Rational Application Developer for WebSphere Software

1. Start IBM Rational Application Developer for WebSphere Software and open a new workspace.

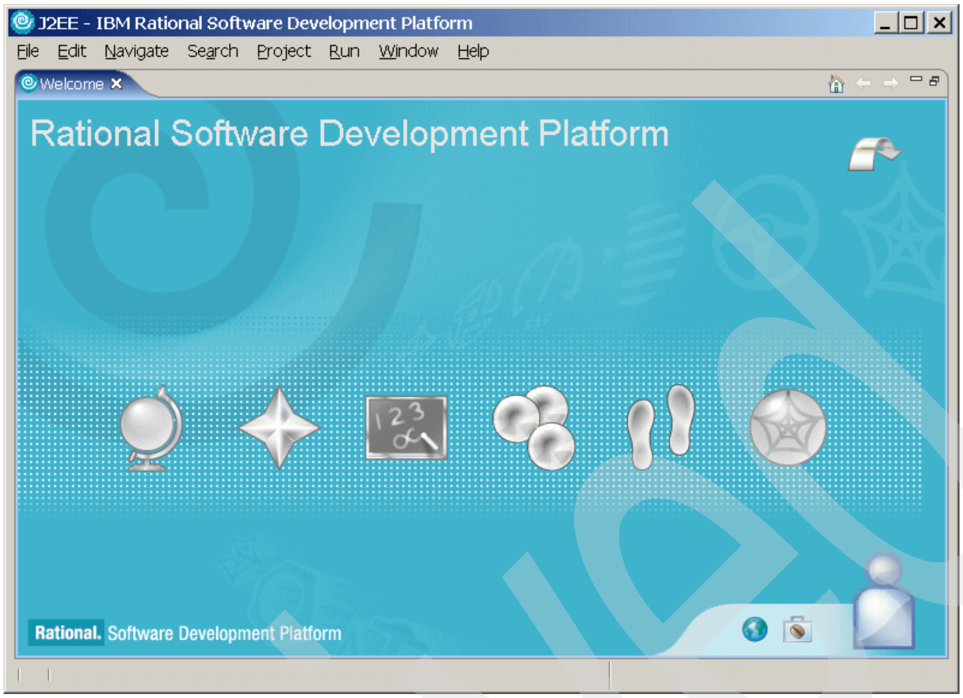
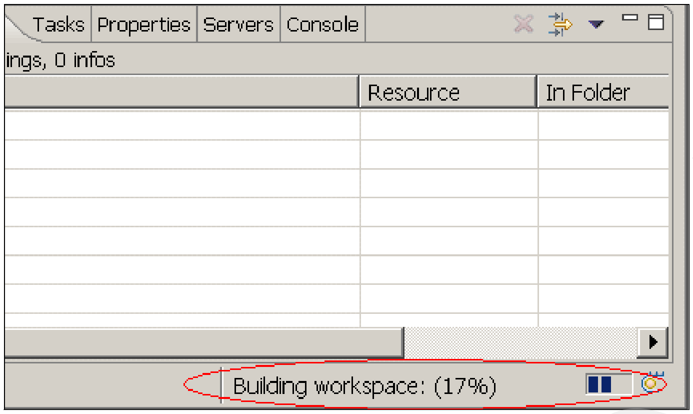

J2EE - IBM Rational Software Development Platform
File Edit Navigate Search Project Run Window Help
Welcome
Rational Software Development Platform
Rational. Software Development Platform

Figure 4-2 Initial screen for IBM Rational Application Developer for WebSphere Software

2. Close the Welcome view to access the Workbench.
3. Select **File** → **Import**. Select **EAR file** for the import source and click **Next**.
4. Browse to the Plants EAR that you exported from WebSphere Studio Application Developer.
5. Confirm that the EAR project is set to `Plants`, **Import EAR Project** is checked, and the Target Server is set to `WebSphere Application Server Version 6`.
6. Click **Finish**.
7. If prompted to switch to the J2EE Perspective, click **Yes**.
8. Select **File** → **Import**. Select **EAR file** for the import source and click **Next**.
9. Browse to the PlantsSupplier EAR that you exported from WebSphere Studio Application Developer.
10. Confirm that the EAR project is set to `PlantsSupplier`, **Import EAR Project** is checked and the Target Server is set to `WebSphere Application Server V6.0`.

11. Click **Finish**.



Building workspace: (17%)

Figure 4-3 Importing an EAR project

12. After the import completes, watch the lower right corner of IBM Rational Application Developer for WebSphere Software. Wait for all background tasks to complete. At this point, there are two errors (problems with dds.xml) and a significant number of warnings in the Problems view.

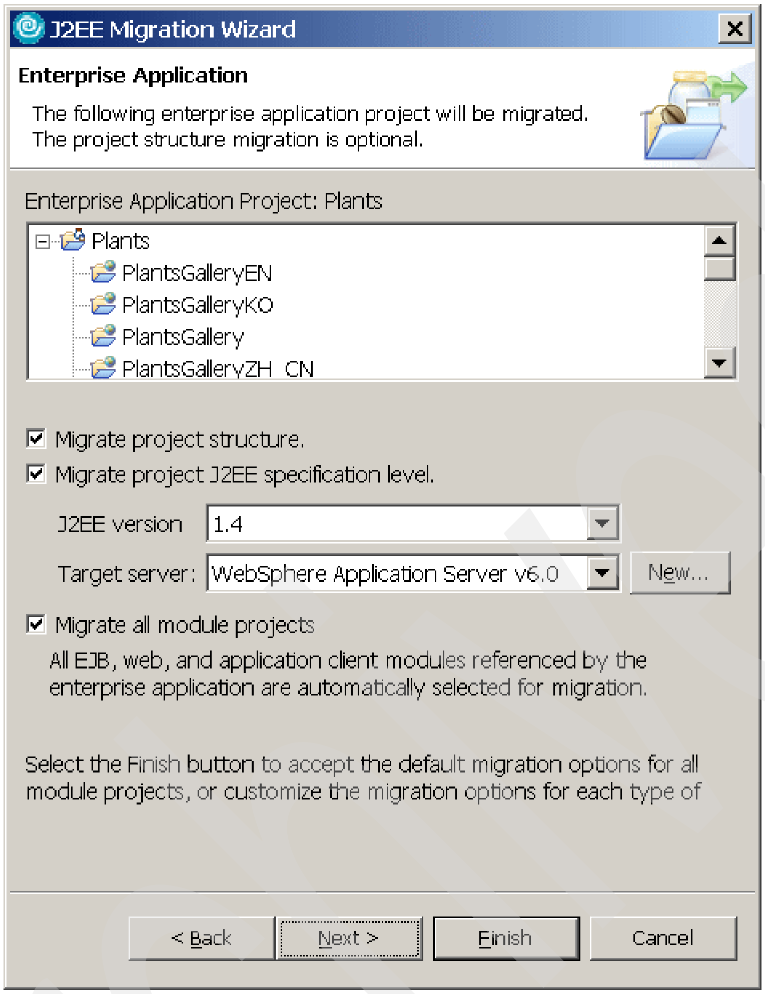
The code validators in IBM Rational Application Developer for WebSphere Software have been enhanced to provide warnings on many issues that were not in WebSphere Studio Application Developer. A close examination of the warnings shows that they address code maintenance issues, but do not represent problems with the code. They can safely be ignored.

The errors in dds.xml are due to name space issues in XML files. These errors can also be ignored.

Migrating the Enterprise Applications

1. In the Project Explorer view, expand **Enterprise Applications**.
2. Right-click **Plants** and select **Migrate** → **J2EE Migration Wizard**.

3. On the J2EE Migration Wizard Welcome Page click **Next**.



Enterprise Application

The following enterprise application project will be migrated.
The project structure migration is optional.

Enterprise Application Project: Plants

- Plants
  - PlantsGalleryEN
  - PlantsGalleryKO
  - PlantsGallery
  - PlantsGalleryZH_CN

☑ Migrate project structure.

☑ Migrate project J2EE specification level.

J2EE version: 1.4

Target server: WebSphere Application Server v6.0

☑ Migrate all module projects

All EJB, web, and application client modules referenced by the enterprise application are automatically selected for migration.

Select the Finish button to accept the default migration options for all module projects, or customize the migration options for each type of

Figure 4-4 J2EE Migration Wizard

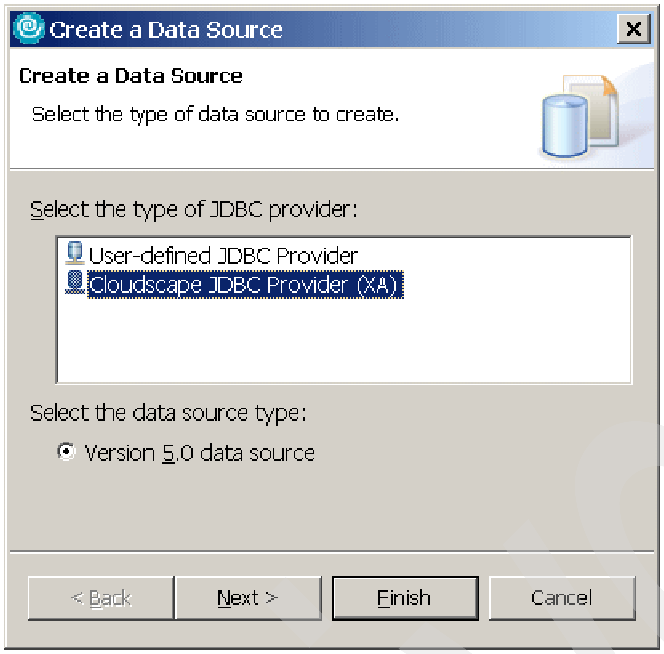
4. On the Enterprise Application page, click **Finish** to migrate the entire application.
5. When the **Migration Complete** dialog appears, click **OK**.
6. Use the same steps to migrate the PlantsSupplier application.

Adding the Data Source definition to the Plants application

It is possible to define the Data Source that is used by the Plants application directly on the server that you deploy to, as is done in WebSphere 5. By putting the Data Source definition in the extended application deployment descriptor instead, you ensure that the data source is created on any application server to which you deploy the application.

1. In the Project Explorer view, expand **Enterprise Applications** → **Plants** and double-click **Deployment Descriptor**.
2. Click the **Deployment** tab of the Deployment Descriptor editor.
3. Select **Cloudscape JDBC Provider(XA)** in the JDBC Provider list.

4. Click **Add** next to **Data source defined in the JDBC provider** to define the data source.



Create a Data Source

Select the type of data source to create.

Select the type of JDBC provider:

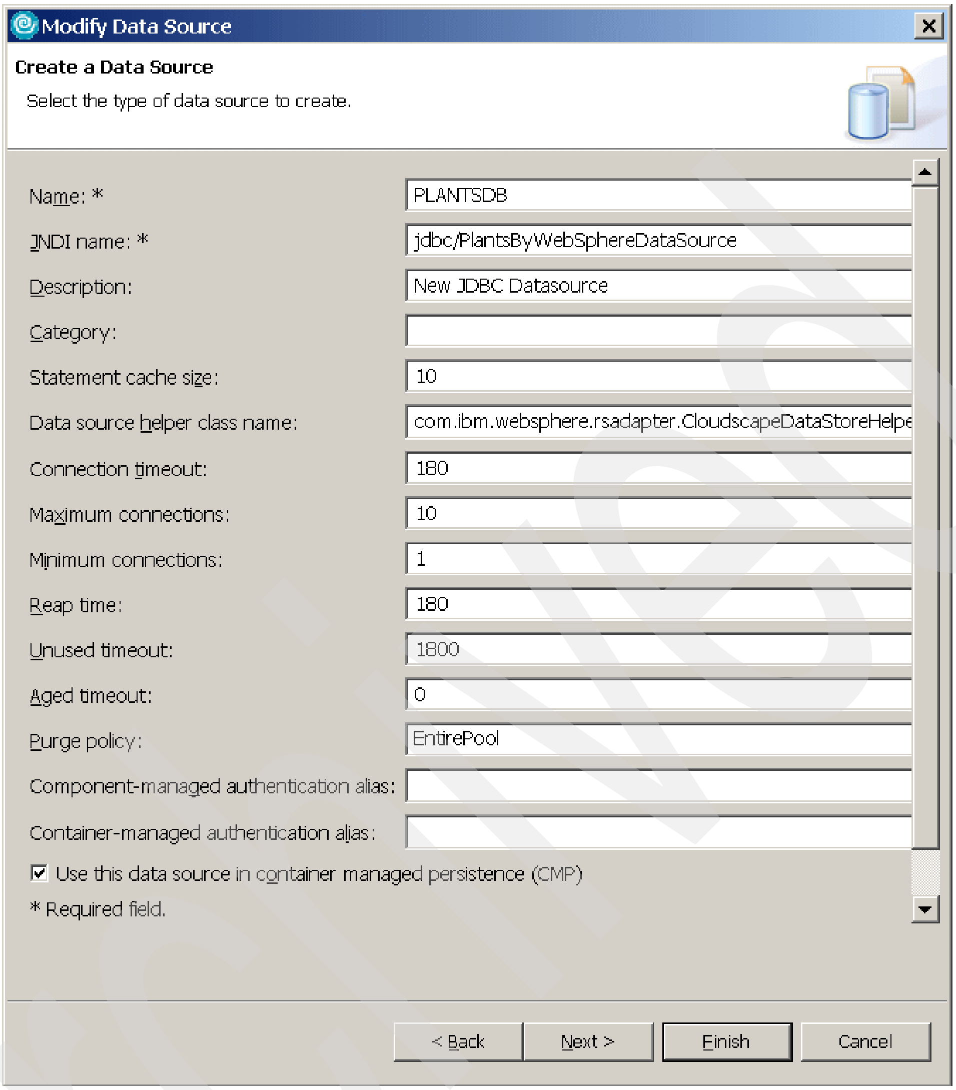
- User-defined JDBC Provider
- Cloudscape JDBC Provider (XA)

Select the data source type:

● Version 5.0 data source

Figure 4-5 Creating a data source

5. Select **Cloudscape JDBC Provider (XA)** and **Version 5.0 data source**. Click **Next**.



Modify Data Source

Create a Data Source

Select the type of data source to create.

Name: *	PLANTSDB
JNDI name: *	jdbc/PlantsByWebSphereDataSource
Description:	New JDBC Datasource
Category:	
Statement cache size:	10
Data source helper class name:	com.ibm.websphere.rsadapter.CloudscapeDataStoreHelpe
Connection timeout:	180
Maximum connections:	10
Minimum connections:	1
Reap time:	180
Unused timeout:	1800
Aged timeout:	0
Purge policy:	EntirePool
Component-managed authentication alias:	
Container-managed authentication alias:	

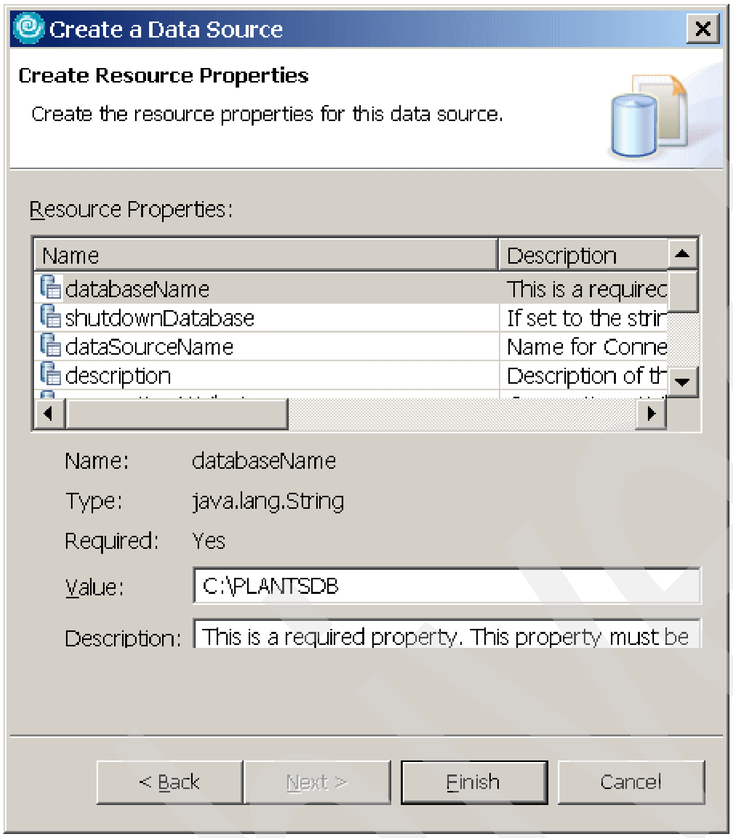
☑ Use this data source in container managed persistence (CMP)

* Required field.

Figure 4-6 Data source attributes

6. Type the Name: `PLANTSDB`.
7. Type the JDBC name: `jdbc/PlantsByWebSphereDataSource`.
8. Ensure that the Data source helper class name is `com.ibm.websphere.rsadapter.CloudscapeDataStoreHelper`.
9. Ensure that **Use this data source in container managed persistence** is checked.

10. Click **Next**.



Create Resource Properties

Create the resource properties for this data source.

Resource Properties:

Name	Description
databaseName	This is a requirec
shutdownDatabase	If set to the strir
dataSourceName	Name for Conne
description	Description of th

Name: databaseName

Type: java.lang.String

Required: Yes

Value: C:\PLANTSDB

Description: This is a required property. This property must be

Figure 4-7 Data source custom properties

11. Type the appropriate value for the databaseName. This should be the same value used to run the application under WebSphere Studio Application Developer. For Cloudscape, the databaseName is the name of a directory that contains the Cloudscape database. If the PLANTSDB is not present on the IBM Rational Application Developer for WebSphere Software system, it should be copied from the WebSphere Studio Application Developer system at this time.

12. Click **Finish**.

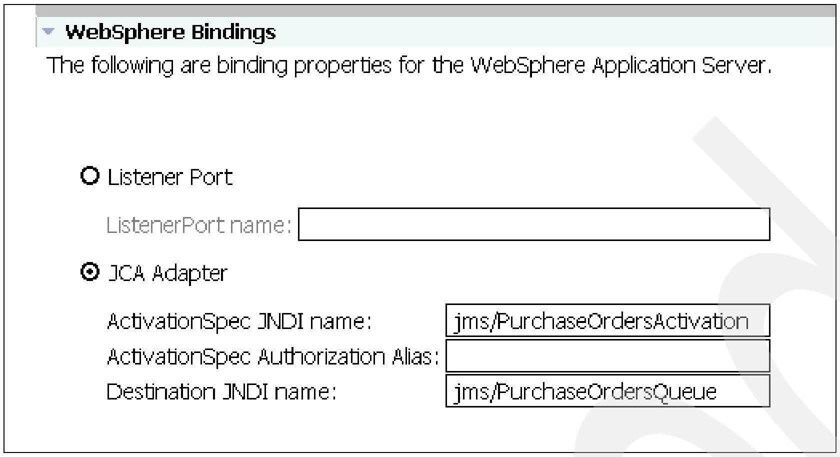
13. Close the Application Deployment Descriptor editor and save the deployment descriptor.

Updating the PurchaseOrders MDB to use a JCA Adapter

The PurchaseOrders MDB was configured in WebSphere Studio Application Developer to use a ListenerPort. You are using WebSphere 6 default messaging, so this must be changed to the new option for MDBs: JCA adapter.

1. In the Project Explorer View, expand **EJB Projects** → **SupplierEJB** and double-click **Deployment Descriptor**.

2. Click the **Beans** tab of the EJB Deployment Descriptor editor and select the **PurchaseOrders MDB EJB.**



WebSphere Bindings

The following are binding properties for the WebSphere Application Server.

○ Listener Port

ListenerPort name:

● JCA Adapter

ActivationSpec JNDI name: jms/PurchaseOrdersActivation

ActivationSpec Authorization Alias:

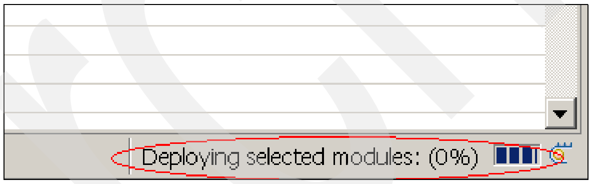
Destination JNDI name: jms/PurchaseOrdersQueue

Figure 4-8 EJB bindings

3. Under WebSphere Bindings, click the **JCA Adapter** radio button.
4. For ActivationSpec JNDI name, type `jms/PurchaseOrdersActivation`.
5. For Destination JNDI name, type `jms/PurchaseOrdersQueue`.
6. Close the EJB Deployment Descriptor editor and save the deployment descriptor.

Generating deploy code for EJB applications

1. In the Project Explorer view, expand **EJB Projects** and select both **PlantsByWebSphereEJB** and **SupplierEJB**.
2. Right-click the selected projects and select **Deploy**. A progress dialog appears and indicates when the deployment is complete.



Deploying selected modules: (0%)

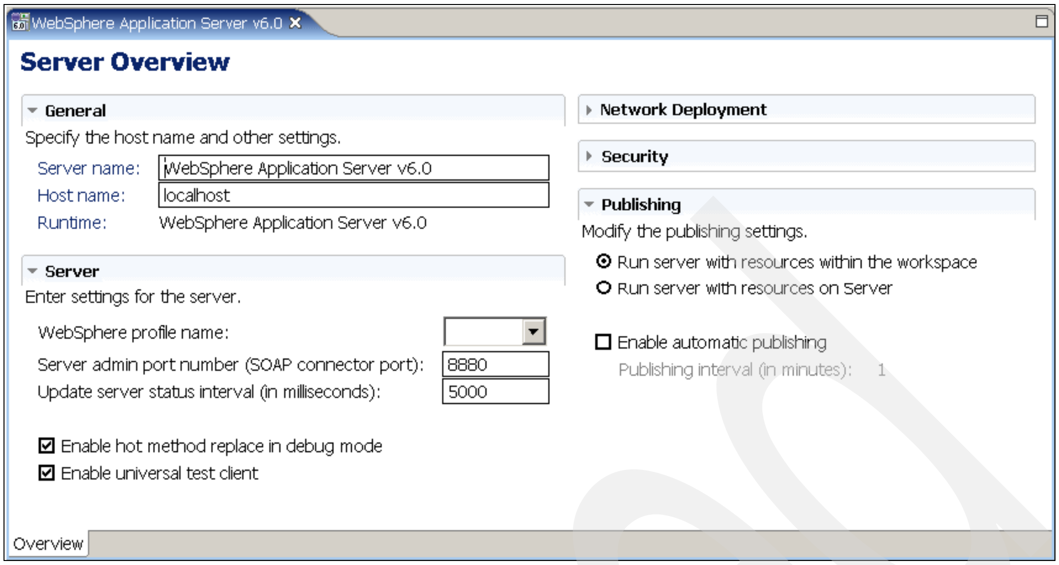
Figure 4-9 Generating deploy code for EJB

3. If you click **Run in Background**, you can monitor the progress by watching the lower right corner of the workbench.

Starting the test server

The test environment in IBM Rational Application Developer for WebSphere Software is significantly different from the test environment in WebSphere Studio Application Developer 5. Most configuration is done using the standard WebSphere administrative console, and most actions (such as adding and removing applications) are performed with the server running.

1. Click the **Servers** view.
2. Double-click **WebSphere Application Server 6.0** to launch the server configuration editor.



WebSphere Application Server v6.0

Server Overview

General

Specify the host name and other settings.

Server name: WebSphere Application Server v6.0

Host name: localhost

Runtime: WebSphere Application Server v6.0

Server

Enter settings for the server.

WebSphere profile name:

Server admin port number (SOAP connector port): 8880

Update server status interval (in milliseconds): 5000

☑ Enable hot method replace in debug mode

☑ Enable universal test client

Network Deployment

Security

Publishing

Modify the publishing settings.

● Run server with resources within the workspace

○ Run server with resources on Server

□ Enable automatic publishing

Publishing interval (in minutes): 1

Figure 4-10 Server configuration editor

Notice that the editor has been simplified. Most of the configuration that was managed by the WebSphere Studio Application Developer 5 Server Editor is managed in the administrative console.

3. Close the WebSphere Application Server editor.
4. In the Servers view, right-click **WebSphere Application Server 6.0** and select **Start**.
5. When the server status changes to `Started`, right-click the server and select **Run administrative console**. This launches the standard WebSphere 6.0 administrative console.
6. Type `migration` for the User ID (the User ID is only used for tracking changes to the configuration.) and click **Log in**.

Configuring the Mail Provider

This task is *optional* and is virtually identical to the Mail Provider configuration in WebSphere 5.

1. Open Resources in the left panel of the administration console. Select **Mail Providers**.
2. Ensure that a scope of **Node** is selected.
3. Click the **Built-in Mail Provider** provider to modify it.
4. In the Additional Properties section, click **Mail Sessions**.
5. Click **New**. Enter a Name of `PlantsByWebSphere Mail Session`. Also enter a JNDI name of `mail/PlantsByWebSphere`.
6. Configure the mail provider for outgoing mail by entering a Mail transport host and a Mail from address.
7. To save the mail provider settings, click **OK**.
8. Click the **Save** link (found at the top of the administrative console). You will see a window prompting you to save to the master configuration. Click the **Save** button on this page to save the settings to disk.

Configuring the JMS resources

Because this example uses WebSphere 6.0 default messaging, this configuration is significantly different from the configuration required for WebSphere Application Server Version 5.

1. Expand **Service integration** in the left panel of the administration console. Click **Buses**.

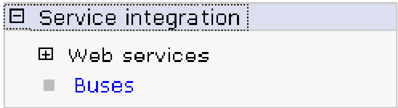
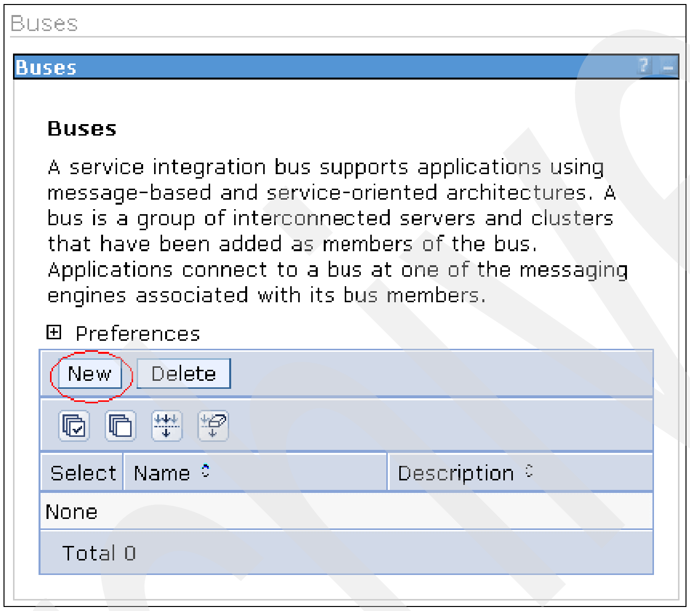


Figure 4-11 Selecting service integration bus

2. Under **Buses**, click **New.**



Buses

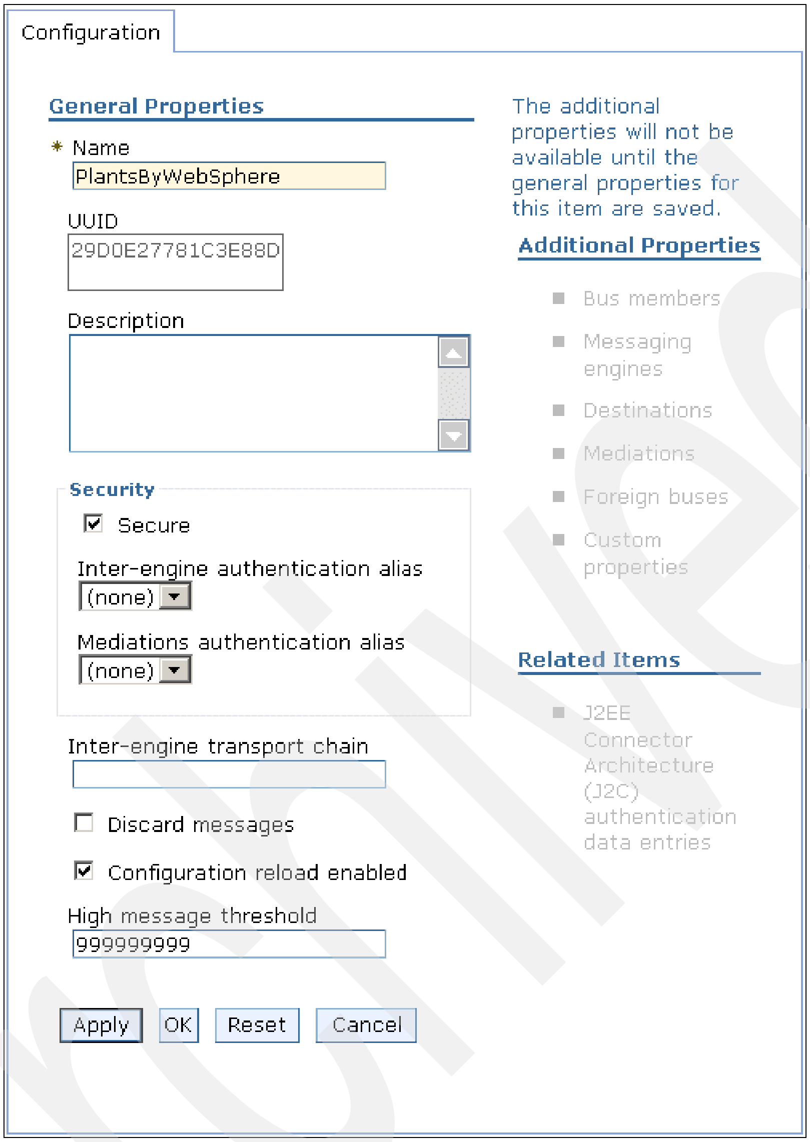
A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

Select	Name	Description
None		
Total 0		

Figure 4-12 Creating new service integration bus

3. Enter the name `PlantsByWebSphere` and click **Apply**.

.



Configuration

General Properties

* Name
PlantsByWebSphere

UUID
29D0E27781C3E88D

Description

Security

☑ Secure

Inter-engine authentication alias
(none)

Mediations authentication alias
(none)

Inter-engine transport chain

☐ Discard messages

☑ Configuration reload enabled

High message threshold
999999999

The additional properties will not be available until the general properties for this item are saved.

Additional Properties

- Bus members
- Messaging engines
- Destinations
- Mediations
- Foreign buses
- Custom properties

Related Items

- J2EE Connector Architecture (J2C) authentication data entries

Figure 4-13 Specifying bus properties

4. Under Additional Properties for the PlantsByWebSphere bus, click **Bus Members**.

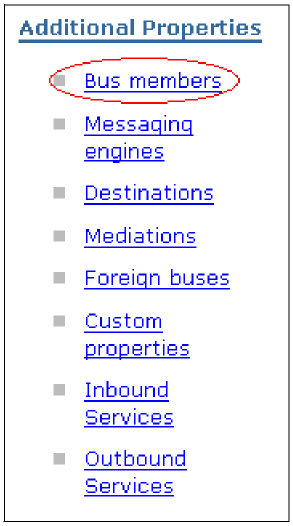
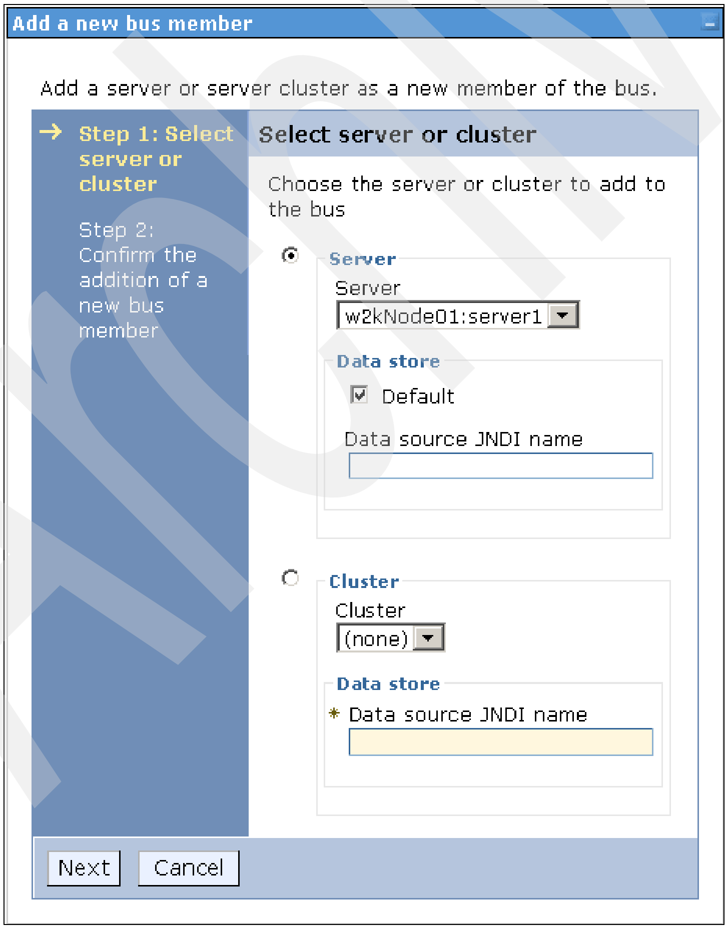


Figure 4-14 Specifying new bus members

5. On the Bus Members page, click **Add**.
6. Leave the default values on the **Select server or cluster** pane and click **Next.**


Add a new bus member
Step 1: Select server or cluster
Step 2: Confirm the addition of a new bus member
Next
Cancel

Add a server or server cluster as a new member of the bus.

Select server or cluster

Choose the server or cluster to add to the bus

● Server

Server: w2kNode01:server1

Data store

☑ Default

Data source JNDI name:

○ Cluster

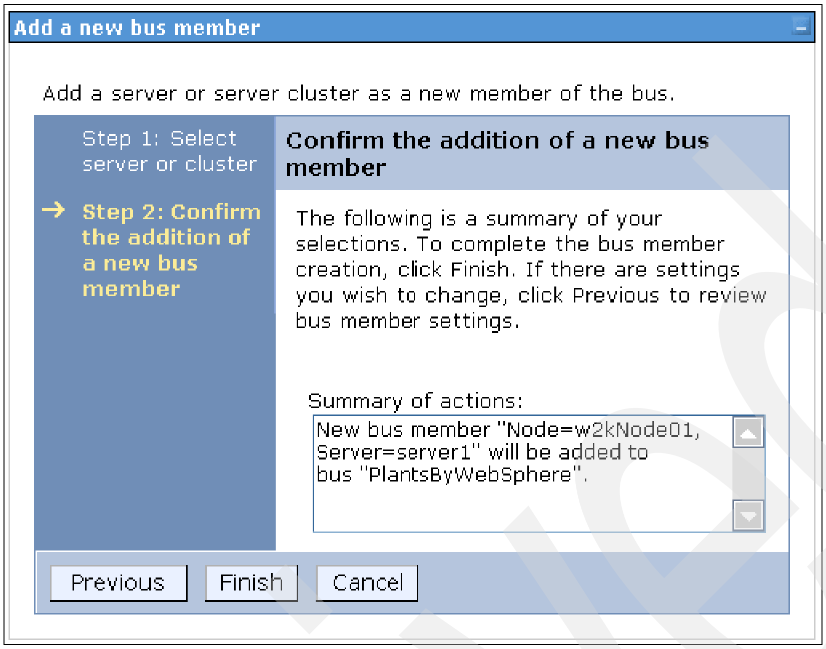
Cluster: (none)

Data store

* Data source JNDI name:

Figure 4-15 Selecting type of bus member

7. On the confirmation page, click **Finish**.



Add a server or server cluster as a new member of the bus.

Step 1: Select server or cluster
→ Step 2: Confirm the addition of a new bus member

Confirm the addition of a new bus member

The following is a summary of your selections. To complete the bus member creation, click Finish. If there are settings you wish to change, click Previous to review bus member settings.

Summary of actions:

New bus member "Node=w2kNode01, Server=server1" will be added to bus "PlantsByWebSphere".

Figure 4-16 Confirmation of bus member creation

8. Click **Buses** and then **PlantsByWebSphere** to return to the bus definition.
9. Under Additional Properties, click **Destinations**.

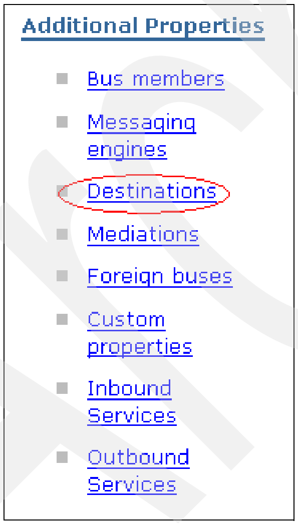
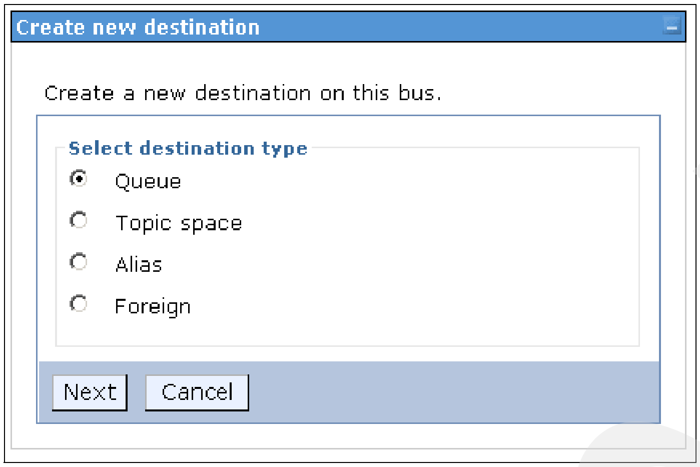


Figure 4-17 Creating bus destinations

10. There are some destinations already defined. Click **New** to create a new JMS destination.

11. Select **Queue** for the destination type and click **Next**.



Create new destination

Create a new destination on this bus.

Select destination type

● Queue

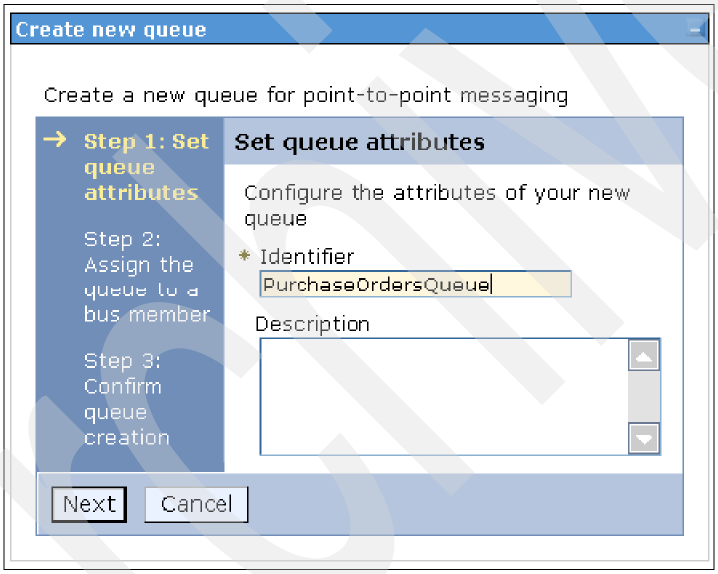
○ Topic space

○ Alias

○ Foreign

Figure 4-18 Creating new bus destination

12. Enter `PurchaseOrdersQueue` for the Identifier and click **Next.**



Create new queue

Create a new queue for point-to-point messaging

→ Step 1: Set queue attributes

Step 2: Assign the queue to a bus member

Step 3: Confirm queue creation

Set queue attributes

Configure the attributes of your new queue

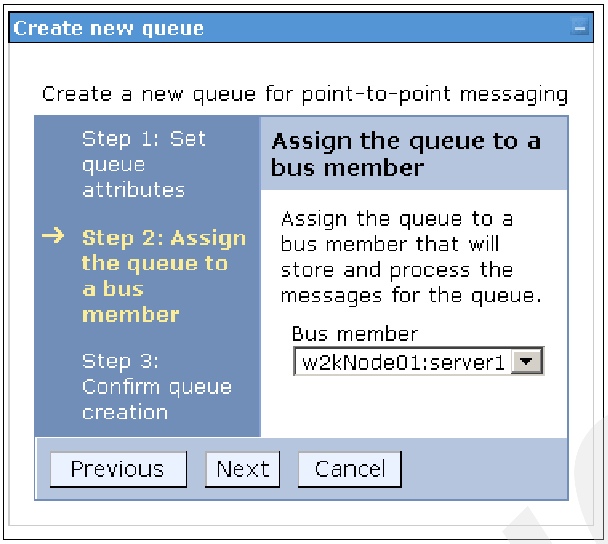
* Identifier

PurchaseOrdersQueue

Description

Figure 4-19 Specifying queue attributes

13..Leave the Bus member set to the member you defined and click **Next**



Create new queue

Create a new queue for point-to-point messaging

Step 1: Set queue attributes
→ Step 2: Assign the queue to a bus member
Step 3: Confirm queue creation

Assign the queue to a bus member

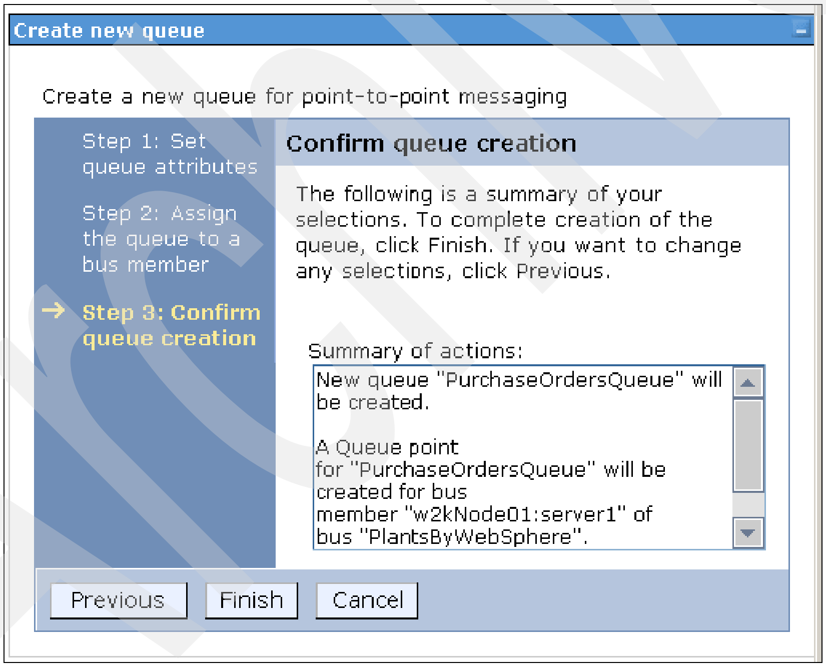
Assign the queue to a bus member that will store and process the messages for the queue.

Bus member: w2kNode01:server1

Figure 4-20 Creating a new bus member queue

14. On the confirmation page, click **Finish**.

.



Create new queue

Create a new queue for point-to-point messaging

Step 1: Set queue attributes
Step 2: Assign the queue to a bus member
→ Step 3: Confirm queue creation

Confirm queue creation

The following is a summary of your selections. To complete creation of the queue, click Finish. If you want to change any selections, click Previous.

Summary of actions:

New queue "PurchaseOrdersQueue" will be created.

A Queue point for "PurchaseOrdersQueue" will be created for bus member "w2kNode01:server1" of bus "PlantsByWebSphere".

Figure 4-21 Confirming queue creation

15. Click the **Save** link (found at the top of the administrative console). You will see a window prompting you to save to the master configuration. Click the **Save** button on this page to save the settings to disk

16. Expand **Resources** in the left panel of the administration console. Expand **JMS Providers** and select **Default messaging**.

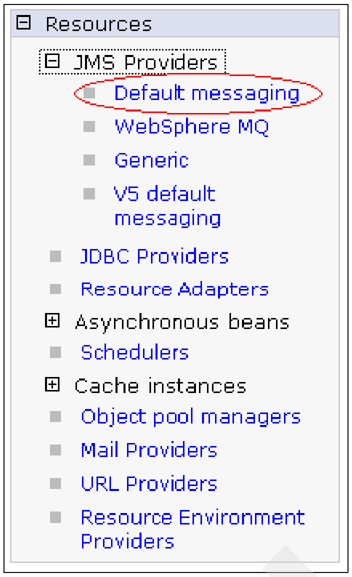


Figure 4-22 Selecting default messaging resources

17. Ensure that a scope of **Node** is selected.

18. Under Connection Factories, click **JMS queue connection factory**.

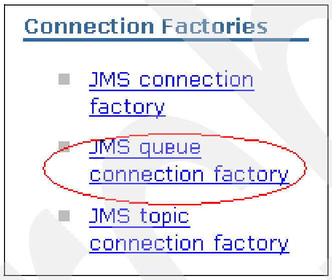
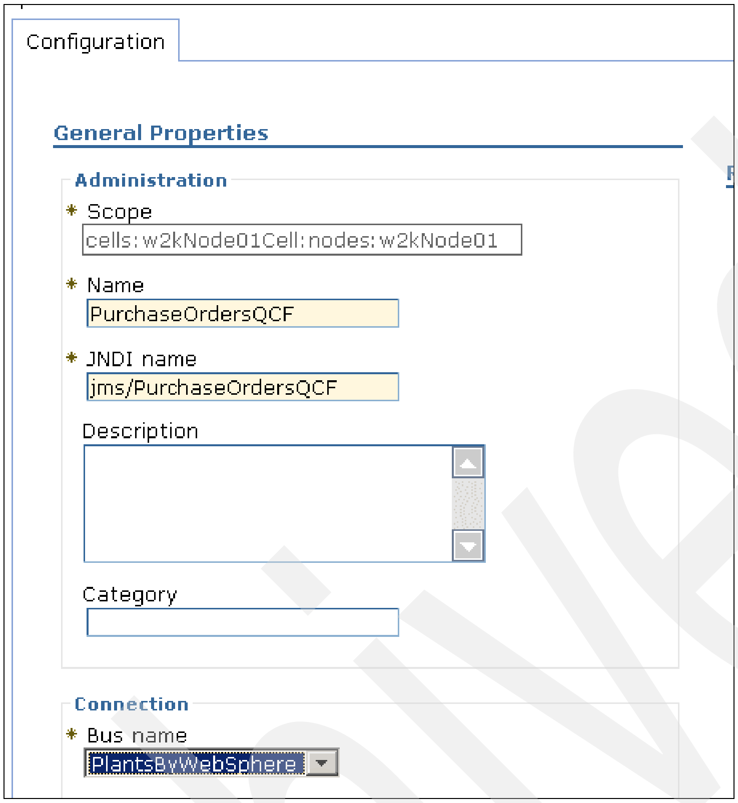


Figure 4-23 Selecting JMS queue connection factory

19. Click **New** to define a new queue connection factory.

20.Enter the name `PurchaseOrdersQCF` and the JNDI name: `jms/PurchaseOrdersQCF`. Under Bus name, select **PlantsByWebSphere**.



Configuration

General Properties

Administration

* Scope: cells:w2kNode01Cell:nodes:w2kNode01

* Name: PurchaseOrdersQCF

* JNDI name: jms/PurchaseOrdersQCF

Description:

Category:

Connection

* Bus name: PlantsByWebSphere

Figure 4-24 Queue connection factory properties

21.Click **OK** to create the QueueConnectionFactory.

22.Click **Default messaging provider** to return to the provider page.

23.Under Destinations, click **JMS queue.**

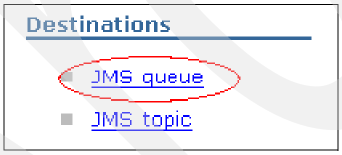
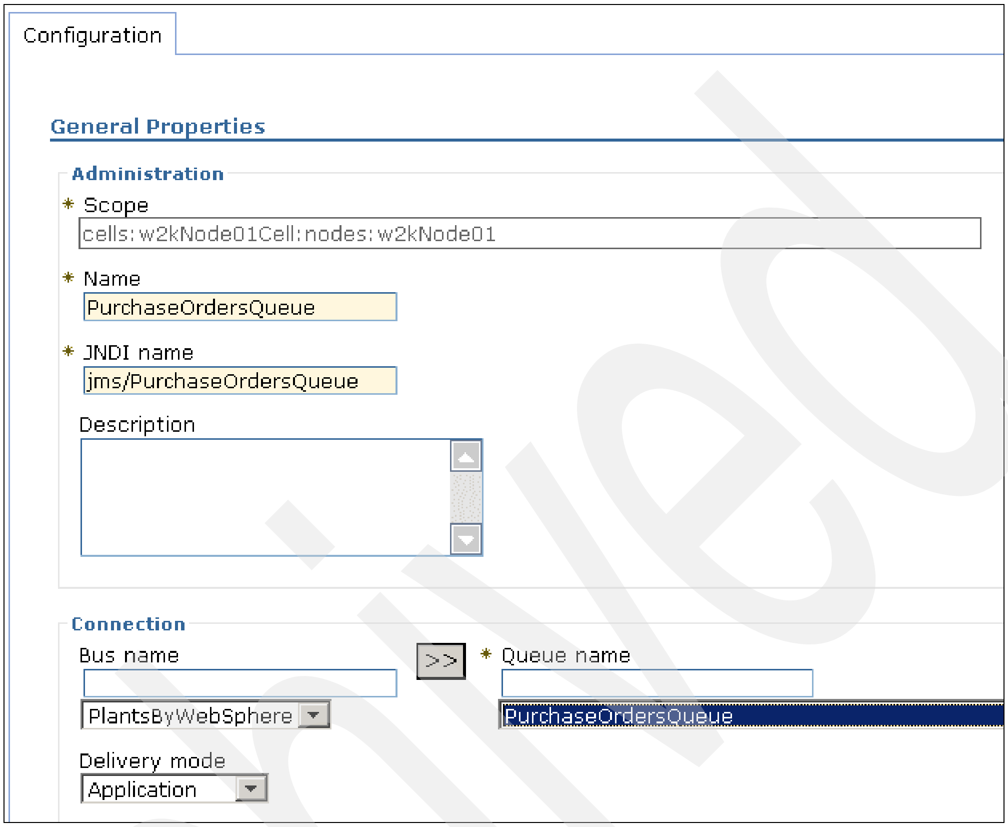


Figure 4-25 Selecting JMS queue

24..Click **New** to create a new JMS queue definition.

25. Type `PurchaseOrdersQueue` for the Name and `jms/PurchaseOrdersQueue` for the JNDI name. Select **PlantsByWebSphere** for the Bus name and click the **>>** button. Select **PurchaseOrdersQueue** for the Queue Name.



Configuration

General Properties

Administration

* Scope
cells:w2kNode01Cell:nodes:w2kNode01

* Name
PurchaseOrdersQueue

* JNDI name
jms/PurchaseOrdersQueue

Description

Connection

Bus name
PlantsByWebSphere

* Queue name
PurchaseOrdersQueue

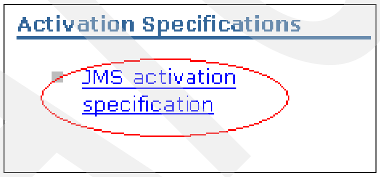
Delivery mode
Application

Figure 4-26 Specifying JMS queue properties

26. Click **OK**.

27. Click **Default messaging provider** to return to the provider page.

28. Under Activation Specifications, click **JMS Activation Specification**.



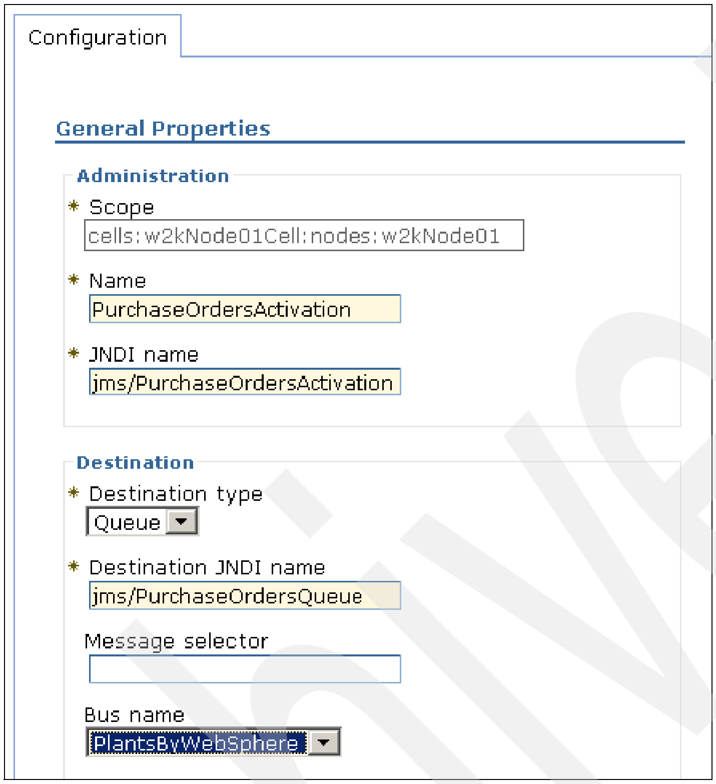
Activation Specifications

- JMS activation specification

Figure 4-27 Selecting JMS activation specification

29. Click **New** to create a new JMS activation specification.

30. Enter `PurchaseOrdersActivation` for the Name and `jms/PurchaseOrdersActivation` for the JNDI name. Make sure that the Destination type is set to `Queue`, and enter `jms/PurchaseOrdersQueue` for the destination JNDI name. For the Bus name, select **PlantsByWebSphere**. Click **OK**.



General Properties

Administration

* Scope
cells:w2kNode01Cell:nodes:w2kNode01

* Name
PurchaseOrdersActivation

* JNDI name
jms/PurchaseOrdersActivation

Destination

* Destination type
Queue

* Destination JNDI name
jms/PurchaseOrdersQueue

Message selector

Bus name
PlantsByWebSphere

Figure 4-28 Specifying JMS queue properties

31. Click the **Save** link (found at the top of the administrative console). You will see a window prompting you to save to the master configuration. Click the **Save** button on this page to save the settings to disk.

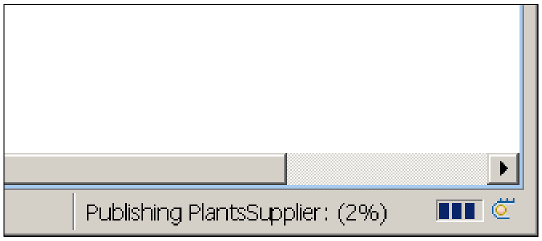
32. Click **Logout** at the top of the page to log out of the WebSphere Administrative Console.

33. Close the Admin Console window.

Deploying and testing the applications

1. In the Servers view, right-click **WebSphere Application Server V6** and select **Restart** → **Start.**
2. When the restart is complete, right-click the application server and select **Add and Remove Project.**
3. Click **Add All** to add both applications to the server and click **Finish.**

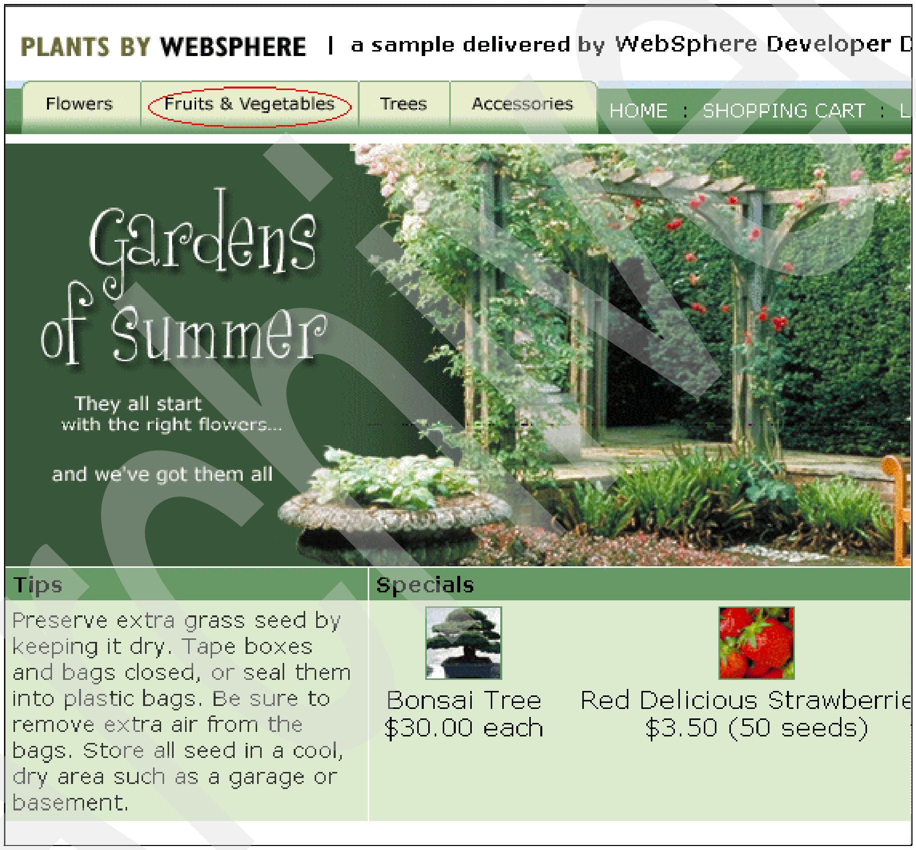
4. Watch the status in the lower right corner of IBM Rational Application Developer for WebSphere Software. When all tasks are complete, the applications are installed and running in the test environment.



Publishing PlantsSupplier: (2%)

Figure 4-29 Publishing to test server

5. Launch a Web browser and enter the URL: `http://localhost:9080/PlantsByWebSphere`.

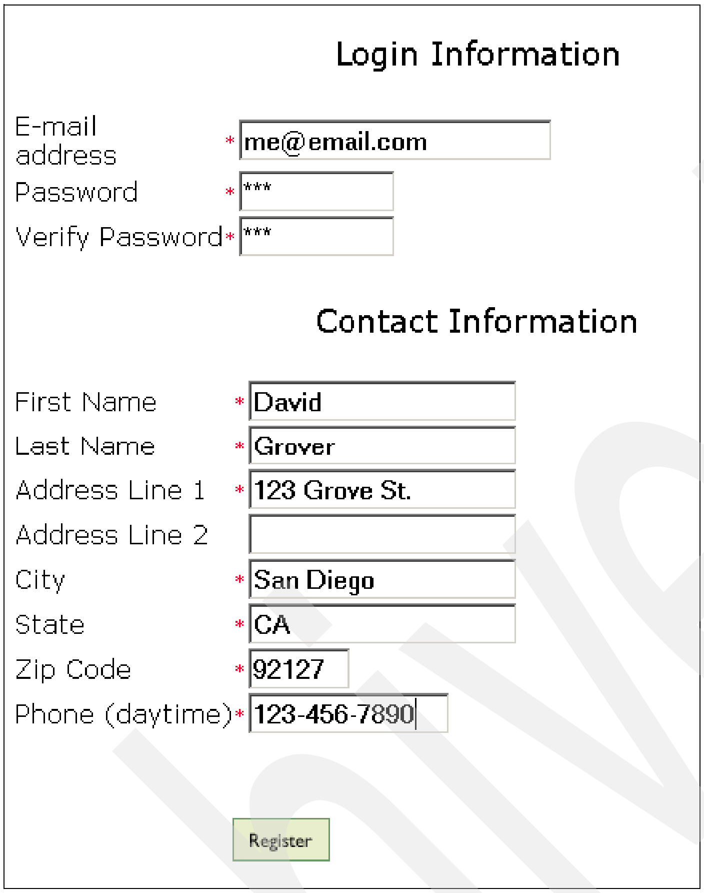


Preserve extra grass seed by keeping it dry. Tape boxes and bags closed, or seal them into plastic bags. Be sure to remove extra air from the bags. Store all seed in a cool, dry area such as a garage or basement.

Figure 4-30 Plants by WebSphere application initial Web page

6. Click **Fruits & Vegetables**. Click **Ornamental Gourd**. Enter `150` in the Quantity field and click **Add to cart**.
7. Click the **Checkout Now** button at the bottom of the page.
8. At the Login page, click the **register for your own account here** link.

9. Provide simulated login information and click **Register.**



Login Information

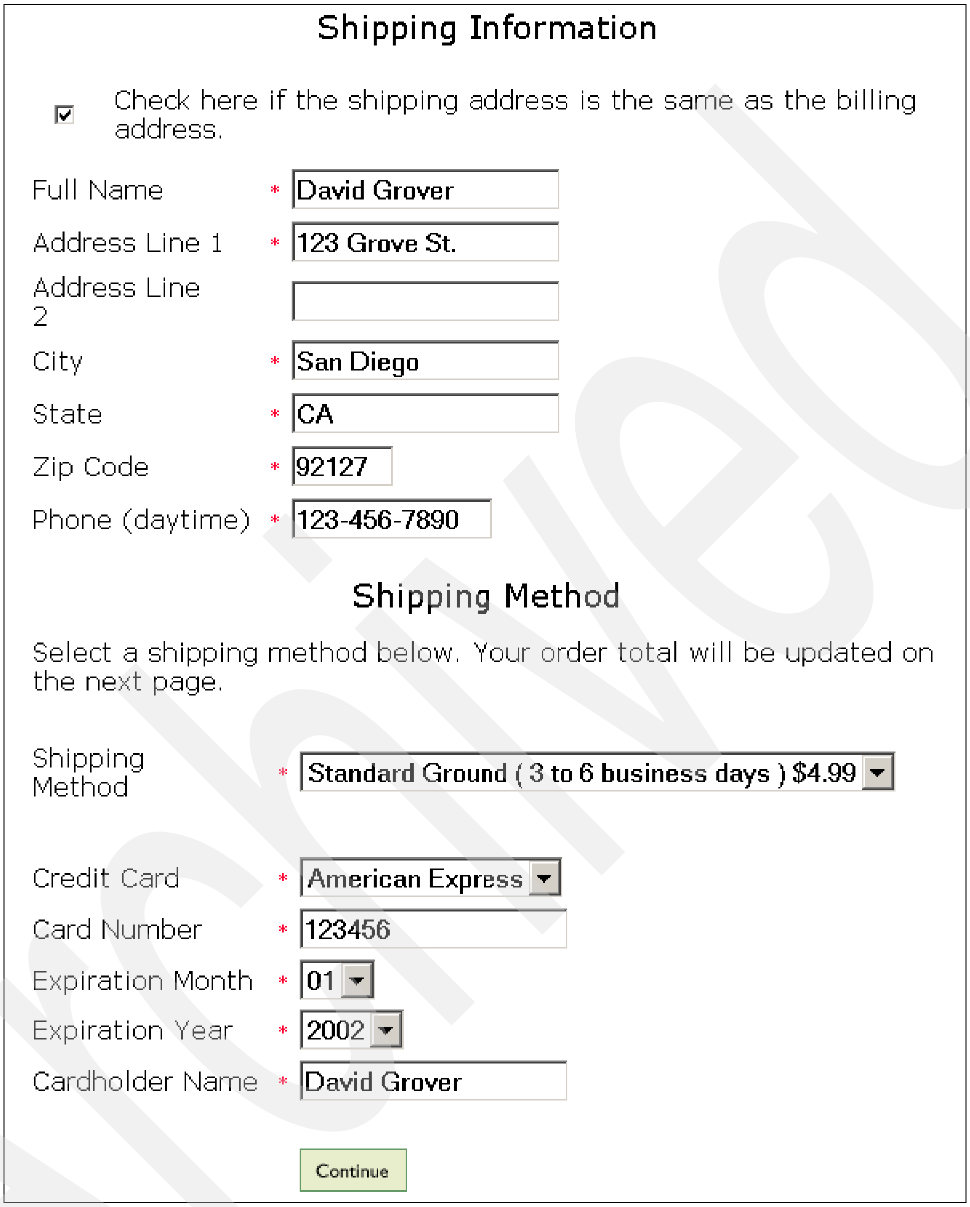
E-mail address *	me@email.com
Password *	***
Verify Password *	***

Contact Information

First Name *	David
Last Name *	Grover
Address Line 1 *	123 Grove St.
Address Line 2	
City *	San Diego
State *	CA
Zip Code *	92127
Phone (daytime) *	123-456-7890

Figure 4-31 Plants by WebSphere application login registration

10. Select the check box next to **Check here if the shipping address is the same as the billing address**. Enter a dummy value for the Credit Card Number (such as `123456`). and a Cardholder Name. Click the **Continue** button.



Shipping Information

☑ Check here if the shipping address is the same as the billing address.

Full Name	*	David Grover
Address Line 1	*	123 Grove St.
Address Line 2		
City	*	San Diego
State	*	CA
Zip Code	*	92127
Phone (daytime)	*	123-456-7890

Shipping Method

Select a shipping method below. Your order total will be updated on the next page.

Shipping Method	*	Standard Ground (3 to 6 business days) $4.99
Credit Card	*	American Express
Card Number	*	123456
Expiration Month	*	01
Expiration Year	*	2002
Cardholder Name	*	David Grover

Figure 4-32 Plants by WebSphere application shipping information

11. You are presented with a page where you can review your order. Click **Submit Order** at the bottom of the page.

12. The order is submitted. In the Console view, you may see an error related to sending e-mail. This occurs if you did not configure a mail provider. This error can be ignored.

13. A page is displayed confirming the order.

14. Visit the administration page to check for back orders:

`http://localhost:9080/PlantsByWebSphere/admin.html`

15. Click **Manage BackOrders**.

16. Under Back Order Items, you should see a shortage of ornamental gourds. To replenish the supply, enter `100` for the quantity to order and click the check box next to the ornamental gourds. Click the **Order Stock** button.
17. Scroll down to the Ordered Items. Observe that 100 gourds have been ordered. (If no gourds have been ordered, make sure that you selected the check box in the previous step.) Click the **Refresh** button on the top of the page (not the browser refresh button) to ensure that the items have been ordered.
18. After clicking the **Refresh** button, the gourds should appear under Received Items. This indicates that the items have been received from the supplier. To put the items back in stock, click the check box next to the item and click **Update Stock**.

    The ornamental gourds no longer appear in the received items, indicating that they have been integrated into the stock that people order from.
19. Testing of the application is complete. In the Servers view, right-click **WebSphere Application Server V6.0** and select **Stop**.

4.1.2 Example: Migrating PlantsByWebSphere using the CVS method

It is important to use a full-featured CVS server for this procedure. CVSNT in particular might not be robust enough to support the steps described.

This example is an alternative to the example shown in 4.1.1, “Example: Migrating Plants By WebSphere using the export/import method” on page 40. The result of this procedure should be the same as for that example.

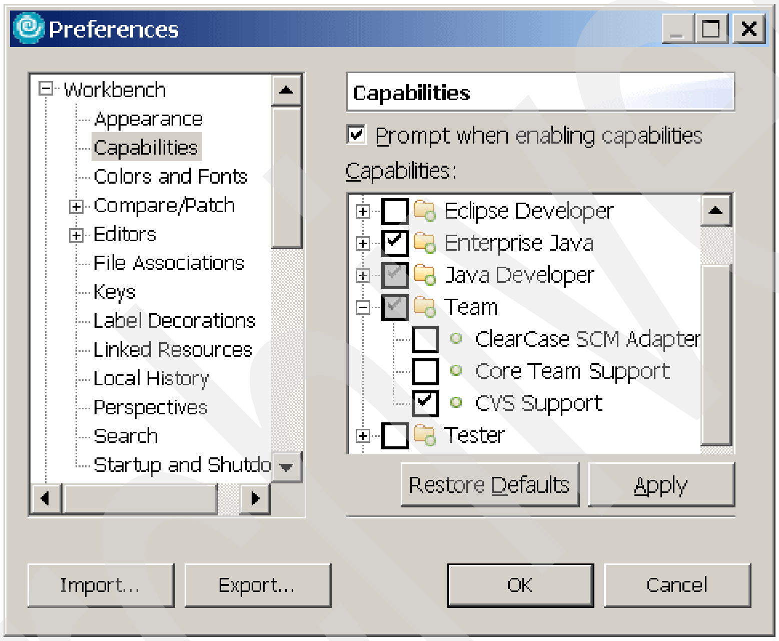
Adding the application into CVS

1. Start WebSphere Studio Application Developer and open the workspace containing the Plants by WebSphere applications.
2. Switch to the CVS Repository Exploring perspective.
3. Right-click the **CVS Repositories** view and select **New** → **Repository Location**.
4. Type in the appropriate Host, Repository path, user, and password for your CVS server.
5. Leave the Connection type as `pserver` and click **Finish**.
6. Switch back to the J2EE Perspective.
7. In the Project Navigator view, select the **Plants** project. Right-click and select **Team** → **Share Project**.
8. In the Share Project dialog, verify that the correct repository is selected and click **Finish**.
9. In the Synchronize view, right-click the **Plants** project and select **Commit**.
10. When prompted to add the resources, click **Yes**.
11. Type in a release comment and click **OK**.
12. Repeat the process to check each of the projects into CVS.
13. In the Project Navigator view, select all the projects.
14. Click **File** → **Export**. Select **Team Project Set** and click **Next**.
15. Type the File Name `C:\PlantsTeamSet` and click **Finish**.
16. Close WebSphere Studio Application Developer.

Importing the projects into IBM Rational Application Developer for WebSphere Software

1. Launch IBM Rational Application Developer for WebSphere Software on a new workspace.
2. Close the Welcome view to access the workbench.

   Before importing the projects, CVS must be configured to not prune empty directories. In order to configure this, the CVS capability must be enabled.
3. Click **Window** → **Preferences.**
4. Expand **Workbench** and click **Capabilities**. Under Capabilities, expand **Team** and check **CVS Support**.
5. Click **OK**.



Preferences

Capabilities

☑ Prompt when enabling capabilities

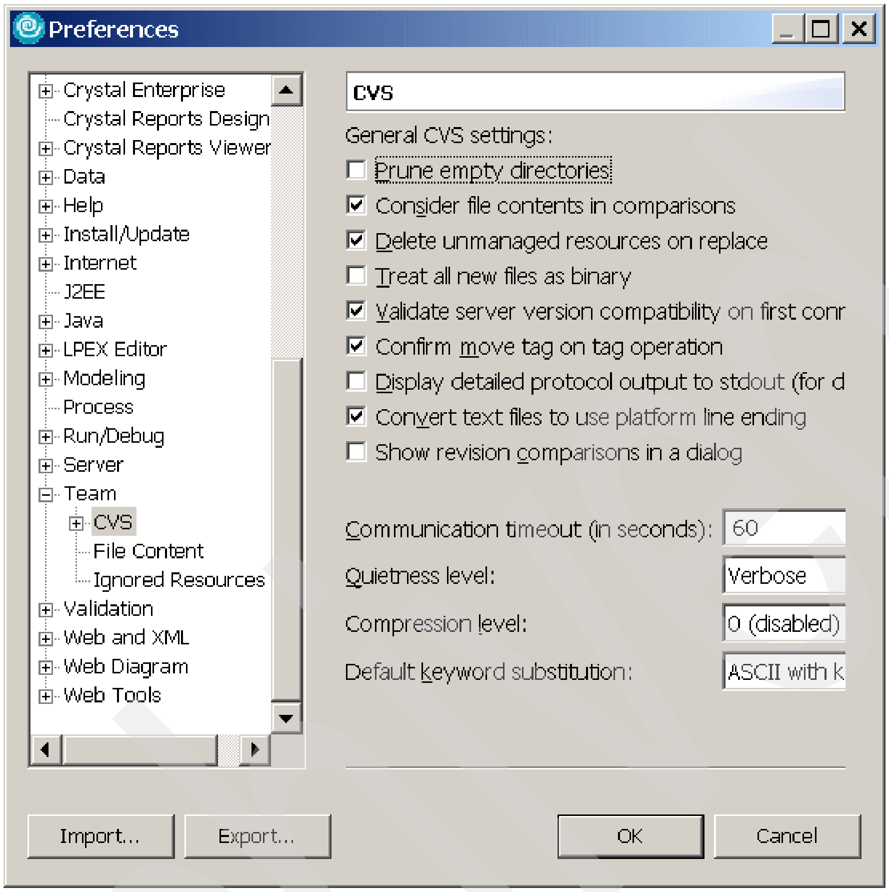
Capabilities:

- □ Eclipse Developer
- ☑ Enterprise Java
- ☑ Java Developer
- ☑ Team
  - □ ClearCase SCM Adapter
  - □ Core Team Support
  - ☑ CVS Support
- □ Tester

Figure 4-33 Enabling CVS support

6. Click **Window** → **Preferences.**

7. Expand **Team** and click **CVS**. Clear the check box next to **Prune empty directories**. Click **OK**.



CVS

General CVS settings:

□ Prune empty directories

☑ Consider file contents in comparisons

☑ Delete unmanaged resources on replace

□ Treat all new files as binary

☑ Validate server version compatibility on first conr

☑ Confirm move tag on tag operation

□ Display detailed protocol output to stdout (for d

☑ Convert text files to use platform line ending

□ Show revision comparisons in a dialog

Communication timeout (in seconds): 60

Quietness level: Verbose

Compression level: 0 (disabled)

Default keyword substitution: ASCII with k

Figure 4-34 CVS preferences

8. Click **File** → **Import**. Select **Team Project Set** and click **Next**.
9. Type the File name `C:\PlantsTeamSet.psf` and click **Finish**.
10. Watch the status in the lower right corner of the Workbench. Wait until all tasks are complete.

Fixing classpath problems

There is a problem with the classpath of one project that must be fixed before all projects can be built.

1. In the Project Explorer, expand **Dynamic Web projects** and select the **PlantsGallery** project.
2. Right-click the project and select **Properties**.
3. Click **Java Build Path** and select the **Libraries** tab.
4. Select **WAS_PLUGINDIR/lib/websphere.jar** and click **Remove**.
5. Click **OK**.

Removing WebSphere Studio Application Developer Version 5.1 compatibility

This step is optional. As the projects were imported, new files were generated to allow the projects to be updated either in WebSphere Studio Application Developer or IBM Rational Application Developer for WebSphere Software. This allows J2EE 1.3 projects to be modified in either environment. If the projects are going to be migrated to J2EE 1.4, or if there is no need to maintain compatibility with WebSphere Studio Application Developer, the compatibility can be removed.

1. In the Project Explorer, expand **Enterprise Applications**.
2. Right-click the **Plants** application and select **Remove Compatibility.**

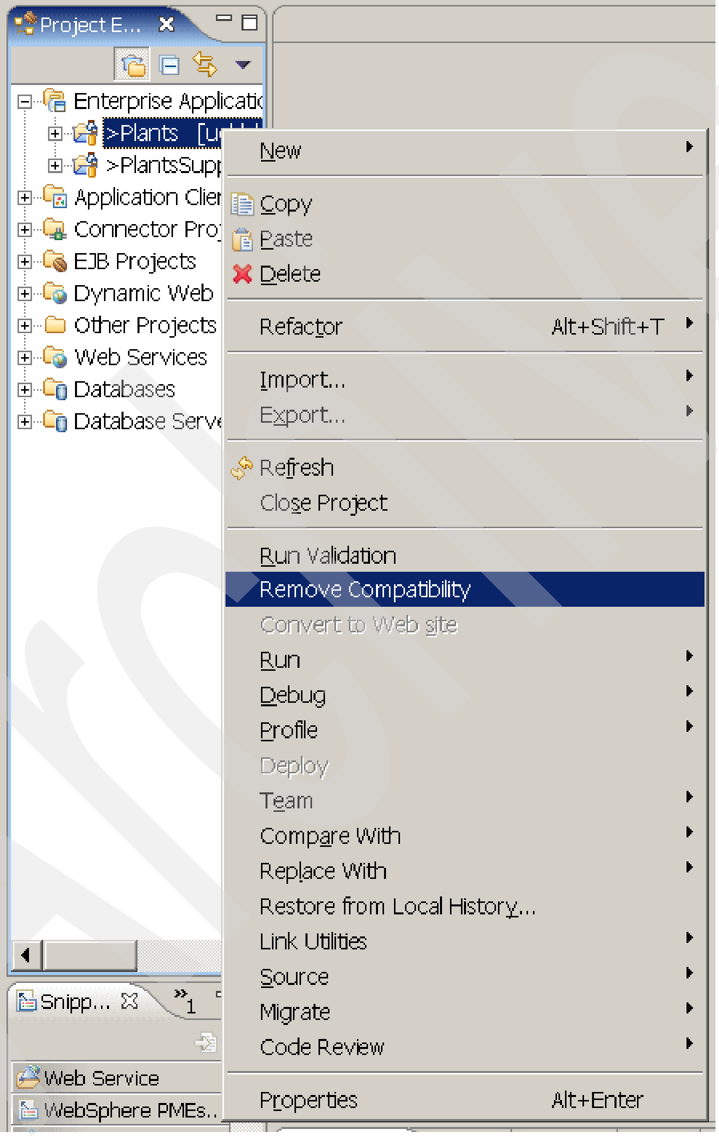
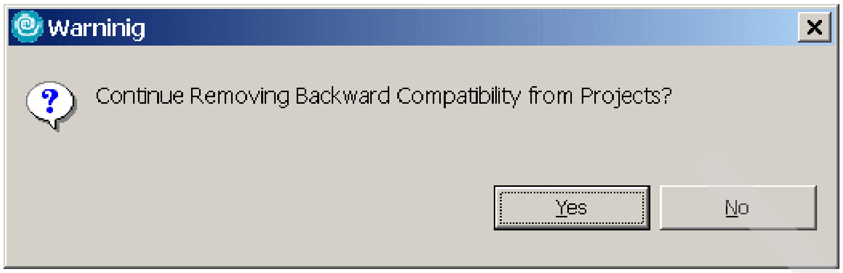


Figure 4-35 Removing workspace backward compatibility

3. When prompted to continue, click **Yes**.



Warninig

Continue Removing Backward Compatibility from Projects?

Figure 4-36 Confirming removing compatibility

4. Right-click the **PlantsSupplier** application and select **Remove Compatibility**.
5. When prompted to continue, click **Yes**.

Migration and testing can proceed as in the previous example.

5

Developer-related changes in V6.1

This chapter discusses application programming changes introduced by WebSphere Application Server V6.1.

5.1 J2SE 1.5

WebSphere Application Server V6.1 adopts J2SE 1.5, which introduces new Java language features that have compatibility implications. You will either have to change your code or compile using a compatibility flag. This is a summary of the compatibility issues:

- The enumeration language construct now forces the token enum to be a reserved word.
- New class names that contain the names Proxy and Queue may collide with your classes if you are not careful about fully qualifying your class names.

You can avoid changing your code by compiling with the flag:

```
-source 1.4
```

For more information about all the enhancements to J2SE 1.5, consult the article titled *J2SE 5.0 New Features and Enhancements* at this address:

http://java.sun.com/j2se/1.5.0/docs/relnotes/features.html

5.2 Application Server Toolkit

Application Server Toolkit (ASTK) has been a component of WebSphere Application Server since V5. In these earlier releases, ASTK has been strictly a deployment and testing tool. You would have had to code and compile your Java code in another program, such as IBM Rational Application Developer for WebSphere Software. Once you created a set of binaries, in the form of class files or jar files, you could then use ASTK to import the binary files and edit the deployment descriptors to create finished, deployable enterprise applications.

In V6.1, ASTK grows up to become a full-fledged development tool. With the V6.1 ASTK, it is now possible to create Java source code using embedded wizards. You also have full source code editing and debugging capabilities. However, it is not necessary to use ASTK if you already have other favorite development tools.

If you currently use IBM Rational Application Developer for WebSphere Software, you will feel right at home with ASTK. ASTK is based on Eclipse and therefore has the same look and feel as Rational Application Developer. Many of the development perspectives and views are the same between the two development tools. All of the Web application development tools come from the Web Tools Platform (WTP), which is a project under the Eclipse open source community. WTP was initially created through code donations from IBM. Many of the WTP tools originated in the WebSphere and Rational developer products. If you use WebSphere Studio Application Developer or Rational Application Developer, the Web development tools will look virtually identical.

We mention ASTK because it has become a convenient development tool, mainly because it is bundled with the WebSphere Application Server product. The ASTK license permits you to make unlimited copies and consequently have your development team use those copies concurrently.

Here are some of the new developer type things you can do with ASTK:

- Develop Web applications using editors and wizards that can create HTML, Javascript, style sheets, and other Web artifacts.
- Develop J2EE applications using editors and wizards that can create JSPs, servlets, and EJBs.

- Develop Web services applications using wizards that can transform your WSDL files into Web services code, a Web services explorer that helps you navigate UDDI registries, and editors to configure Web services deployment descriptors and WSDL.
- Develop Python administrative scripts.
- Create XML artifacts with XML editors and XML schema editors.

You can get a tour of ASTK by visiting the IBM Education Assistant site, V6.1 development tools section, and viewing the presentations and show-me demos at this address:

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/DevelopmentTools.html

5.3 Code incompatibilities and deprecations

V6.1 introduces a handful of API deprecations and a very short list of APIs that have been removed. These deprecations and removals are most likely to affect you in maintaining your applications. For a complete list see the article titled *Deprecated and removed features* at this address:

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

5.3.1 WebSphere API deprecations

Table 5-1 lists the API's that are deprecated in V6.1.

Table 5-1 WebSphere deprecated functions

Function	Release when deprecated
com.ibm.websphere.rsadapter.DataStoreHelperMetaData.setDatabaseDefaultIsolationLevel	6.1
com.ibm.websphere.sib.mediation.handler.SIMessageContextException com.ibm.websphere.sib.mediation.messagecontext.SIMediationBeanMessageContext	6.1
com.ibm.servlet.ClientList com.ibm.servlet.ClientListElement com.ibm.servlet.MLNotFoundException com.ibm.servlet.PageListServlet com.ibm.servlet.PageNotFoundException	6.1
com.ibm.webtools.runtime.AbstractStudioServlet com.ibm.webtools.runtime.BuildNumber com.ibm.webtools.runtime.NoDataException com.ibm.webtools.runtime.StudioPervasiveServlet com.ibm.webtools.runtime.TransactionFailureException com.ibm.webtools.runtime.WSUtilities	6.1
com.ibm.ws.security.util.LoginHelper	6.1

- The method setDatabaseDefaultIsolationLevel (int) in com.ibm.websphere.rsadapter.DataStoreHelperMetaData. is deprecated. It should be replaced with the method setDatabaseDefaultIsolationLevel (int, int).

- The class com.ibm.websphere.sib,.mediation.handler.SIMessageContextException is deprecated. It should be replaced with com.ibm.websphere.sib.mediation.handler.MessageContextException.
- The interface com.ibm.websphere.sib.mediation.messagecontext.SIMediationBeanMessageContext is deprecated and has no replacement. Redesign your application to avoid use of this interface.
- The WebSphere PageListServlet classes are deprecated. Redesign your application to use javax.servlet.filter class functionality.
- The classes in com.ibm.webtools.runtime are deprecated. There is no replacement. Redesign your application to use standard J2EE classes.
- The LoginHelper CORBA authentication helper is deprecated and should be replaced with the Java Authentication and Authorization Service (JAAS). See the article titled *Migrating Common Object Request Broker Architecture programmatic login to JAAS* at this address:

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

5.3.2 WebSphere API removals

Table 5-2 shows the APIs that have been removed in V6.1.

Table 5-2 Removed APIs in V6.1

com.ibm.websphere.security.CustomRegistry interface
Secure Authentication Service (SAS) IIOP protocols
Secure Authentication Service (SAS) CORBA APIs
Common Connector Framework APIs

- Use of the com.ibm.websphere.security.CustomRegistry interface should be replaced by com.ibm.websphere.security.UserRegistry interface.
- Use of the Secure Authentication Service (SAS) IIOP protocols should be replaced by Common Interoperability Version 2 (CSIv2) protocols.
- Use of the Secure Authentication Service (SAS) CORBA APIs should be replaced by the APIs for Java Authentication and Authorization Service (JAAS).
- Use of the Common Connector Framework APIs should be replaced by J2EE Connector Architecture APIs.

Part 2

For system administrators

The chapters in this part are for system administrators who manage application server systems on the Windows and UNIX® architectures.

6

Product overview

This chapter provides a broad overview of WebSphere Application Server Version 6.

6.1 The new product: Version 6

The WebSphere software platform for e-business is based on a foundation that is formed from Web application serving and integration. IBM WebSphere Application Server Version 6 software provides the core software to deploy, integrate, and manage e-business applications.

WebSphere Application Server software supports custom-built applications based on integrated WebSphere software platform products or on other third-party products. Such applications can range from dynamic Web presentations to sophisticated transaction processing systems.

WebSphere Application Server Express, WebSphere Application Server, and WebSphere Application Server Network Deployment are incrementally related. As you scale up your e-business demand, you can also scale up your e-business capability by moving from one product to the next.

The Express product is the entry point. With the base application server product, you can host applications on more machines than Express. The Network Deployment product includes all of the application server functionality of the other two products. However, you can manage multiple machines and processes from a centralized Web application that is the administrative console of the deployment manager.

The WebSphere Application Server for z/OS product runs only on the IBM z/OS operating system platform. It is functionally equivalent to WebSphere Application Server Network Deployment.

6.1.1 Product offerings for V6

IBM WebSphere Application Server products provide a next-generation application server on an industry-standard foundation. Each product addresses a distinct set of scenarios and needs. WebSphere Application Server Version 6 includes the following product offerings:

- WebSphere Application Server Express
- WebSphere Application Server
- WebSphere Application Server Network Deployment
- WebSphere Application Server for z/OS

WebSphere Application Server Express

This offering addresses the basic programming and runtime needs of desktop developers and single-server production scenarios. The runtime environment addresses standards-based programming for the Web, component-based programming, and Web services. The administration model for this offering is a single-server environment without clustering, and without centralized administration of multiple server instances.

The basic license for the server component of the Express product includes the ability to install the server on a system with up to two CPUs. The package also includes a single-use license for the Rational Web Developer tool, which is a fully integrated development environment (IDE). Rational Web Developer has a single-use license, but can be installed on up to two different systems. This means that, even though you can install it on two systems, only one person may use any of those installations at any given time. Please read your license agreement for exact details of the restrictions on the license terms.

Installing the Express product offering on the same machine as the Rational Web Developer is not necessary. The IDE contains an exact replica of the V6 application server as a test environment.

WebSphere Application Server

This offering is similar to the Express product but does not have the two-machine license limitation. It addresses the basic programming and runtime needs of desktop developers and single-server production scenarios. The runtime environment addresses standards-based programming for the Web, component-based programming, and Web services.

The administration model is a single-server environment without clustering, and without centralized administration of multiple server instances. The development environment offering is an unlimited license of the Application Server Toolkit and a trial version of the IBM Rational Application Developer for WebSphere Software.

See this article for diagrams of topologies that support the WebSphere Application Server environment: *Planning to install WebSphere Application Server*, found at:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tins_scenario2.html

WebSphere Application Server Network Deployment

This offering addresses application servers that run in multiple-server production scenarios. The WebSphere Application Server Network Deployment product offering can create:

- Application server profiles

  An application server profile includes default applications and a default application server. The application server in the Network Deployment product can run as a managed node or as a stand-alone application server.

  The stand-alone application server is the same one found in the Express product and in the base WebSphere Application Server product with one important exception: you can add Network Deployment stand-alone application server nodes to a cell under the centralized management of the deployment manager.

- Deployment manager profiles

  The deployment manager profile includes the dmgr process. The deployment manager provides centralized administration of multiple application server nodes and custom nodes as a single cell. The deployment manager provides administration for basic clustering and caching support, including failover support and workload balancing.

- Custom profiles

  A custom profile is an empty node that you must federate. Use the deployment manager to customize the node by creating servers and clusters. The node does not include a default application server or default applications.

See this article for examples of common topologies that you can create with the WebSphere Application Server Network Deployment product: *Planning to install Network Deployment,* found at:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tins_scenario3.html

WebSphere Application Server for z/OS

WebSphere Application Server for z/OS is functionally equivalent to WebSphere Application Server Network Deployment. The product is administered differently due to the nature of the z/OS operating system. However, applications deployed to WebSphere Application Server for z/OS behave identically to those for the other products in the WebSphere Application Server product family.

6.2 New features for V6

IBM WebSphere Application Server products provide the following new and enhanced features and tools:

- New installation and operating model
- New programming model extensions
- New and improved developer tools bundled with V6 packages

6.2.1 New installation and operating model

V6 implements a new installation method compared to previous versions. Installation is divided into two major steps:

1. Installation of read-only executable files.
2. Creation of one or more sets of writable configuration files, each of which corresponds to a server instance.

The product executable files remain unchanged after installation until you install maintenance. All server processes that share the binaries use the updated level of the system files after installing the service.

See 8.2, "Installation" on page 100 for more details about how this new installation method is organized.

6.2.2 New programming model extensions

Many programming model extensions that were formerly part of the Enterprise (V5.0.x) product or the WebSphere Business Integration Server Foundation Version 5.1 product are now integrated into WebSphere Application Server Version 6 and WebSphere Application Server Network Deployment Version 6.

See 3.2.5, "Programming Model Extensions" on page 36 for more details about which extensions are packaged in which products.

See these articles for more information about programming model extensions:

- *Migrating WebSphere programming model extensions (PMEs)*

  `http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/rins_migratepme.html`

- *Learn about WebSphere programming extensions*

  `http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc6tech_appsvcs.html`

6.2.3 Development and assembly tools

Development and assembly tools are packaged in various combinations with the WebSphere Application Server packages. See 6.3, "Version 6 package descriptions" on page 79 for information about how these tools are packaged with the runtime products. These tools are discussed in more depth in Part 1, "For application developers" on page 1.

- IBM Rational Application Developer for WebSphere Software is intended for development of all types of J2EE applications.
- IBM Rational Web Developer for WebSphere Software is a simpler tool intended for development of Web applications that do not incorporate J2EE EJB functionality.

- WebSphere Application Server Toolkit is a limited purpose assembly tool intended for those who must repackage and redeploy existing applications. It is not a full function development environment.

6.3 Version 6 package descriptions

The following sections describe installable components within each product package:

- WebSphere Application Server Express
- WebSphere Application Server
- WebSphere Application Server Network Deployment

6.3.1 WebSphere Application Server Express

WebSphere Application Server Express contains these subproducts:

- Application Server Express V6.0
  These subcomponents can be installed:
  - Application server
  - IBM HTTP Server Version 6
  - Web server plug-ins
  - Application client environment
- IBM Rational Web Developer for WebSphere Software
- Application Server Toolkit (ASTK)
  ASTK has no restrictions on its license. You may make unlimited copies.
- DB2® UDB Database Express Edition V8.2
  DB2 UDB Database Express Edition has a limited development-only license and is not to be used for production purposes.
- Data Direct JDBC Drivers
  These are Java-based drivers supported on any of the application server platforms.
- Business Solutions package
  The Business Solutions package installs these integrated applications:
  - IBM Telephone Directory
  - IBM Welcome Page

Table 6-1 on page 80 shows which subcomponents are available and which operating system platform is supported for each. Note that ASTK is only supported in Linux and Windows operating systems for Version 6. In previous versions, ASTK was supported on all the platforms that supported the application server.

For more exact information about which operating system version and fix level is needed, see this document:

`http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27006921`

Table 6-1 Supported platforms for WebSphere Application Server Express

	Operating system platforms[1]						
Product and subcomponents	**AIX**	**HP-UX**	**Linux i p**	**Linux IA32**	**Linux z**	**Solaris**	**Windows**
Application Server Express V6.0							
Application server	X	X	X	X		X	X
IBM HTTP Server	X	X	X	X		X	X
Web server plug-ins	X	X	X	X		X	X
Application client environment	X	X	X	X		X	X
Rational Web Developer for WebSphere Software				X			X
Application Server Toolkit				X			X
DB2 UDB Database Express Edition V8.2				X			X
Data Direct JDBC Drivers	X	X	X	X		X	X
Business Solutions Applications	X	X	X	X		X	X

[1] Linux i p = Linux for iSeries™ and pSeries®
Linux z= Linux for zSeries
Linux IA32= Linux for Intel 32 bit architecture

6.3.2 WebSphere Application Server

The WebSphere Application Server contains these subproducts:

- Application Server V6.0
  These subcomponents can be installed:
  - Application Server
  - IBM HTTP Server V6.0
  - Web Server Plug-ins
  - Application client environment
- Data Direct JDBC Drivers
  These are Java-based drivers supported on any of the application server platforms.
- Application Server Toolkit (ASTK)
  ASTK has no restrictions on its license. You may make unlimited copies.
- DB2 UDB Database Express Edition V8.2
  DB2 UDB Database Express Edition has a limited development-only license and is not to be used for production purposes.
- Rational Application Developer Trial
  This is a limited license for the purposes of evaluating the software. The license expires 60 days after installation.

Table 6-2 shows which subcomponents are available and which operating system platform is supported for each. Note that WebSphere Application Server adds an additional platform for Linux on zSeries that is not supported for WebSphere Application Server Express.

For more exact information about which operating system version and fix level is needed, see this document:

http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27006921

Table 6-2 Supported platforms for WebSphere Application Server

Product and subcomponents	Operating system platforms[1]						
	AIX	**HP-UX**	**Linux i p**	**Linux IA32**	**Linux z**	**Solaris**	**Windows**
Application Server V 6.0							
Application server	X	X	X	X	X	X	X
IBM HTTP Server	X	X	X	X	X	X	X
Web server plug-ins	X	X	X	X	X	X	X
Application client environment	X	X	X	X	X	X	X
Rational Application Developer for WebSphere Software Trial version				X			X
Application Server Toolkit				X			X
DB2 UDB Database Express Edition V8.2				X			X
Data Direct JDBC Drivers	X	X	X	X	X	X	X

[1] Linux i p = Linux for iSeries and pSeries
Linux z= Linux for zSeries
Linux IA32= Linux for Intel 32 bit architecture

6.3.3 IBM WebSphere Application Server Network Deployment

These subcomponents can be installed:

- Application server
- IBM HTTP Server V6.0
- Web server plug-ins
- Application client environment
- Data Direct JDBC Drivers
  These are Java-based drivers supported on any of the application server platforms.
- Application Server Toolkit (ASTK)
  ASTK has no restrictions on its license. You may make unlimited copies.
- Edge Components
  Edge Components, formerly called WebSphere Edge Server, provide enhancements to load balancing and request caching above that offered by the application server.
- DB2 Universal Database™ for WebSphere Application Server V8.2
- IBM Tivoli® Directory Server 5.2
- IBM Tivoli Access Manager 5.1

- IBM Rational Application Developer for WebSphere Software Trial
This is a limited license for the purposes of evaluating the software. The license expires 60 days after installation.

Table 6-3 shows which subcomponents are available and which operating system platform is supported for each. For more exact information about which operating system version and fix level is needed, see this document:

http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27006921

Table 6-3 Supported platforms for WebSphere Application Server Network Deployment

	Operating system platforms[1]						
Product and subcomponents	**AIX**	**HP-UX**	**Linux i p**	**Linux IA32**	**Linux z**	**Solaris**	**Windows**
Application Server Network Deployment V 6.0							
Application server	X	X	X	X	X	X	X
IBM HTTP Server	X	X	X	X	X	X	X
Web server plug-ins	X	X	X	X	X	X	X
Application client environment	X	X	X	X	X	X	X
Rational Application Developer for WebSphere Software Trial version				X			X
Application Server Toolkit				X			X
DB2 Universal Database for WebSphere Application Server V8.2	X	X	X	X	X	X	X
Data Direct JDBC Drivers	X	X	X	X		X	X
Edge Components	X	X	X	X		X	X
IBM Tivoli Access Manager 5.1	X	X		X	X	X	X
IBM Tivoli Directory Server 5.2	X	X	X	X	X	X	X

[1] Linux i p = Linux for iSeries and pSeries
Linux z= Linux for zSeries
Linux IA32= Linux for Intel 32 bit architecture

6.3.4 IBM WebSphere Application Server for z/OS

The application server component is the only component of the product package that is installed on the z/OS system. All the other components are installed on alternate systems such as UNIX, Linux, or Windows.

These subcomponents can be installed:

- Application server
- IBM HTTP Server V6.0
- Web server plug-ins
- Application client environment

- Data Direct JDBC Drivers
  These are Java-based drivers supported on any of the application server platforms.
- Application Server Toolkit (ASTK)
  ASTK has no restrictions on its license. You may make unlimited copies.
- Edge Components
  Edge Components, formerly called WebSphere Edge Server, provide enhancements to load balancing and request caching above that offered by the application server.
- DB2 Universal Database for WebSphere Application Server V8.2
- IBM Tivoli Directory Server 5.2
- IBM Tivoli Access Manager 5.1
- IBM Rational Application Developer for WebSphere Software Trial
  This is a limited license for the purposes of evaluating the software. The license expires 60 days after installation.

Table 6-4 shows which subcomponents are available and which operating system platform is supported for each. For more exact information about which operating system version and fix level is needed, see this document:

http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27006921

Table 6-4 Supported platforms for WebSphere Application Server for z/OS

	Operating system platforms[1]							
Product and subcomponents	**AIX**	**HP-UX**	**Linux i p**	**Linux IA32**	**Linux z**	**Solaris**	**Windows**	**z/OS**
Application Server for z/OS V 6.0								
application server								X
IBM HTTP Server	X	X	X	X	X	X	X	
Web server plug-ins	X	X	X	X	X	X	X	
application client environment	X	X	X	X	X	X	X	
Rational Application Developer for WebSphere Software Trial version				X			X	
Application Server Toolkit				X			X	
DB2 Universal Database for WebSphere Application Server V8.2	X	X	X	X	X	X	X	
Data Direct JDBC Drivers	X	X	X	X		X	X	
Edge Components	X	X	X	X		X	X	
IBM Tivoli Access Manager 5.1	X	X		X	X	X	X	
IBM Tivoli Directory Server 5.2	X	X	X	X	X	X	X	

[1] Linux i p = Linux for iSeries and pSeries
Linux z= Linux for zSeries
Linux IA32= Linux for Intel 32 bit architecture

7

Runtime migration strategy

This chapter discusses the overall strategy of migration.

7.1 Environments

The ideal corporate WebSphere environment is shown in Figure 7-1. This environment, which is described with greater detail in *The Ideal WebSphere Development Environment*[1], represents (as established in the first sentence) the ideal; it is very likely that your environment contains many similar elements.

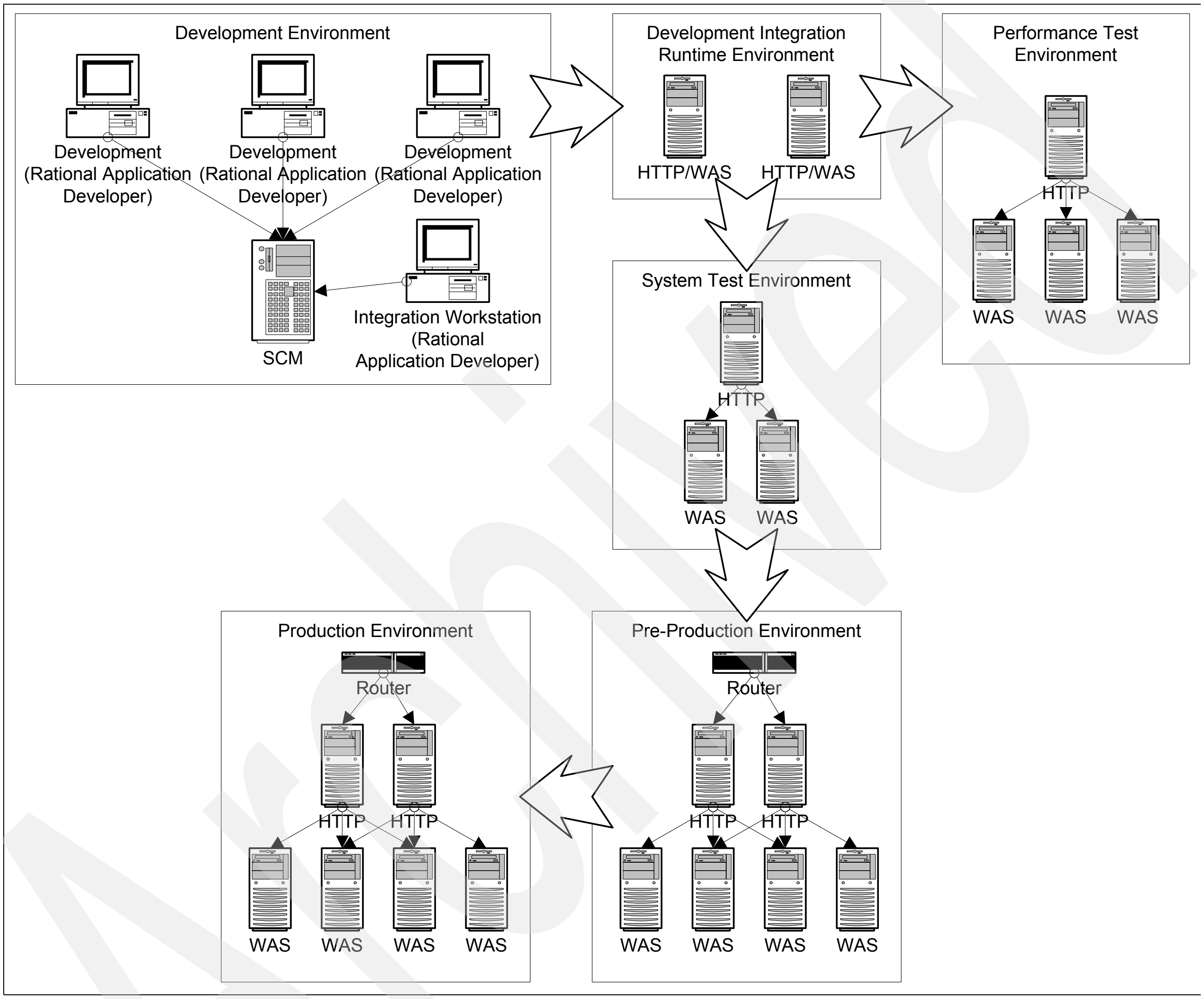

Development Environment
Development (Rational Application Developer)
Development (Rational Application Developer)
Development (Rational Application Developer)
SCM
Integration Workstation (Rational Application Developer)
Development Integration Runtime Environment
HTTP/WAS
HTTP/WAS
Performance Test Environment
HTTP
WAS
WAS
WAS
System Test Environment
HTTP
WAS
WAS
Production Environment
Router
HTTP
HTTP
WAS
WAS
WAS
WAS
Pre-Production Environment
Router
HTTP
HTTP
WAS
WAS
WAS
WAS

Figure 7-1 The ideal WebSphere environment

This figure represents an ideal environment, which includes all of the necessary stages for a first-class rigorous environment. In this environment, each sub-environment is as complete as it needs to be to serve the intended task. Because these environments all need to be migrated, it is important that you understand why each of these sub-environments exists and the constraints placed upon them so that you know what needs to be done during the migration.

Pragmatically speaking, very few organizations achieve the ideal described here. It is more realistic to expect that parts of the ideal are represented in your corporate environment.

Figure 7-2 on page 87 shows a scale of the risk associated with each of the environments.

[1] http://www-106.ibm.com/developerworks/websphere/techjournal/0312_beaton/beaton.html

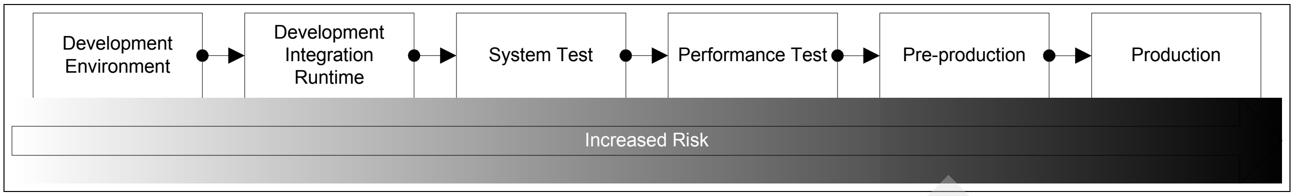


Figure 7-2 Increasing scale of risk

As you move from left to right:

- The risk to the business—should the environment suffer extended downtime—increases.
- The length of maintenance windows decreases.
- The amount of acceptable downtime decreases.
- The complexity of the environment increases.

Ideally, additional hardware is available for operations staff to use to familiarize themselves with the new version of WebSphere Application Server, but reality is that hardware is expensive and it may well be the case that additional hardware is not available. In this case, operations staff can gain familiarity with installation of WebSphere Application Server as they move left to right through these environments, starting, of course, with the development integration runtime.

There are at least three different types of runtime migration, each with increasing aversion to risk.

- The development test and system test environments
- Performance testing and pre-production environments
- Production runtime environment

These environments are discussed in detail in the following sections.

7.1.1 Development environment

The development environment is where developers work every day and this is where they need to be productive. Thus, they need the best tools and the fewest barriers to progress. This environment is composed of a number of development workstations (one for each developer), a source code management (SCM) tool, and an integration target system.

In general, the machines in a development environment do not run WebSphere Application Server outside of the WebSphere Test Environment embedded within IBM Rational Application Developer for WebSphere Software. Strictly speaking, from a “runtime migration” point-of-view, there is no inherent risk in migrating this environment.

More pragmatically, the cost of this environment not working is the cost of stopped development. A quick and orderly migration of this environment is critical and may require updates to related components, including the SCM, database drivers, databases, connectors, and so forth. Careful planning is key when migrating this (and every other) environment.

Timing of the migration of related resources introduces challenges, especially if it is difficult or impossible to run different versions of those related resources. You may find yourself in a situation where some developers have migrated their own personal workstations, but some have not. In that case, you may have to run two separate versions of a database for a period of time. This may require additional hardware in the meantime.

7.1.2 Development integration runtime

The development integration runtime environment is used by developers to test their applications on hardware and software that resembles the target production environment. Testing in this environment is concerned with uncovering issues related to subtle differences between the development and production systems as well as testing the deployment procedures. This may include such things as the use of various operating system services, WebSphere Application Server security, back-end systems, and others.

Developers use this environment to perform integration tests among all system components. This environment is also used to test installation and operational procedures, which are often operating system specific. The development integration runtime environment is configured to mirror the production environment at the smallest possible scale and complexity. In general, this environment does not include network devices such as load balancers, routers, or firewalls.

Systematic testing on this environment does not typically occur on a daily basis, but does occur regularly, perhaps bi-weekly as significant change is introduced into the application. This environment is controlled by the development team; it is used informally by developers and updated as often as necessary by developers while performing their tests. Periodically, this environment is refreshed using a formal build, deploy, and test procedure thereby removing any inconsistency and testing the full build and install procedures. In general, this environment does not include any development tools. As such, testing depends on the use of test scripts and tracing to determine correctness and identify problems.

This environment has perhaps the lowest amount of risk of all the environments presented here. If this environment is not available, developers cannot perform their integration tests on “production-like” hardware and software. This environment tends to be used sporadically by the development team and so it should be easy to schedule a few days of downtime for migration. It is also expected that this environment is rebuilt periodically anyway, so developers will be accustomed to the environment being unavailable.

This environment provides an opportunity for administrators to “cut their teeth” on WebSphere Application Server Version 6.0. Since this environment tends to be rebuilt periodically, a straight installation of the application server (rather than using the migration tools) is recommended. This gives the administrators an environment to learn more about the new version and experiment with configuration options. As part of the exercise, administrators should work out a plan (possibly involving silent install scripts) to quickly and easily reinstall and reconfigure the application server.

7.1.3 System test environment

The system test environment is a carefully controlled formal test environment. Development teams run their applications on this environment on a relatively infrequent basis – perhaps every six to eight weeks. A system test environment mirrors the production environment more closely than does a development integration environment, but it still does so at the smallest possible scale.

A key aspect of the system test environment is *formality*. The purpose of this environment is to ensure that the application truly deploys and runs as required in production. Thus, the system test team is responsible for testing all aspects of the application, including both functional and non-functional requirements. This is not a place to experiment. Development experiments are executed informally in the development integration runtime environment.

A system test environment may serve multiple masters. In addition to being used formally by testers, other groups may use it as well depending on what is appropriate to the environment.

For example, the administration staff may use this environment to test new patches and configuration changes before they are rolled into the pre-production and production environments.

Extended downtime in this environment can have an impact on delivery schedules. This environment is a key piece of the testing infrastructure and downtime means that important testing done in this environment cannot be undertaken. Still, risk is relatively low and with planning it should be possible to schedule several days to migrate this environment.

This is probably the first environment that administrators can use to gain experience with the migration tools and start to develop a plan for migrating the downstream environments. It may make sense to use one of the options outlined in 7.3, “Migration options” on page 95 to migrate this environment.

Your administrators should be prepared to “start over” if problems arise. It is important that issues be exposed and remediated during this migration. At the end of the day, you need to have a functioning system test environment and a good start on a formal migration plan. To this end, it may be necessary to reinstall the old environment to run through the migration scenario a second (or third) time to ensure that key concepts are understood and that outstanding issues are resolved. This is particularly true if your overall environment does not match the ideal and does not include a dedicated performance test environment.

7.1.4 Performance test environment

Performance and load testing is performed to find load-related problems in applications. This testing requires highly specialized skills and equipment to perform optimally. Hence, this is a dedicated environment and team.

Like the system test environment, the performance test environment is a carefully controlled formal test environment. Development teams run their applications on this environment on an even less frequent basis. A performance test environment mirrors the production environment in complexity, but it does so at the smallest possible scale.

Ideally this environment is owned and operated by a dedicated performance testing group whose members have specialized load testing skills. Each development team schedules time with this group. Typically, an application is load tested less frequently than it is run on the system test environment. As with a system test environment, one application is tested at a time.

Migration of the performance test environment carries similar risk to that of the system test environment. When this environment is unavailable, performance testing cannot be undertaken, which may impact delivery schedules. Much like for a system test environment, however, it should be possible to schedule a period of extended downtime to migrate this environment.

Your experience migrating the system test environment should have an impact on the migration of this environment. Here, you take what worked with the system test environment and apply it; you then modify what did not work. By this point, your migration process should be getting more formal and your administration staff should be quite comfortable with the migration process and the use of the migration tools (if applicable). The lessons learned while migrating this and the system test environments should translate into a formal (and documented) plan for migrating the pre-production environment.

You may consider running load testing scripts while you migrate this environment to confirm that the system continues to be responsive to user requests during the migration process.

7.1.5 Pre-production environment

The purpose of pre-production is to mimic production as much as possible (with *exactly* being the norm). This is the final chance to ensure that things really work in production.

This environment serves three purposes:

1. It gives the operations team a final place to familiarize themselves with the application and its procedures.
2. It provides the opportunity to test unrelated applications running together. This is crucial with shared deployment environments. Prior to this point, the applications have been tested and built independently.
3. It provides the operations team with a chance to test their operational procedures (backup, failover, problem resolution, and so forth).

The pre-production environment might also be used for user acceptance testing. In any case, testing on a pre-production staging environment generally coincides with an application's release schedule. Each external release of the application is tested on the pre-production system before it is finally moved into production. This environment is generally used to prove that the application works well with other applications.

For the purposes of migration, the pre-production environment should be treated as if it were the production environment. This is not an environment for experimenting; it is an environment for testing your formal migration plan. By the time you get to migrating this environment, your administrators should be very familiar with WebSphere Application Server V6.0 and should be confident in their plan for migration. Some tweaking of the plan is inevitable, but it is important to at least try to start this migration with as many answered questions as possible.

If possible, the pre-production environment should be migrated while under some test load to confirm that the application continues to respond throughout the process.

7.1.6 Production environment

Production is where you really run your applications. The key point is that if you have carefully followed procedures up to this point, the actual roll into production will be uneventful and predictable, since you have tested everything already.

The risks inherent in this environment being out of service are obvious: this is the environment that faces your customers and without this environment in service, it cannot make you money.

Many organizations have service-level agreements (SLAs) in place that dictate response times, throughput, maintenance windows, and acceptable amounts of downtime. Organizations that do not put formal SLAs in place tend to have informal ones that dictate (at least) how much downtime is permitted. Most organizations permit a very small amount of downtime, generally measured in minutes out of the year (these same organizations can tell you how much money it costs when their systems are down for a second).

Several minutes of acceptable downtime combined with short maintenance windows can typically be leveraged as part of runtime migration, but doing so requires careful planning and testing, and an effective backout strategy (what to do when something goes wrong). Honoring SLAs can be a real challenge during a migration.

7.2 Interoperability

Interoperability is part of almost every migration, at least for some short period of time while the runtime environment is migrated. In some cases, there may be a need to retain some number of servers running the older version. Use of features that are not supported by V6 might be one possible reason. Continued use of VisualAge® for Java's Enterprise Access Builders or the Common Connector Framework are examples. For organizations with large numbers of applications, it may not be practical to prepare, test, and certify all of the applications for V6.0 at the same time and so some extended period of interoperability may be required as those applications are pushed through.

Even if all the applications can be moved forward to WebSphere Application Server V6.0 at the same time, there is likely to be some short period of time when interoperability is required. The bottom line is that some amount of interoperability is inevitable.

7.2.1 Mixed version cell

WebSphere Application Server Version 6 supports a mix of Version 5.x and 6.0 nodes in a cell. There are, however, limitations to the support.

With a Version 6.0 deployment manager, you can:

- Start and stop V5 server instances
- Add and remove applications to V5 server instances

With a Version 6.0 deployment manager, you cannot:

- Add V5.x nodes
- Create new V5.x server instances
- Create new V5.x cluster members
- Add V5.x servers to existing clusters

Migrating a V5.x deployment manager to V6.0 is quite easy to do and works well. The interoperability works well with the restrictions noted above. Migrating the existing deployment manager can save significant time and money. Since migration conserves the current cell configuration, you do not need to spend time rebuilding it (though you may want to spend some time reviewing the settings to confirm that everything migrated as expected). You save money because you do not need to acquire additional hardware to make the migration work (see 7.3, "Migration options" on page 95).

Whether or not a mixed version cell is appropriate for your organization depends on a number of factors. Perhaps the most important factor is the amount of time that the complete migration takes. In every migration, there is some period of interoperability while a new production runtime is rolled out. If all the applications have been migrated, tested, and declared ready for Version 6.0, then the period of interoperability is relatively short (hours or days). If only some applications are declared ready for Version 6.0, then your complete migration to Version 6.0 may take weeks or months.

Why does the period of time required to migrate matter? For some organizations, the ability to reconstruct an environment from scratch (perhaps in the event of hardware or catastrophic failure) is important. It is time consuming to reconstruct a mixed version configuration from scratch. A mixed version cell can be reconstructed using a configuration backup. Without a backup, a mixed version cell must be reconstructed by first rebuilding the V5.x nodes and then repeating the migration. This can be very time consuming.

Furthermore, it is difficult, or sometimes impossible, to make significant modifications to V5.x nodes in a V6.0 cell. Specifically, since new V5.x server instances cannot be added, it is very difficult to reprovision V5.x assets. You could not, for example, add additional V5.x nodes to handle an increase in traffic should the need arise. If your V5.x applications cannot be migrated immediately, it may be better to retain your V5.x cell until those applications can be migrated.

It is important to note that J2EE 1.3 applications deployed on WebSphere Application Server V5.x should require little or no modification to run on V6.0 and, in general, testing should amount to little more than running existing test scripts with the application deployed on the new environment. While this is generally true, some organizations may choose to do more extensive tests—including, for example, performance testing—prior to production deployment. Such testing can take a considerable amount of time, especially when multiple applications are involved.

7.2.2 HTTP servers

As part of your application server migration, it may be necessary to migrate your HTTP server as well. Table 7-1 shows a support matrix for the HTTP servers that lists all the HTTP servers supported by WebSphere Application Server V6.0 at the time of this writing[2]. It also shows whether or not each of those HTTP server versions is supported by previous versions of WebSphere Application Server.

Table 7-1 Support matrix for HTTP servers

	6.0	**5.1.1**	**5.1**	**5.02**	**4.07AE**
Apache 2.0.49	Yes	No	No	No	No
Covalent 1.4	No	Yes	Yes	No	No
Covalent 2.3.2	No	Yes	Yes	Yes	Yes
Covalent 2.4	No	Yes	Yes	No	No
IBM HTTP Server 2.0.47.1	Yes	Yes	Yes	No	No
IBM HTTP Server 6.0	Yes	No	No	No	No
Internet Information System 5.0	Yes	Yes	Yes	Yes	Yes
Internet Information System 6.0	Yes	Yes	Yes	Yes	No
SunOne 6.0 SP7	Yes	Yes	No	No	No
SunOne 6.1 SP1	Yes	Yes	No	No	No

During a rolling migration, there is some period of interoperability where you have both the existing and new versions of WebSphere Application Server running concurrently. During that period of interoperability (depending on what version of the HTTP server you are currently using), you may have to take care that the tiers are configured correctly.

If your current topology organizes your HTTP and application servers into silos (that is, each application server node has a dedicated HTTP server) as shown in Figure 7-3 on page 93, then the HTTP server version differences are irrelevant. You can update the HTTP server as the application server is upgraded.

[2] For up to date information, see http://www-306.ibm.com/software/webservers/appserv/doc/latest/prereq.html

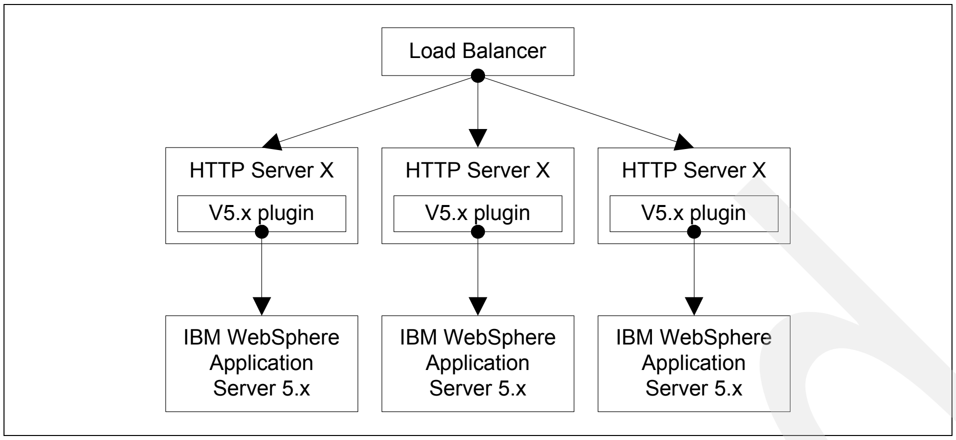

Load Balancer
HTTP Server X
V5.x plugin
HTTP Server X
V5.x plugin
HTTP Server X
V5.x plugin
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 5.x

Figure 7-3 A topology organized in silos

Since the HTTP server is effectively dedicated to a single instance of WebSphere Application Server, it can easily co-exist with mixed HTTP server versions as shown in Figure 7-4.

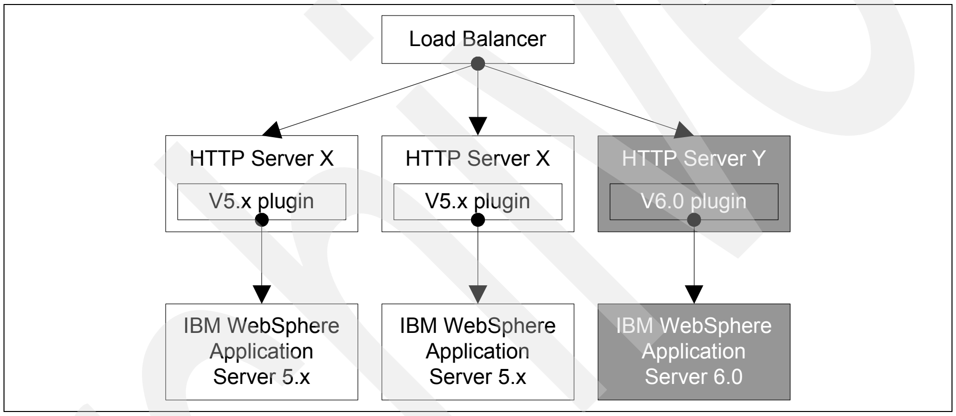

Load Balancer
HTTP Server X
V5.x plugin
HTTP Server X
V5.x plugin
HTTP Server Y
V6.0 plugin
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 6.0

Figure 7-4 With a silo topology, mixed HTTP server versions can easily work together

This form of interoperability is relatively easy to maintain through a rolling migration (or even for extended periods if required).

For more complex topologies, compatibility can have a significant impact on the migration path. Figure 7-5 on page 94 shows a more complex configuration that employs workload management from the HTTP servers to a collection of application server nodes.

.

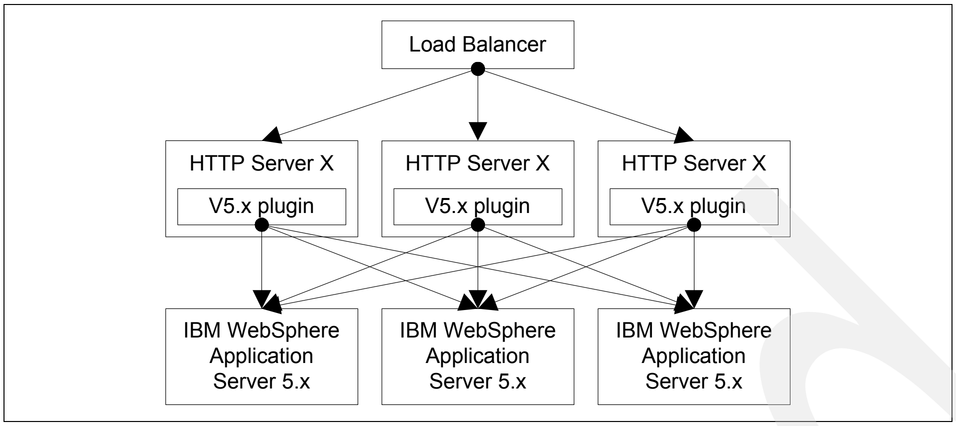

Load Balancer
HTTP Server X
V5.x plugin
HTTP Server X
V5.x plugin
HTTP Server X
V5.x plugin
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 5.x

Figure 7-5 A more complex topology that uses workload management

Version differences between HTTP servers impact the strategy for moving forward. In this event, different versions of the HTTP servers need to be fitted into silos temporarily during the rolling migration. Figure 7-6 shows an example where the first application server node is upgraded to V6.0 along with one of the HTTP servers.

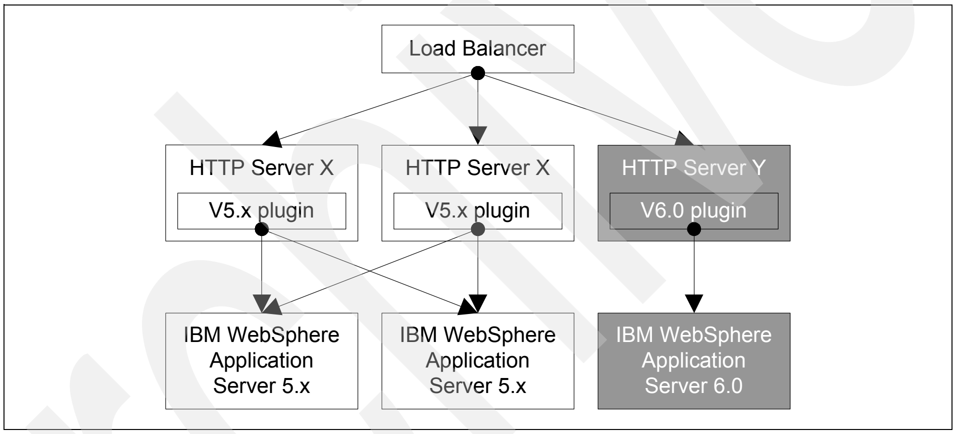

Load Balancer
HTTP Server X
V5.x plugin
HTTP Server X
V5.x plugin
HTTP Server Y
V6.0 plugin
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 6.0

Figure 7-6 Upgrading first server node

As the migration progresses, the new V6 silo is extended as shown in Figure 7-7.

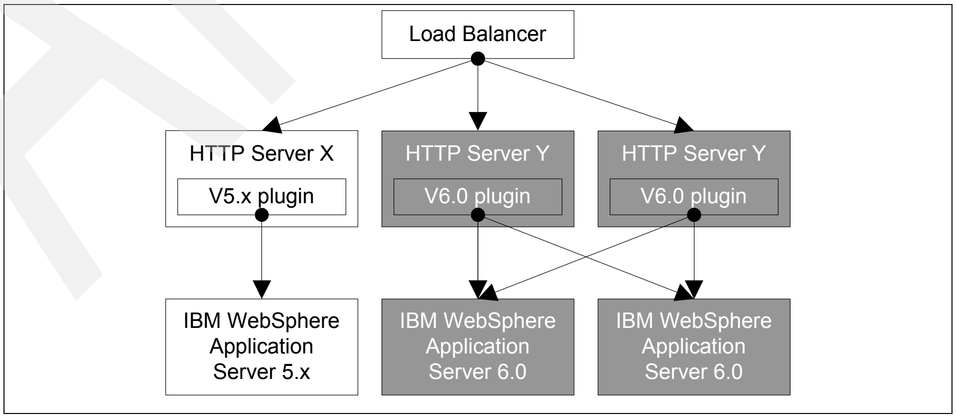

Load Balancer
HTTP Server X
V5.x plugin
HTTP Server Y
V6.0 plugin
HTTP Server Y
V6.0 plugin
IBM WebSphere Application Server 5.x
IBM WebSphere Application Server 6.0
IBM WebSphere Application Server 6.0

Figure 7-7 Upgrading second server node

By the end of the migration, all HTTP servers and application server nodes are migrated and the original topology is again in place.

The previous two examples represent “horizontal” migrations. This means that the migration must span tiers. If a common version of the HTTP server is in place at the start of the migration, then the migration can be processed vertically. In this case, the entire HTTP server tier might be completely migrated before a single application server node is migrated.

7.3 Migration options

There are three ways to migrate an existing WebSphere Application Server cell to Version 6.0. The path you choose depends on a number of factors, including:

- Expected time to complete the migration
- Amount of acceptable risk
- Availability of hardware/resources
- Trust in automated tools

The three runtime migration options are presented in the following sections.

7.3.1 Fully automated migration

The fully automated migration option uses the **WASPreUpgrade** and **WASPostUpgrade** tools to perform the migration. Strictly speaking, the migration is not fully automatic. It is more accurate to say that the configuration of the migrated elements is automated. We refer to this approach as “fully automated” because we use the migration tools on every node in the cell.

The migration tools are used as part of a four-step migration process:

1. Run **WASPreUpgrade** on the existing WebSphere Application Server configuration.
2. Uninstall the existing version of WebSphere Application Server.
3. Install the new version of WebSphere Application Server.
4. Run **WASPostUpgrade** on the new version.

In the second step, it is recommended that rather than uninstalling the existing version, it be made dormant. By keeping the existing software installed on the machine, it will be much easier to back out of the migration should unexpected problems arise.

See 10.2.3, “Multiple nodes for V5” on page 156 for more details about the fully automated approach to migration.

The fully automated migration includes the following steps:

1. Migrate the existing Version 5.x deployment manager to Version 6.0 using the migration tools.
2. Migrate the HTTP servers and HTTP server plug-ins on all Web tier nodes.
3. Individually migrate the application server nodes using the migration tools.

If you must upgrade the HTTP server as part of your migration, the final two steps may need to be interleaved with groups of HTTP servers and application servers being migrated in steps.

This method has the benefit of not requiring any additional hardware to work. The existing machine running the existing deployment manager can be upgraded while the application server nodes continue to receive and process requests. The Version 5.x application server nodes continue to work even while the deployment manager is out of service for the upgrade. Additionally, this method requires the least amount of time and manual intervention. The **WASPreUpgrade** and **WASPostUpgrade** tools effectively move the existing configuration into WebSphere Application Server Version 6.0.

7.3.2 Partially automated migration

Like the fully automated migration, this method uses the migration tools, but only to migrate the deployment manager. You might use this method if you are not 100% confident in the abilities of the migration tools and you need a mixed version cell (automated migration of the deployment manager is the only way to include Version 5.x nodes in a Version 6.0 cell).

The steps for a partially automated migration include:

1. Migrate DM using Migration Wizard.
2. Migrate HTTP plug-in on all HTTP server nodes.
3. Migrate nodes manually.

Again, you may have to interleave the migration of the HTTP servers and application servers if you must upgrade the HTTP server as part of this migration.

When migrating nodes, first remove them from the cell, install the new version of WebSphere Application Server and then add them back into the cell. In our experiments with the early version, the “remove node” functionality did not work when attempting to remove a Version 5.x node from a Version 6.0 cell. To remove the node, we had to do a “force remove” from the console.

7.3.3 Manual migration

In a manual migration, a new Version 6.0 cell is created to first augment and then replace the existing Version 5.x cell. You might use this method if you need to keep Version 5.x running for an extended period (that is, you require an extended period of interoperability) and you have adequate hardware to support a second deployment manager. As noted earlier, an extended period of interoperability may be necessary if you have a large number of applications and some, but not all, of them are ready for deployment on WebSphere Application Server Version 6.0.

In this method, each node runs either Version 5.x or 6.0 and is a member of exactly one cell. Since the migration process itself is completely manual, it is impossible to have a mixed version cell (that is, it is not possible to add a Version 5.x node to a Version 6.0 cell). It may be possible to install and run both versions of the application server on a single physical machine; that is limited by available resources.

The steps for this method include:

1. Create a new (parallel) WebSphere Application Server Version 6.0 deployment manager.
2. Migrate the HTTP server plug-in on selected HTTP server nodes. Only migrate those HTTP servers that will be handling traffic destined for Version 6.0.
3. Remove the node from the Version 5.x deployment manager, upgrade the node to Version 6.0, and add it to the V6 deployment manager.

Depending on available resources, it may be possible to install the deployment manager for both versions on the same machine. Alternatively, if there is sufficient capacity in the existing infrastructure, one of the existing nodes might be temporarily made the deployment manager for the new cell (this assumes that you have a homogeneous collection of machines running your cell).

8

Distributed runtime administration overview

This chapter discusses the most important aspects of how WebSphere Application Server Version 6 is administered on the distributed operating platforms.

8.1 Significant concept changes from previous versions

You should already be somewhat familiar with WebSphere Application Server and specifically, how to administer V4 or V5. Given that, the quickest way to be productive on V6 is to understand the significant administration task changes.

8.1.1 Significant changes from V4

If you are familiar with administering V4, you should be aware of these changes.

XML file based administrative repository

WebSphere Application Server Version 4 Advanced Edition manages application servers with a single administration server. The administration server accesses server configuration data that is housed in a relational database. WebSphere Application Server Version 6 performs the same server management function via an administration server that is called the deployment manager. The deployment manager accesses configuration data that is implemented as files in XML format. See 8.3, “Administrative console” on page 113 and 8.4, “How to perform basic functions” on page 115 for details about how the deployment manager is administered.

The concept of a deployment manager was introduced in the WebSphere Application Server Network Deployment Version 5 product. V6 continues this same concept, and therefore readers familiar with V5 operation can apply this same knowledge to administering V6.

HTTP administrative console

The administrative console that is used to interact with the administration server for WebSphere Application Server Version 4 Advanced Edition is a Java application client. Both WebSphere Application Server Version 5 and Version 6 products use an HTTP-based administrative console that is accessed via a Web browser. See 8.3, “Administrative console” on page 113 for more details about how to access the administrative console.

Installation

WebSphere Application Server Version 4 installs the application server, the Web server, and the Web server plug-ins with a single installer that installs all three in sequence. WebSphere Application Server Version 6 uses independent installers to install these three components. See 8.2, “Installation” on page 100 for details about how to perform installations.

8.1.2 Significant changes from V5

If you are familiar with administering V5, you should be aware of these changes.

Asynchronous messaging architecture

WebSphere Application Server Version 5 implements asynchronous messaging with a native code implementation that is derived from the WebSphere MQ product line. This native implementation is loosely integrated with the Java environment that comprises the application server execution platform.

WebSphere Application Server Version 6 implements asynchronous messaging entirely in Java. The integration with the application server Java environment is seamless. Consequently the administration of the messaging components is performed differently compared to V5. Figure 8-1 on page 99 shows the conceptual contrast between the two implementations.

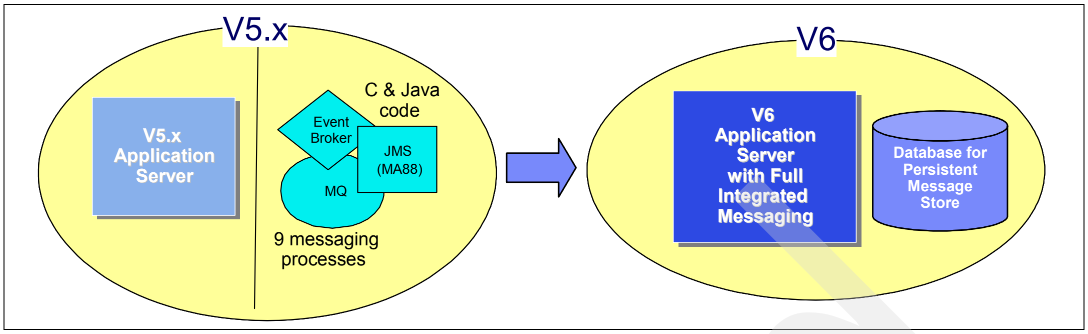

V5.x
V5.x Application Server
C & Java code
Event Broker
JMS (MA88)
MQ
9 messaging processes
V6
V6 Application Server with Full Integrated Messaging
Database for Persistent Message Store

Figure 8-1 Comparison of Version 5 versus Version 6 messaging implementation

In addition to the implementation differences, asynchronous messaging has a different architecture. Components that are new are a messaging communications component called the System Integration Bus, or SIBus, and JMS Unified Connection Factories. Figure 8-2 summarizes the relationships between these components.

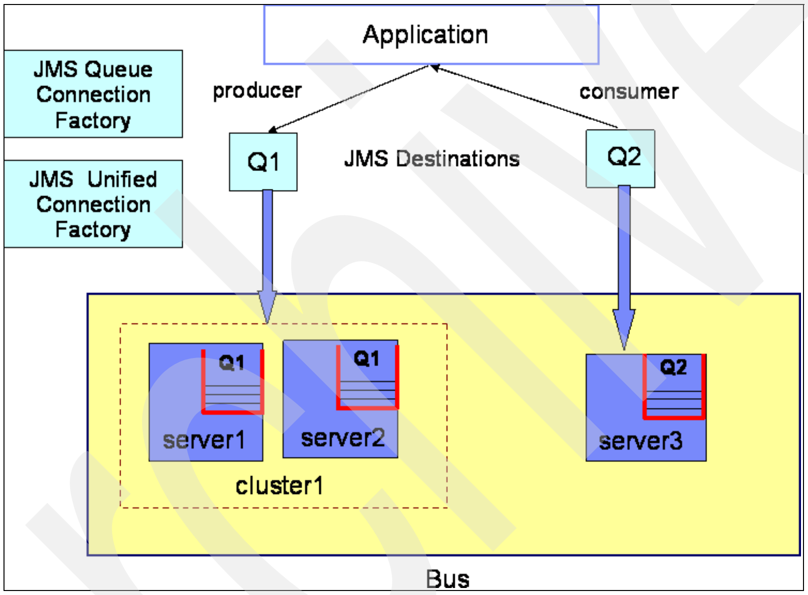


Figure 8-2 Version 6 Messaging components

An application that uses point-to-point messaging acts as a producer or consumer of messages with JMS queues.

An administrator can define a JMS queue, an administrative object that encapsulates the queue name and other configuration properties that the administrator wants to preserve.

The administrator configures the queue onto a destination on a SIBus. Such a queue is available, over a long period of time, to all applications with access to the destination. The destination is provided by only one member (an application server or cluster) of the service integration bus. A destination for a queue is localized to either an application server or cluster. A queue that is localized to a cluster is partitioned across the servers in that cluster.

A JMS connection factory creates connections to the messaging engine that localizes the destination. The preferred way for an administrator to define a JMS connection factory for queues is to define a unified JMS connection factory. Such unified JMS connection factories support both queues and topics, which enables applications to use the same common connection factories. As an alternative to defining unified JMS connection factories, an administrator can define domain-specific JMS queue connection factories, as used for administration before JMS 1.1.

See the IBM Education assistant section on service integration technologies for a set of tutorials and a more detailed discussion on the messaging design:

ftp://ftp.software.ibm.com/software/eod/WAS_6-0/WPM/index.html

Installation

WebSphere Application Server Version 5 installs the application server, the Web server, and the Web server plug-ins with a single installer that installs all three in sequence. WebSphere Application Server Version 6 uses independent installers to install these three components. See 8.2, “Installation” for details on how to perform installations.

Profiles

Profiles are a concept of packaging a WebSphere installation into a single set of read-only binary executable files, and multiple sets of writable data files that represent application server instances. See 8.2.3, “Profiles” on page 105 for details of how profiles work and how to administer them.

8.1.3 Concepts preserved from V5

If you are familiar with V5, these topics are a review, but if you have only used V4, you should be aware of these conceptual changes.

HTTP administrative console

The administrative console has much the same structure compared to Version 5, but the screens have been reformatted to reduce scrolling. Some of the configuration categories are changed or moved. The default address has changed to:

http://localhost:9060/ibm/console

See 8.3, “Administrative console” on page 113 for more details on the changes to the administrative console.

Cell administration

Network Deployment cells continue to be managed by a deployment manager as they had been in V5. The deployment manager manages the configuration data for the nodes in the cell and pushes that data to the node as required. The concept of a cell has been expanded to include both V5 nodes and V6 nodes in order to allow a staged migration of large numbers of nodes. See 10.2.3, “Multiple nodes for V5” on page 156 for an overview of cell migrations. Also see “Starting servers” on page 116 for details about starting and stopping cell members.

8.2 Installation

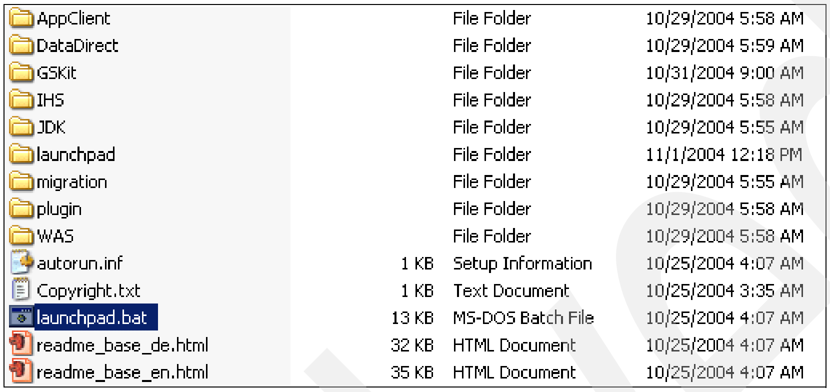
Installation of the WebSphere Application Server Version 6 has changed the basic structure and sequence of the installers. Previous versions used a single integrated installer which installed the application server, Web server, and Web server plug-ins all in one installation step. Version 6 uses independent installer steps coordinated by a Web page containing links to the individual installers.

Another major new aspect of the application server installer is that installation is now segmented into independent installation wizards.

8.2.1 Launchpad

The installer launchpad page offers links that start each of the individual installers.

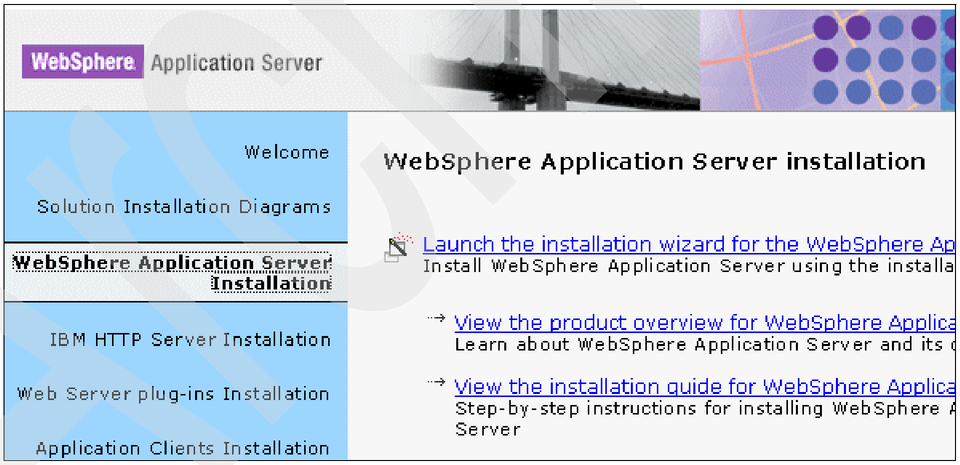
The launchpad is started by running the launchpad script from the top directory of the installation media. Figure 8-3 shows a Windows file view of the installation media. If your operating system is not Windows, the filename would be launchpad.sh and it would be in the same position on the media.



AppClient		File Folder	10/29/2004 5:58 AM
DataDirect		File Folder	10/29/2004 5:59 AM
GSKit		File Folder	10/31/2004 9:00 AM
IHS		File Folder	10/29/2004 5:58 AM
JDK		File Folder	10/29/2004 5:55 AM
launchpad		File Folder	11/1/2004 12:18 PM
migration		File Folder	10/29/2004 5:55 AM
plugin		File Folder	10/29/2004 5:58 AM
WAS		File Folder	10/29/2004 5:58 AM
autorun.inf	1 KB	Setup Information	10/25/2004 4:07 AM
Copyright.txt	1 KB	Text Document	10/25/2004 3:35 AM
launchpad.bat	13 KB	MS-DOS Batch File	10/25/2004 4:07 AM
readme_base_de.html	32 KB	HTML Document	10/25/2004 4:07 AM
readme_base_en.html	35 KB	HTML Document	10/25/2004 4:07 AM

Figure 8-3 Launchpad location

Running the launchpad script starts the default browser; the browser screen looks similar to Figure 8-4. Each installer is started by selecting the appropriate link in the blue navigation bar on the left side of the launchpad page. When you select the link at the left, the right side of the page changes. You then select the link that begins with the phrase `Launch the installation wizard`.


WebSphere Application Server
Welcome
Solution Installation Diagrams
WebSphere Application Server Installation
IBM HTTP Server Installation
Web Server plug-ins Installation
Application Clients Installation

WebSphere Application Server installation

Launch the installation wizard for the WebSphere Ap
Install WebSphere Application Server using the installa

View the product overview for WebSphere Applica
Learn about WebSphere Application Server and its

View the installation guide for WebSphere Applica
Step-by-step instructions for installing WebSphere A
Server

Figure 8-4 Launchpad initial screen

8.2.2 Individual component installers

Each component of the WebSphere Application Server Version 6 installation is created by an installer specific to that component.

Application server

The application server installer is started from the launchpad page. It is a Java installer that installs the binary portion of WebSphere Application Server. The application server is installed in two major phases. The first phase consists of installation of the product binary read-only files. The second phase of the installation is the creation of a profile, which is a conceptual container for all the writable files that constitute the configuration description for an application server. After the binary files are installed and a profile is created, the installation is considered complete. If you need to run multiple servers on the same system, you would optionally create additional profiles to correspond to those additional server configurations.

WebSphere Application Server and WebSphere Application Server Express are both products that use only application servers as the building blocks of systems. Figure 8-5 applies explicitly to these products in that the installation wizard creates a default profile that describes an application server configuration. Contrast this notion of these two products to the WebSphere Application Server Network Deployment product by examining Figure 8-6. WebSphere Application Server Network Deployment uses three types of configuration descriptions, namely application servers, deployment managers, and node agents. Therefore, WebSphere Application Server Network Deployment needs to create three different types of profile to represent those types of configuration entities. The installation wizard for WebSphere Application Server Network Deployment gives you the option to start the profile creator, which is a separate command that creates profiles.

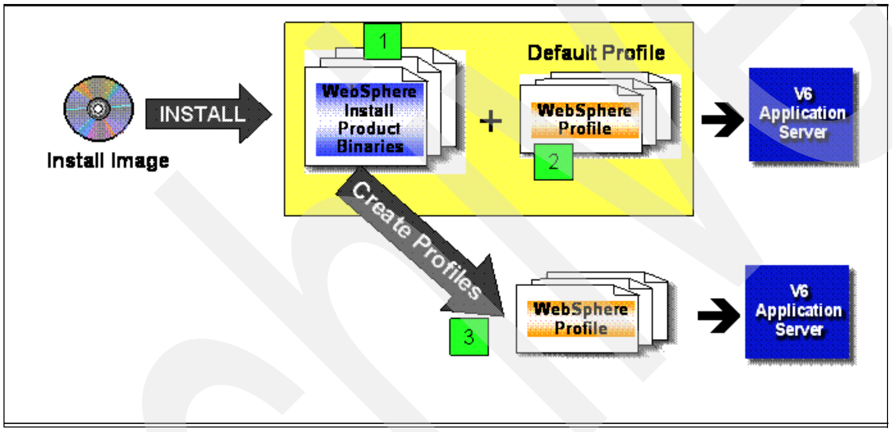


Figure 8-5 Conceptual installation for WebSphere Application Server

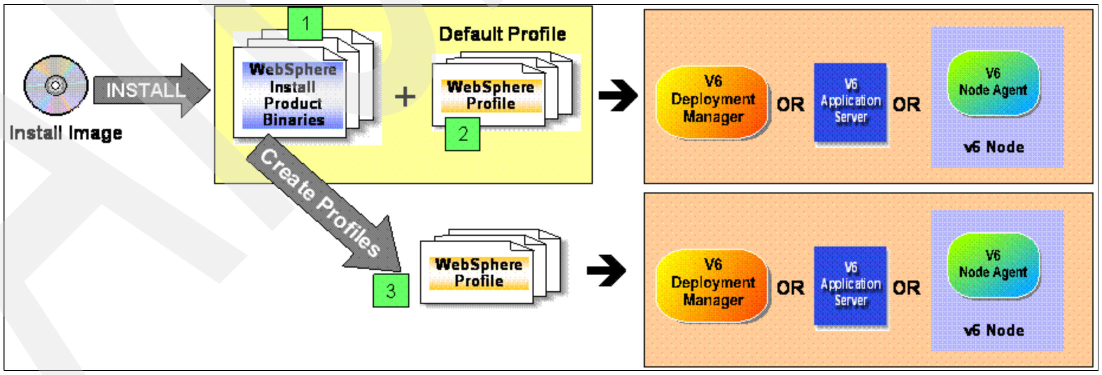
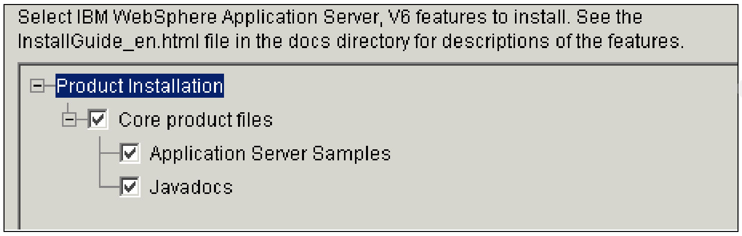


Figure 8-6 Conceptual installation for WebSphere Application Server Network Deployment

After launching the installer, you accept the software license agreement and specify the directory where you want to install the binary files. The installer also verifies that you have the correct operating system prerequisites and prompts you if your system does not meet the prerequisites.

Figure 8-7 shows the feature selection choices you can make. Notice that the Web server is no longer listed here as it was in previous versions. You can only choose whether you want samples installed or the Javadoc™ documentation.



Select IBM WebSphere Application Server, V6 features to install. See the InstallGuide_en.html file in the docs directory for descriptions of the features.

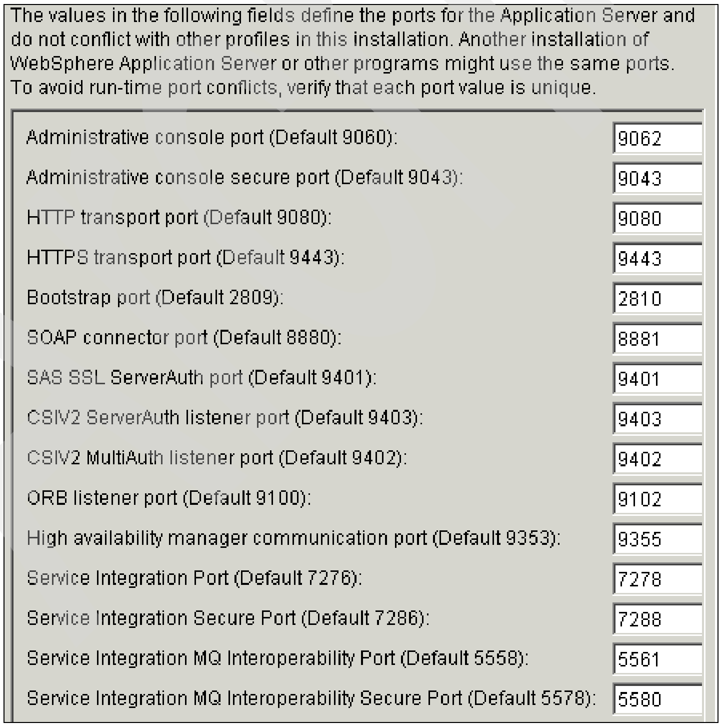
- Product Installation
  - ☑ Core product files
    - ☑ Application Server Samples
    - ☑ Javadocs

Figure 8-7 Installation selection choices

If you are installing WebSphere Application Server Network Deployment, the installer would proceed to confirm your selections and install all the binary files. When the installer is done installing the binary file portion, you are given the option to run the profile creation wizard. See "Creating profiles" on page 106 for more information about the profile creation wizard.

If you are installing WebSphere Application Server or WebSphere Application Server Express, the installer would present more screens to fill in. For these products the first default profile configuration is integrated with installation of the binary portion.

Figure 8-8 shows the port number assignments. The default selections work fine if you are only installing one application server on this system. As you create more profiles, the ports increment to avoid port conflicts. If you have multiple installations on the same system, you must be aware of all the port assignments and assign numbers such that there are no conflicts.

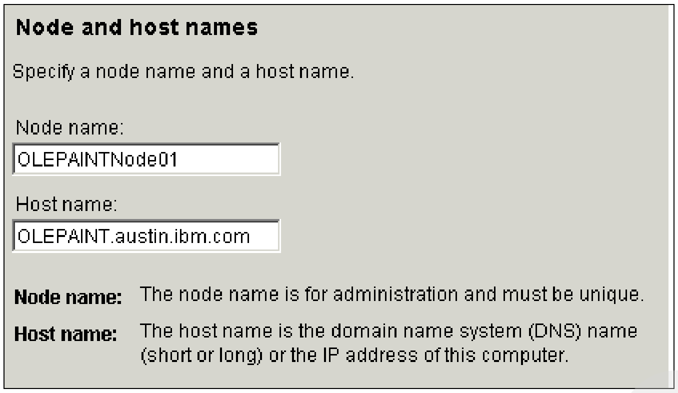


The values in the following fields define the ports for the Application Server and do not conflict with other profiles in this installation. Another installation of WebSphere Application Server or other programs might use the same ports. To avoid run-time port conflicts, verify that each port value is unique.

Administrative console port (Default 9060):	9062
Administrative console secure port (Default 9043):	9043
HTTP transport port (Default 9080):	9080
HTTPS transport port (Default 9443):	9443
Bootstrap port (Default 2809):	2810
SOAP connector port (Default 8880):	8881
SAS SSL ServerAuth port (Default 9401):	9401
CSIV2 ServerAuth listener port (Default 9403):	9403
CSIV2 MultiAuth listener port (Default 9402):	9402
ORB listener port (Default 9100):	9102
High availability manager communication port (Default 9353):	9355
Service Integration Port (Default 7276):	7278
Service Integration Secure Port (Default 7286):	7288
Service Integration MQ Interoperability Port (Default 5558):	5561
Service Integration MQ Interoperability Secure Port (Default 5578):	5580

Figure 8-8 Installation port assignments

Figure 8-9 on page 104 shows the screen that asks for the node name and host name. You can specify any name for the node name. The host name typically matches the hostname setting of the operating system, which is probably the correct value. You may need to specify some other name if the system has multiple network interfaces.



Node and host names

Specify a node name and a host name.

Node name:
OLEPAINTNode01

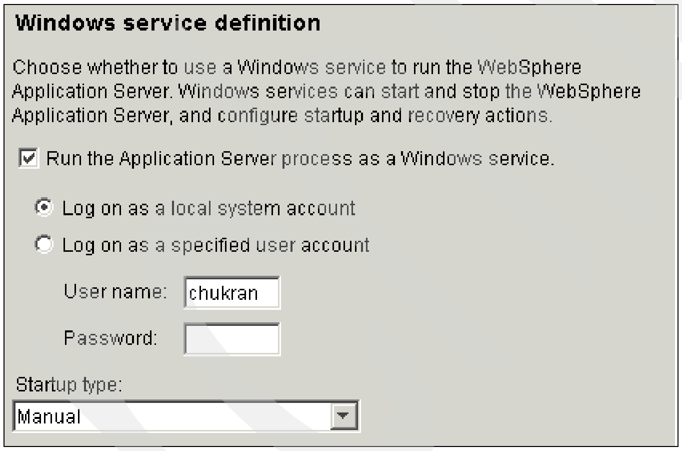
Host name:
OLEPAINT.austin.ibm.com

Node name: The node name is for administration and must be unique.

Host name: The host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Figure 8-9 Installation node name and host name

Figure 8-10 shows the Windows Services definition, which is displayed only for a Windows operating system. If the top box is checked, a Windows service is created for this server. You would then be able to start the server through the Windows Services control panel. By default, this service runs as the System account. If you do not want the System account, you can specify an arbitrary user and password if you check the last box.



Windows service definition

Choose whether to use a Windows service to run the WebSphere Application Server. Windows services can start and stop the WebSphere Application Server, and configure startup and recovery actions.

☑ Run the Application Server process as a Windows service.

● Log on as a local system account

○ Log on as a specified user account

User name: chukran

Password:

Startup type:
Manual

Figure 8-10 Installation service definition

The installer presents a final screen confirming your values, the binary files are installed, and a default profile, named default, is created.

Web server

The Web server that is bundled with WebSphere Application Server Version 6 is IBM HTTP Server Version 6. In previous versions this Web server installation was integrated with the application server installation. For V6, IBM HTTP Server Version 6 is installed independently with a separate installer. The installer is launched via the launchpad. The installation is straightforward and asks for information such as directory location, port numbers for the HTTP port and administrative port, and configuration parameters for the Windows Service. You can default all the screen fields except for the Windows Service settings, which are optional. If you select to have the startup managed by Windows Services, you must either specify the system user, or specify a valid user and password.

Web Server plug-ins

The WebSphere Web server plug-ins are installed with a separate installer, which also is accessed through the Launchpad. See 8.2.1, “Launchpad” on page 101 for more details on the launchpad.

The plug-in is native executable code that analyzes incoming HTTP requests and routes appropriate requests to a WebSphere Application Server. The Web server plug-ins must be installed on the same system as the Web server because the plug-in files must be locally accessible to the Web server.

The plug-in installer asks you for several options, specifically:

- Directory into which to install.
- Location of Web server and its configuration file.
- Type of Web server.
- Is the application server remote or local from the Web server?
- Name of the Web server by which it is identified in the administrative console.
- Location of the application server when local is specified.
- Location of the plug-in configuration file.

If you specify `local` such that the application server is on the same system, the installer creates a Web server instance under the administrative console Web servers section. If you specify `remote`, the installer creates a script that you must transport to the system where the application server is installed. You then run the script on that system to create the Web server instance.

See 12.3.6, “Migrating Web server and WebSphere plug-ins” on page 201 for a complete example of creating a local Web server.

8.2.3 Profiles

A new method of managing installation configurations is the concept of creating multiple profiles under a single installation on a system. Understanding this concept is crucial in learning how to create cells of multiple nodes.

Understanding profiles

Profiles allow multiple server configurations to share the same binary files. WebSphere Application Server Version 4 allows multiple installations to achieve the creation of multiple servers on the same system. This type of configuration wastes disk space because it creates multiple copies of the Java binary files. WebSphere Application Server Version 5 creates an improvement over this situation with the **`wsinstance`** command. **`wsinstance`** creates multiple servers by creating sets of writable configuration files for each server. The servers can then share the same Java binary files, thus saving disk space.

Since profiles are a more general solution, the V5 **`wsinstance`** command no longer exists. **`wsinstance`** has been replaced by the profile creation wizard and the **`wasprofile`** command.

Profiles do not have any limitations in that each profile has the same flexibility and power as any of the other profiles. There is no distinction of profiles with regard to the order of creation. Table 8-1 on page 106 shows that there are three profile types. Table 8-2 on page 106 shows that only WebSphere Application Server Network Deployment is capable of creating Deployment Manager and Custom profiles because the function of those types of profiles only makes sense in a Network Deployment cell environment.

Table 8-1 Purposes for profile types

Profile type	Creates
Application Server	Single application server
Deployment manager	New cell with single deployment manager
Custom	Federated node with node agent but no application servers

Table 8-2 Profile types allowed

Product	Allows these profiles
WebSphere Application Server	Application Server
WebSphere Application Server Express	Application Server
WebSphere Application Server Network Deployment	Application Server Deployment manager Custom

Creating profiles

Profiles are created by launching the profile creation wizard. Figure 8-11 shows the location of the profile creation wizard for a Windows system. If you have a different operating system, it will be in the same relative location within the installation directory, however the name of the file will be slightly different depending on your operating system.

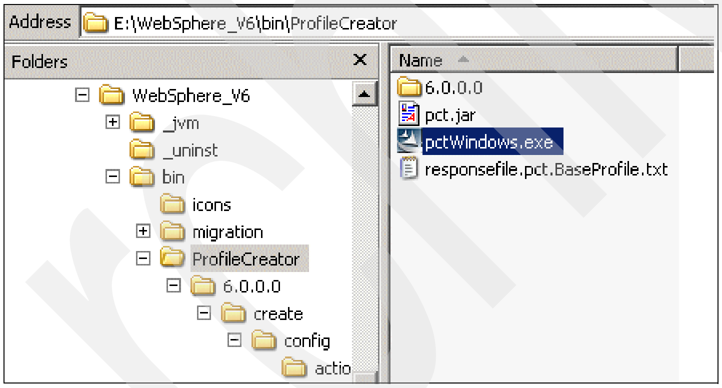
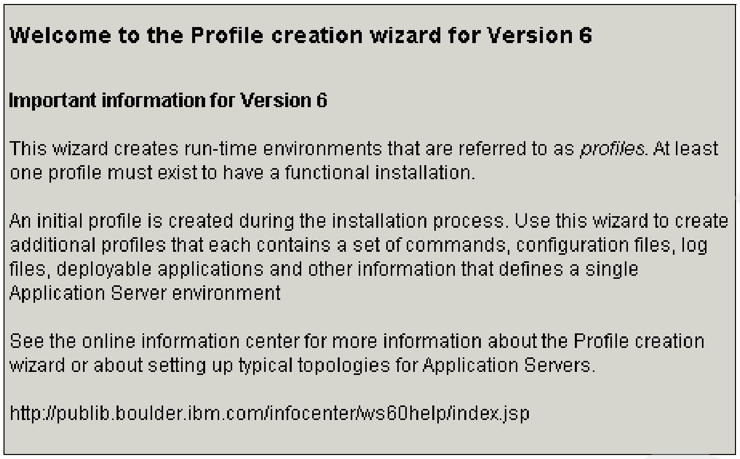


Figure 8-11 Location of Profile Creation Wizard

Figure 8-12 on page 107 shows the initial screen of the profile creation wizard.



Welcome to the Profile creation wizard for Version 6

Important information for Version 6

This wizard creates run-time environments that are referred to as *profiles*. At least one profile must exist to have a functional installation.

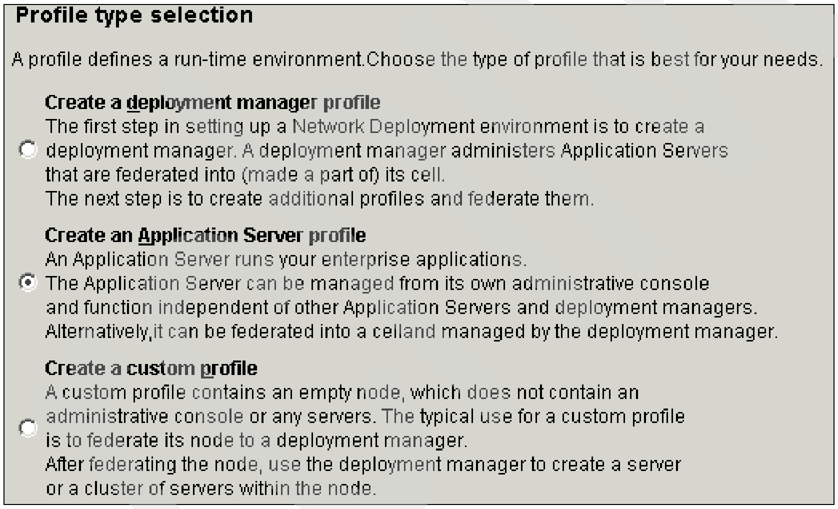
An initial profile is created during the installation process. Use this wizard to create additional profiles that each contains a set of commands, configuration files, log files, deployable applications and other information that defines a single Application Server environment

See the online information center for more information about the Profile creation wizard or about setting up typical topologies for Application Servers.

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp

Figure 8-12 Initial screen for Profile creation wizard

Figure 8-13 shows the choice of type of profile you wish to create. You are presented with this screen only for WebSphere Application Server Network Deployment. If you are creating a profile for WebSphere Application Server Express or WebSphere Application Server, this selection screen is not shown because Application Server is the only available type.



Profile type selection

A profile defines a run-time environment.Choose the type of profile that is best for your needs.

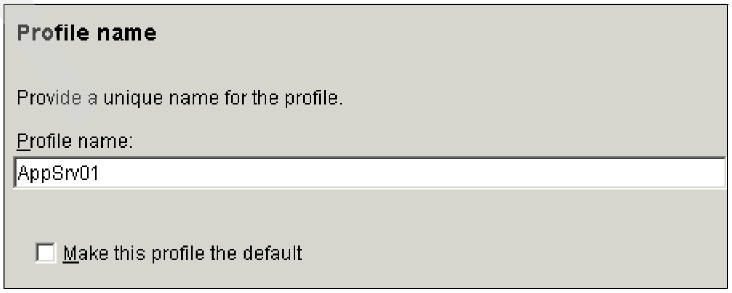
○ **Create a deployment manager profile**
The first step in setting up a Network Deployment environment is to create a deployment manager. A deployment manager administers Application Servers that are federated into (made a part of) its cell.
The next step is to create additional profiles and federate them.

● **Create an Application Server profile**
An Application Server runs your enterprise applications.
The Application Server can be managed from its own administrative console and function independent of other Application Servers and deployment managers.
Alternatively,it can be federated into a celland managed by the deployment manager.

○ **Create a custom profile**
A custom profile contains an empty node, which does not contain an administrative console or any servers. The typical use for a custom profile is to federate its node to a deployment manager.
After federating the node, use the deployment manager to create a server or a cluster of servers within the node.

Figure 8-13 Profile choices

Figure 8-14 and Figure 8-15 on page 108 show selection of the profile name and location. The default location is within the installation directory, but you can place the profile in any directory.



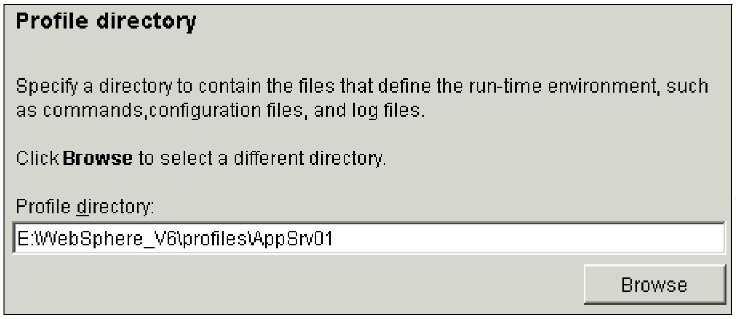
Profile name

Provide a unique name for the profile.

Profile name:
AppSrv01

□ Make this profile the default

Figure 8-14 Profile name selection



Profile directory

Specify a directory to contain the files that define the run-time environment, such as commands,configuration files, and log files.

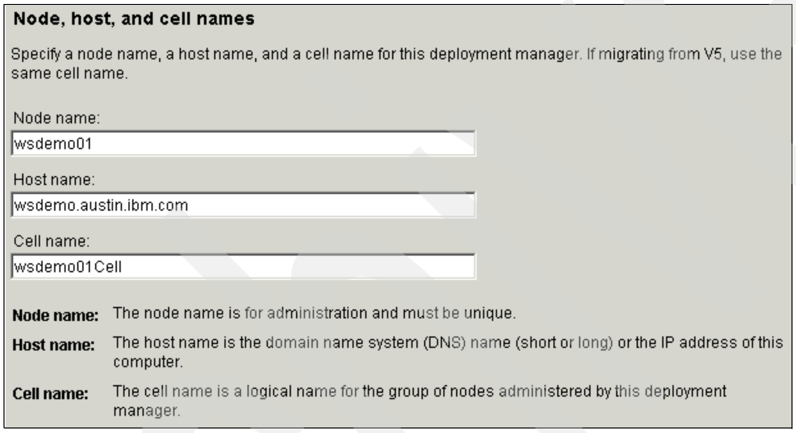
Click **Browse** to select a different directory.

Profile directory:

E:\WebSphere_V6\profiles\AppSrv01

Figure 8-15 Profile directory location

Figure 8-16 shows specification of the node name, cell name, and host name for the application server. If you are creating a profile for WebSphere Application Server Express or WebSphere Application Server, this cell name is not shown because the cell name is not pertinent to these products.



Node, host, and cell names

Specify a node name, a host name, and a cell name for this deployment manager. If migrating from V5, use the same cell name.

Node name:

wsdemo01

Host name:

wsdemo.austin.ibm.com

Cell name:

wsdemo01Cell

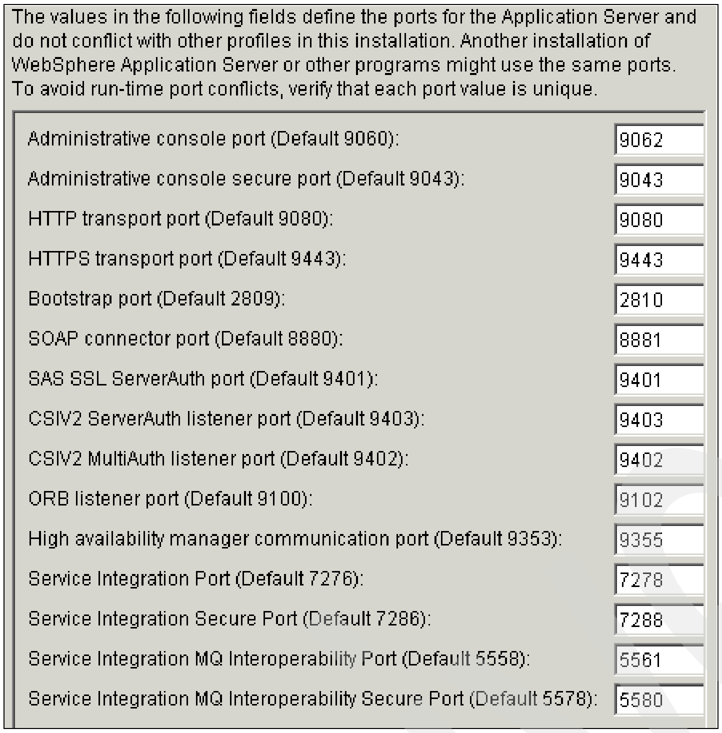
Node name: The node name is for administration and must be unique.

Host name: The host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Cell name: The cell name is a logical name for the group of nodes administered by this deployment manager.

Figure 8-16 Node, host, and cell name for profiles

Figure 8-17 on page 109 shows the entire list of server ports that can be changed. This screen shows all the default values. The profile creation wizard detects that there are multiple profiles already in existence and increments some of the port numbers in an attempt at maintaining port uniqueness. You should not rely on this feature. You should change port assignments by strict analysis of which servers will be running on the system concurrently.

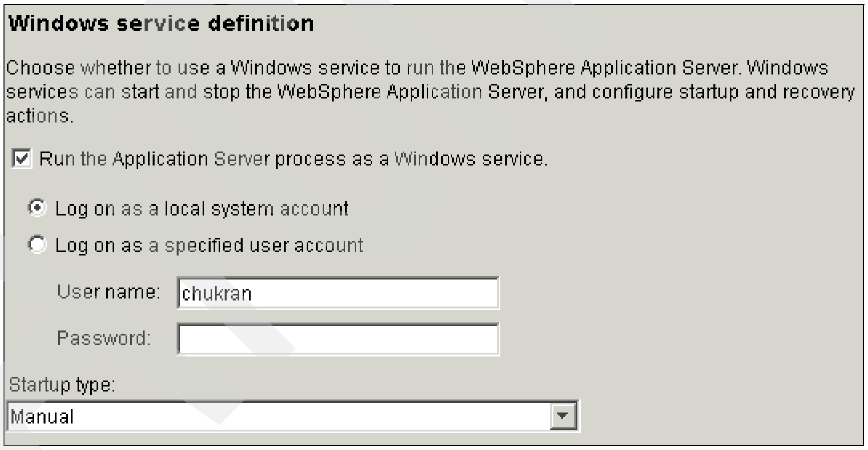


The values in the following fields define the ports for the Application Server and do not conflict with other profiles in this installation. Another installation of WebSphere Application Server or other programs might use the same ports. To avoid run-time port conflicts, verify that each port value is unique.

Administrative console port (Default 9060):	9062
Administrative console secure port (Default 9043):	9043
HTTP transport port (Default 9080):	9080
HTTPS transport port (Default 9443):	9443
Bootstrap port (Default 2809):	2810
SOAP connector port (Default 8880):	8881
SAS SSL ServerAuth port (Default 9401):	9401
CSIV2 ServerAuth listener port (Default 9403):	9403
CSIV2 MultiAuth listener port (Default 9402):	9402
ORB listener port (Default 9100):	9102
High availability manager communication port (Default 9353):	9355
Service Integration Port (Default 7276):	7278
Service Integration Secure Port (Default 7286):	7288
Service Integration MQ Interoperability Port (Default 5558):	5561
Service Integration MQ Interoperability Secure Port (Default 5578):	5580

Figure 8-17 Profile port assignmens

If you are installing on Windows, you are presented with the screen in Figure 8-18, which allows you the choice to install the application server as a Windows service. If you check that option, you must supply a valid user and password that is used to run the service every time your system reboots.



Windows service definition

Choose whether to use a Windows service to run the WebSphere Application Server. Windows services can start and stop the WebSphere Application Server, and configure startup and recovery actions.

☑ Run the Application Server process as a Windows service.

● Log on as a local system account

○ Log on as a specified user account

User name: chukran

Password:

Startup type:

Manual

Figure 8-18 Profile Windows service definition

After entering all the required information, the profile creation wizard creates the profile and then offers to allow you to run the First Steps page in your browser, where you can start and stop the application server.

Profiles can also be created from the command line using the **wasprofile** command. The syntax of **wasprofile** for creation mode is:

```
wasprofile -create
-profileName profile_name
```

```
-profilePath fully_qualified_profile_path
-templatePath template_path
-nodeName node_name
[-cellName cell_name]
-hostName host_name
-server iSeries_server_name
[-startingPort starting_port | -portsFile filepath]
-winserviceCheck true | false
-winserviceAccountType specifieduser | localsystem
-winserviceUserName yourusername
-winservicePassword yourpassword
-winserviceStartupType manual | automatic | disabled
```

This is an example of using **`wasprofile`** to create a profile using a ports specification file. Note that this command would be entered on one line. The example is shown in multiple lines only for readability.

```
wasprofile
-create
-profileName node1
-profilePath
C:\ExpressV6\IBM\WebSphere\AppServer\profiles\node1
-templatePath
C:\ExpressV6\IBM\WebSphere\AppServer\profileTemplates\default
-nodeName node1
-cellName cell1
-hostName production1
-startingPort 20002
```

For a more complete overview of profiles, see the IBM Education Assistant for tutorials on using profiles at this address:

`ftp://ftp.software.ibm.com/software/eod/WAS_6-0/Install_Migration/index.html`

8.2.4 Silent installation

Silent installation is accomplished via command line options just as in V4 and V5. However, the contents of the response file that includes the installation responses has changed format. If you have customized a response file for a V4 or a V5 installation, you must discard that file and re-specify your option selections with the new V6 response file.

Each installer has its own response template file with options that are specific to that installer.

Application server response file

Response template files for application servers are located in the WebSphere Application Server subdirectory on the installation media. Figure 8-19 on page 111 shows the location of the response template file, responsefile.base.txt, for WebSphere Application Server. For WebSphere Application Server Express, the response template file is named responsefile.express.txt.

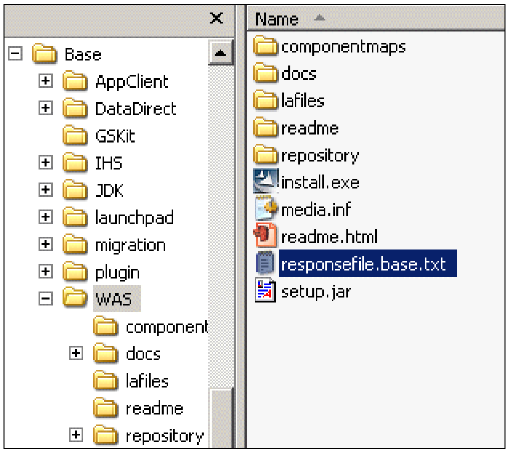


Figure 8-19 Silent installation response file for WebSphere Application Server

Figure 8-20 shows the location of the response template file for WebSphere Application Server Network Deployment. In this case, the response template file is named responsefile.nd.txt. Also note the files beginning with names responsefile.pct. If you want to create a profile as part of the silent install, you need to edit these files to specify profile settings.

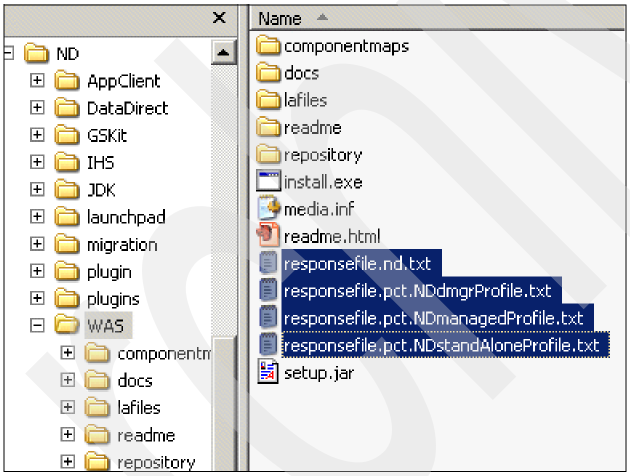


Figure 8-20 Silent installation response file for WebSphere Application Server Network Deployment

IBM HTTP Server response files

The response template file for the IBM HTTP Server is located in the ihs subdirectory of the installation media. Figure 8-21 on page 112 shows the location of the file named responsefile.txt.

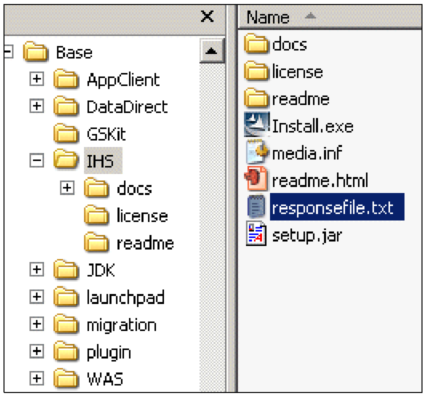


Figure 8-21 Silent installation response file for IBM HTTP Server

Web server plug-in response files

The response template file for the plug-in is located in the plugin subdirectory of the installation media. Figure 8-22 shows the location of the file named responsefile.txt.

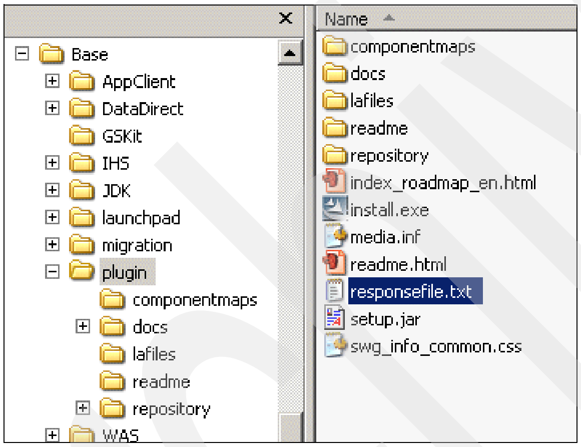


Figure 8-22 Silent installation response file for Web server plug-ins

Customizing response files

To customize a response file, first make a copy of the response template file that is on the installation media. When you edit the response file, refer to the commentary in the file that explains each setting. Be sure you change the setting that indicates you agree to the licensing terms, as shown in this example.

```
-W silentInstallLicenseAcceptance.value="true"
```

If you fail to change this setting from the default of `false`, the installer will not run since the interpretation of the false setting means that you do not agree to the license terms.

Each section of the response file has a description of the setting surrounded in comment lines. Follow the instructions in the commentary and customize the setting as necessary for your requirements. Following is an example section showing how the ports are specified with the default settings.

```
##########################################################################
#
# Node name
#
```

```
# Please select the node name for the Application Server. Node name
# under one cell has to be unique.
#
# Replace YOUR_NODE_NAME with the actual node name. Comment the line to use
# the default value.
#
-W nodehostandcellnamepanelInstallWizardBean.nodeName="YOUR_NODE_NAME"
```

In this example, you could not use the default. You must replace the phrase `YOUR_NODE_NAME` with the actual node name you wish to assign.

Running the installer

Once you have customized the response files, you run the installer from the command line similar to this example. In this case the file C:\temp\myoptionsfile.txt is your customized response file.

```
install -options "C:\temp\myoptionsfile.txt" -silent
```

8.3 Administrative console

The administrative console is accessed from your Internet browser as is the case for any of the WebSphere Application Server Version 5 products as well as WebSphere Application Server Version 4 Advanced Edition Single Server. Server configurations are changed by navigating to the appropriate section in the navigation bar. You then change values in a browser form and submit the values by clicking the **OK** button. While you are editing the configuration, a temporary copy is saved while you work. When all the changes are complete, save the configuration. This commits your changes to the config directory in the profile of the server that you are editing.

8.3.1 Console address

The default administrative console address has been changed to:

```
http://localhost:9060/ibm/console
```

The default port has been changed to 9060 to avoid conflicting with the 9090 port on the AIX operating system. Note that 9060 is a default port assignment, which can be changed when you create a profile for the server that runs the administrative console application. See “Creating profiles” on page 106 for more information about creating profiles.

8.3.2 Messaging components

The administrative console has two sections in the left navigation area that pertain to asynchronous messaging. Figure 8-23 on page 114 shows the JMS resources section, which is similar to that section for V5. WebSphere MQ and Generic refer to external JMS providers and have the identical configuration semantics as in V5.

Default messaging refers to the embedded JMS messaging provider that reflects the new messaging design for V6. See “Asynchronous messaging architecture” on page 98 for an overview of the new messaging design.

V5 default messaging refers to the V5 compatibility interface to messaging. This interface has two purposes. The first purpose is to represent the JMS resources for a V5 node running in a V6 cell. Since a V5 node implements messaging with V5 components, the semantics of configuration are identical to those of V5. The second purpose is to represent the JMS resources that have been migrated from a V5 node to a V6 node using the automatic migration utilities. For this second purpose, the administrative console shows you an interface that looks similar to how V5 is configured, but the actual messaging implementation is that of V6. For both of these cases, the application that uses these JMS resources requires no changes.

See 12.3, “Automatic migration: Application Server V5 to Express V6” on page 191 for an example of migrating a V5 node with an application that uses JMS default resources.

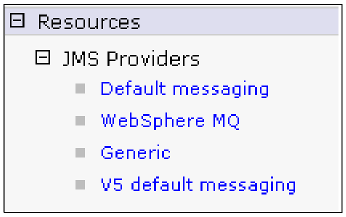


Figure 8-23 JMS resource provider

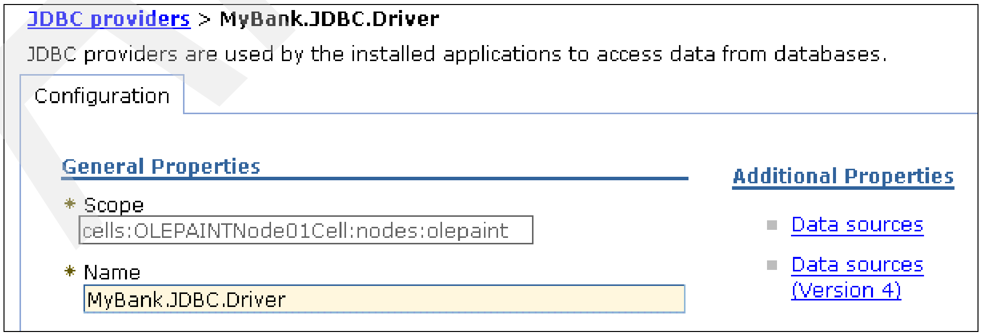
8.3.3 Data sources

Data source configuration is nested under the JDBC Provider section of the left navigation area of the administrative console. Figure 8-24 shows that data source configuration has two different types of data source. The topmost, simply called *Data Sources*, which we call a default data source, uses a JCA architecture to manage connections to the database that backs the data source. This architecture was introduced for V5 and continues to be configured identically in V6. The second type of data source, identified as *Data Sources (Version 4),* uses the V4 Connection Managers architecture.

The choice of which data source to configure is determined by your application. If your application uses the EJB 1.1 programming interfaces, you should configure a V4 data source. If your application uses the EJB 2.0 or EJB 2.1 programming interfaces, you should configure a default data source.

The automatic migration utilities are aware of these two types of data source and migrate to the right type. For the case of migrating from a V4 system, a V4 type data source is created in the V6 profile. For the case of migrating from V5, the same type of data source is migrated to the V6 profile.

See Figure 12-6 on page 191 for an example of migrating a V4 data source and Figure 12-12 on page 196 for an example of migrating a default data source.


JDBC providers > MyBank.JDBC.Driver

JDBC providers are used by the installed applications to access data from databases.

* Scope
cells:OLEPAINTNode01Cell:nodes:olepaint

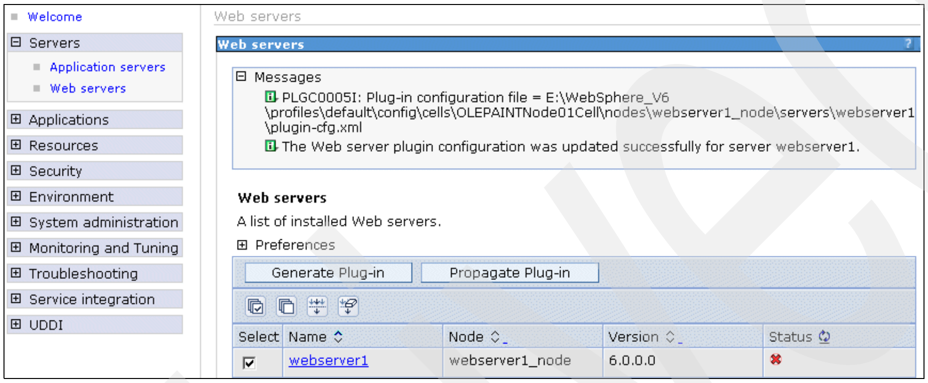
* Name
MyBank.JDBC.Driver

Figure 8-24 JDBC Provider

8.3.4 Plugin-Cfg generation

Plug-in configuration file generation for previous versions is controlled either from the administrative console or with the command **GenPluginCfg**. The plug-in configuration file is then manually transported to the server.

The administrative console method of regenerating the plug-in configuration file has changed for V6. The control button to regenerate the plug-in configuration file is located under the Web server instance. Figure 8-25 shows a Web server that is found under the Servers section. To regenerate the plug-in configuration file, select the Web server, in this case webserver1, and click the **Generate Plug-in** button. The message at the top indicates successful file generation and also indicates the location of the file that was generated.



PLGC0005I: Plug-in configuration file = E:\WebSphere_V6 \profiles\default\config\cells\OLEPAINTNode01Cell\nodes\webserver1_node\servers\webserver1 \plugin-cfg.xml

The Web server plugin configuration was updated successfully for server webserver1.

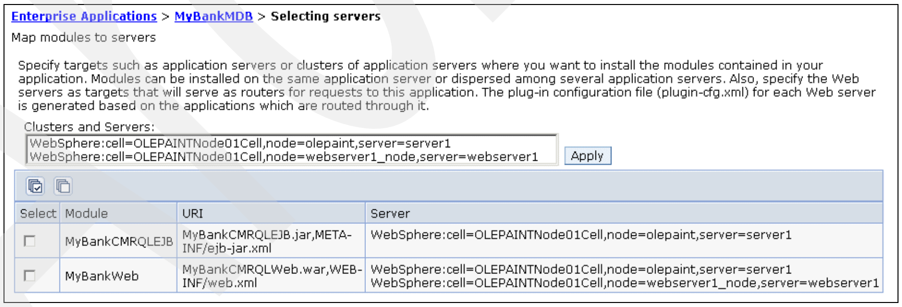
Web servers

A list of installed Web servers.

Select	Name	Node	Version	Status
☑	webserver1	webserver1_node	6.0.0.0	✖

Figure 8-25 Web server configuration

Generating the plug-in configuration file from the Web server instance has different semantics than that of previous versions. Previous versions select all applications mapped to any application server. V6 has a different selection semantic in that only applications that are mapped to that Web server are selected for mapping into that file. Figure 8-26 shows an example of an application that is mapped to both webserver1 and server1. This application must be mapped to webserver1 or it is not included in the plug-in configuration file when it is generated by the Generate Plug-In button.



Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that will serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated based on the applications which are routed through it.

Clusters and Servers:
WebSphere:cell=OLEPAINTNode01Cell,node=olepaint,server=server1
WebSphere:cell=OLEPAINTNode01Cell,node=webserver1_node,server=webserver1

Select	Module	URI	Server
□	MyBankCMRQLEJB	MyBankCMRQLEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=OLEPAINTNode01Cell,node=olepaint,server=server1
□	MyBankWeb	MyBankCMRQLWeb.war,WEB-INF/web.xml	WebSphere:cell=OLEPAINTNode01Cell,node=olepaint,server=server1 WebSphere:cell=OLEPAINTNode01Cell,node=webserver1_node,server=webserver1

Figure 8-26 Selecting servers for Web servers

8.4 How to perform basic functions

This section describes some basic system management tasks that you will perform frequently.

Application installation

The installation for WebSphere Application Server Version 6 is started by the launchpad command located on the top level directory of the install media. Many operating systems automatically start the launchpad when you insert the media into the external media drive. If the launchpad is not started automatically, start it from the command line or a graphical files system viewer. See 8.2.1, “Launchpad” on page 101 for more details on using launchpad.

Starting servers

A stand-alone node is started from the command line with the **startServer** command. The most common syntax for **startServer** is:

```
startServer server_name [-profileName profile]
```

where the profile specification is optional. If `profile` is omitted, then the default profile is assumed. The profile can be implicitly selected by running the command from the bin directory of the profile to be selected.

Here are some examples:

- Start the server named server1 in the default profile:

  ```
  cd /d c:\Websphere\bin
  startServer server1
  ```

- Start the server division1 in the profile named sales:

  ```
  startServer division1 -profileName sales
  ```

- Start the server named server1 in the profile under the profiles directory:

  ```
  cd /d c:\Websphere\profiles\policies\bin
  startServer server1
  ```

Stopping servers

Application servers are stopped with the **stopServer** command. The most common syntax for **stopServer** is:

```
stopServer server_name [-profileName profile]
```

The profile specification is optional, and uses the same profile selection semantics as the **startServer** command described previously.

Some examples of stopServer are as follows:

- Stop the server named server1 in the default profile:

  ```
  cd /d c:\Websphere\bin
  stopServer server1
  ```

- Stop the server division1 in the profile named sales:

  ```
  stopServer division1 -profileName sales
  ```

- Stop the server named server1 in the policies profile under the profiles directory:

  ```
  cd /d c:\Websphere\profiles\policies\bin
  stopServer server1
  ```

Starting a deployment manager

A deployment manager is started with the **startManager** command. The most common syntax is:

```
startManager [-profileName profile]
```

As with all commands that accept a profile name, the name is optional and defaults to the default profile. The selected profile must be a deployment manager type. Some usage examples follow.

- Start the deployment manager in the profile named dmgr1:

  ```
  startManager -profileName dmgr1
  ```

- Start the deployment manager in the dmgr2 profile under the profiles directory:

  ```
  cd /d c:\Websphere\profiles\dmgr2\bin
  startManager
  ```

Stopping a deployment manager

A deployment manager is stopped with the **stopManager** command. The most common syntax is:

```
stopManager [-profileName profile]
```

Some usage examples are shown.

- Stop the deployment manager in the profile named dmgr1:

  ```
  stopManager -profileName dmgr1
  ```

- Stop the deployment manager in the dmgr2 profile under the profiles directory:

  ```
  cd /d c:\Websphere\profiles\dmgr2\bin
  stopManager
  ```

Starting a node agent

The node agent is created when an application server joins a Network Deployment cell. See “Joining a Network Deployment cell” on page 118 for details on how to join a server into a cell.

The node agent must be started before any application servers in the node can start. The most common syntax is:

```
startNode [-profileName profile]
```

As with all commands that accept a profile name, the name is optional and defaults to the default profile. The selected profile must be a deployment manager type. Some usage examples are shown.

- Start the node agent in the profile named node1:

  ```
  startNode -profileName node1
  ```

- Start the node agent in the node1 profile under the profiles directory:

  ```
  cd /d c:\Websphere\profiles\node1\bin
  startNode
  ```

Stopping a node agent

A node agent is started with the **stopNode** command. The most common syntax is:

```
stopNode [-profileName profile]
```

As with all commands that accept a profile name, the name is optional and defaults to the default profile. The selected profile must be a federated node profile. Some usage examples are shown.

- Start the node agent in the profile named node1:

  ```
  stopNode -profileName node1
  ```

- Start the node agent in the node1 profile under the profiles directory:

```
cd /d c:\Websphere\profiles\node1\bin
stopNode
```

Joining a Network Deployment cell

A Network Deployment cell is created when the deployment manager profile is created. This cell, which consists of only a deployment manager, would be considered an empty cell. The cell is considered empty because there are no nodes in the cell yet.

A node joins a cell by running the **addNode** command. The act of joining a cell is also called *federating* a node into a cell. The **addNode** command operates on an application server profile and contacts a deployment manager and asks it to join the node to the cell.

The most common syntax of the **addNode** command is:

```
addNode dmgr_host [dmgr_port] [-profileName profile]
```

The deployment manager host name is the only required argument. The second optional argument is the port that the deployment manager listens on. As with all commands that select a profile, the profile name is optional and defaults to the default profile. Here are some examples of usage of **addNode**:

- Add the node defined by the default profile to the cell managed by dmanager1:

  ```
  addNode dmanager1
  ```

- Add the node in the profile node1 to the cell managed by dmanager1, which is listening on port 8885:

  ```
  addNode dmanager1 8885 -profileName node1
  ```

After **addNode** is complete, the node is part of the cell. Assuming the cell was empty at the start, the cell now has a single node member. The node now consists of two entities, a node agent and an application server. The node agent is created and started automatically by **addNode**. The application server that is created at the time the profile is created still exists, but it is not automatically started. See “Starting servers” on page 116 for examples of how to start application servers.

An alternate way to join a cell is to create a custom profile. The profile creation wizard, upon receiving your selection of a custom profile type, prompts you to specify a deployment manager host name and port number. The deployment manager must be started before the custom profile creation can complete. Upon completion, the cell has a single node member. However, this node only has a node agent. There are no application servers defined for this type of profile. Application servers would then be created through the administrative console for the deployment manager.

8.5 Changes in default settings

There are a few server or application settings that have changed either semantics or a default value, such that applications may not behave in an identical manner from one version of WebSphere Application Server to a succeeding version. The following settings changed in the transition to V5. What this means is that an application running on V4 that is migrated to V6 may not run correctly unless these settings are verified for needed changes. Any applications being migrated from V5 to V6 do not need any such verification because the V6 settings are identical to those for V5.

8.5.1 Class loader

Class loader semantics have changed in V6 compared to those in V4. There are several possibilities for symptoms:

- Your application has utility classes that duplicate classes contained in the WebSphere system classes directories. Your classes fail to override the system classes.
- Your application contains classes such that each module may contain classes that conflict with classes in another module.

Version 4 has three attributes of classloading that work differently from V6:

1. In V4, the application classloader for Web modules precedes the system class loaders such that classes in the Web module are searched first. This default behavior can be changed by setting the system property com.ibm.ws.classloader.warDelegationMode to the value `false`.

   In V6, and also V5, a Web module classloader is searched last with respect to the system class loaders. This default behavior is changed by changing the setting class loader mode to `PARENT_LAST` using the administrative console.

   > **Note:** Version 4 refers to the term *classloader delegation mode*, while V5 and V6 refer to the same term as simply *classloader mode*.

   EJB modules in V4, and also V5 and V6, have their classloader searched last as a default. V4 uses the system property com.ibm.ws.classloader.ejbDelegationMode to change the default behavior. V5 and V6 use the classloader delegation mode setting `PARENT_LAST` to change the behavior such that the EJB module classloader is searched first.

   V6 classloader mode is changed in the administrative console. Classloader mode may be changed at three levels:

   – Server
   – Enterprise application
   – Web module

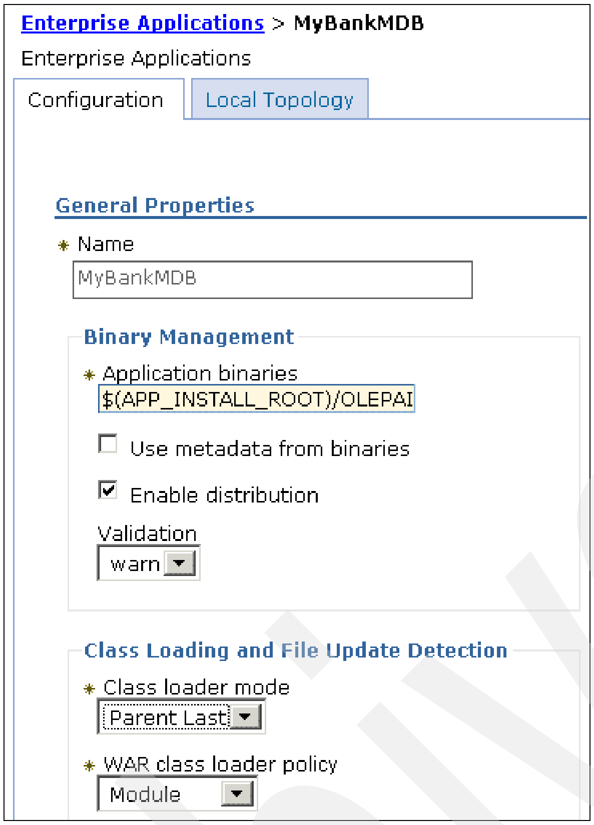
   We recommend that you do not change the classloader mode at the server level, since this affects all the applications deployed on that server. Our recommendation is to change classloader mode at the enterprise application level, which affects all the EJB modules. If any of the Web modules need a different classloader mode, then you would change each Web module individually.

   Figure 8-27 on page 120 shows changing the classloader mode for the enterprise application MyBankMDB.

   Figure 8-28 on page 120 shows changing the classloader mode for a Web module.

   You can read more about the topic of class loaders and how to change the behavior in the WebSphere Application Server InfoCenter. Refer to this article: *Class Loaders*, found at:

   http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/crun_classload.html



Enterprise Applications > MyBankMDB

Enterprise Applications

Configuration | Local Topology

General Properties

* Name

MyBankMDB

Binary Management

* Application binaries

$(APP_INSTALL_ROOT)/OLEPAI

□ Use metadata from binaries

☑ Enable distribution

Validation

warn

Class Loading and File Update Detection

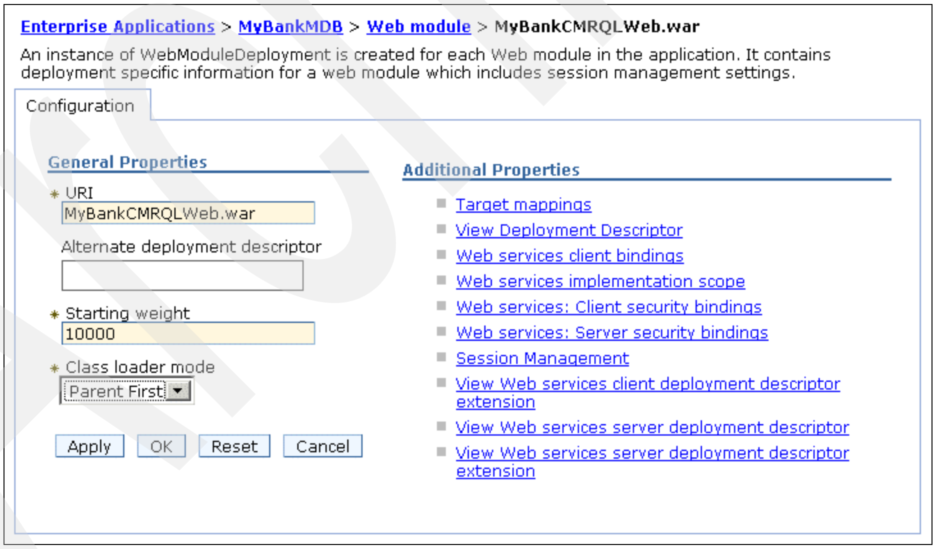
* Class loader mode

Parent Last

* WAR class loader policy

Module

Figure 8-27 Changing class loader mode and policy for an application



Enterprise Applications > MyBankMDB > Web module > MyBankCMRQLWeb.war

An instance of WebModuleDeployment is created for each Web module in the application. It contains deployment specific information for a web module which includes session management settings.

Configuration

General Properties

* URI

MyBankCMRQLWeb.war

Alternate deployment descriptor

* Starting weight

10000

* Class loader mode

Parent First

Additional Properties

- Target mappings
- View Deployment Descriptor
- Web services client bindings
- Web services implementation scope
- Web services: Client security bindings
- Web services: Server security bindings
- Session Management
- View Web services client deployment descriptor extension
- View Web services server deployment descriptor
- View Web services server deployment descriptor extension

Figure 8-28 Changing class loader mode for a Web module

2. As the default, the application classloader has a module visibility that allows modules to have their own class loaders, thus eliminating class conflicts. V4 specifies module visibility settings at the server level only. By contrast, V5 and V6 specify similar settings at these levels:

a. Server settings allow you to specify the class loader policy for applications, which can be one of these values:

   i. *Single* means that a single classloader is used for every application on that server. This setting provides the most class sharing, but it also offers opportunity for class conflicts.

   ii. *Multiple* means that multiple class loaders are used. Each application has its own classloader. This value is the default.

b. Each application can be configured to specify the WAR, or Web module, classloader policy, which can be one of these values:

   i. *Module* means that each Web module has its own class loader. This value is the default.

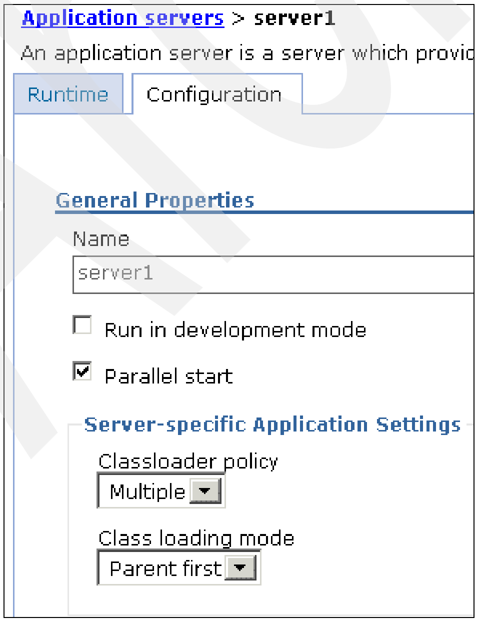
   ii. *Application* means that every Web module uses the application class loader. Thus, a single application class loader loads every module in the application, including Web modules, EJB modules, and dependency JAR files.

Table 8-3 shows a summary of how to convert from V4 settings to V6 settings.

Table 8-3 MIgrating visibility settings from V4 to V6

Version 4 module visibility	Version 6 application class loader policy	Version 6 Web module class loader policy
Server	Single	Application
Compatibility	Single	Module
Application	Multiple	Application
Module	Multiple	Module
J2EE	Multiple	Module

Figure 8-27 on page 120 shows an example of setting the WAR class loader policy for the application MyBankMDB. Figure 8-29 shows an example of setting the classloader policy for the server server1.



Application servers > server1

An application server is a server which provic

General Properties

Name

server1

☐ Run in development mode

☑ Parallel start

Server-specific Application Settings

Classloader policy

Multiple

Class loading mode

Parent first

Figure 8-29 Changing class loader policy for a server

3. The application extensions classpath location from V4 has been dropped in favor of a more flexible scheme called *application shared libraries*. V4 configures a special directory under the installation root directory, lib/ext, as a directory where you can place class jar files and have every application class loader search this classpath. The system classloaders do not search this path.

   The shared library method that is used in V5 and V6 works similarly in that a single directory, or even a list of directories, can contain application classes that are not visible to the system class loaders. The advantage over the applications extension classpath of V4 is that the configuration is more flexible.

   a. Each shared library can be restricted to a subset of applications rather than the entire set of applications deployed to a server.

   b. Each shared library can be configured to point to an arbitrary list of directories on the system.

   Figure 8-30 shows that you navigate to the shared library definition screen under the Environment category. Figure 8-31 on page 123 shows creating the shared library definition. For the classpath field, specify a list of directories, all on a separate line. Do not use any punctuation such as semicolons. Each directory will contain one or more JAR files.

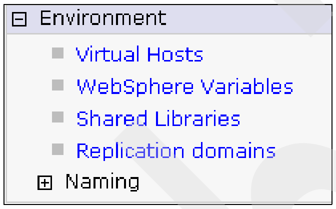
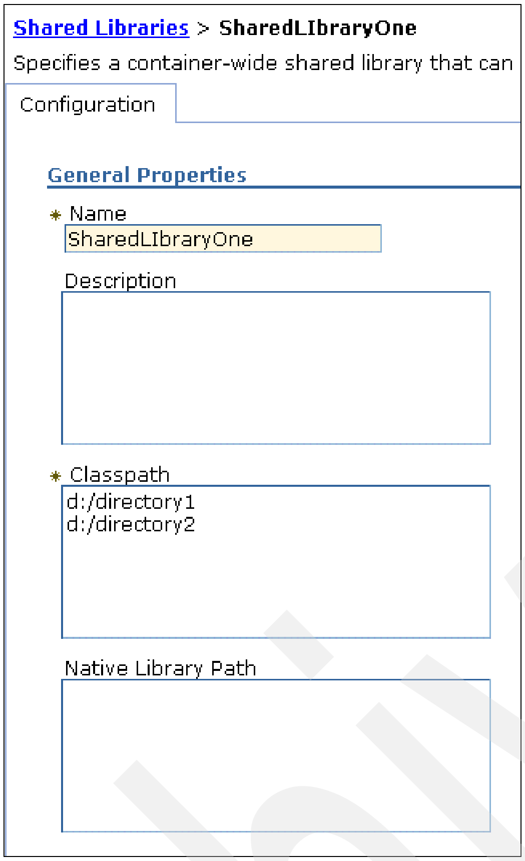


Figure 8-30 Navigating to shared library creation



Shared Libraries > SharedLIbraryOne

Specifies a container-wide shared library that can

Configuration

General Properties

* Name: SharedLIbraryOne

Description:

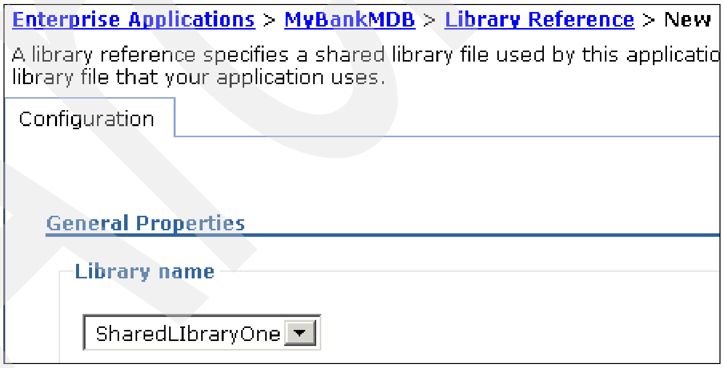
* Classpath:

```
d:/directory1
d:/directory2
```

Native Library Path:

Figure 8-31   Creating a shared library definition

Once you create a shared library definition, you can associate the shared library with one or more applications. Figure 8-32 shows specifying the shared library with the MyBankMDB application.



Enterprise Applications > MyBankMDB > Library Reference > New

A library reference specifies a shared library file used by this applicatio
library file that your application uses.

Configuration

General Properties

Library name: SharedLIbraryOne

Figure 8-32   Associating a shared library with an application

8.5.2 Transaction isolation level

Applications that perform transactions to databases may have issues with the transaction isolation level when migrating from V4 to V6. Applications that perform transactions in a Web module may encounter changes in how the application behaves with respect to the database. The issue is that the isolation level for transactions in a Web module are defaulted to TRANSACTION_READ_COMMITTED for V4. When the application is deployed to V6, the default value for the isolation level is TRANSACTION_REPEATABLE_READ for all supported databases except for Oracle. The default isolation level for Oracle is TRANSACTION_READ_COMMITTED. The result is that the application may experience a loss of performance due to database lock contention. In the worst case, the application may experience database deadlocks. See this article for more information about isolation level settings for both Web modules and EJB modules: *Isolation level and resource reference*, found at:

`http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cdat_isolevel.html`

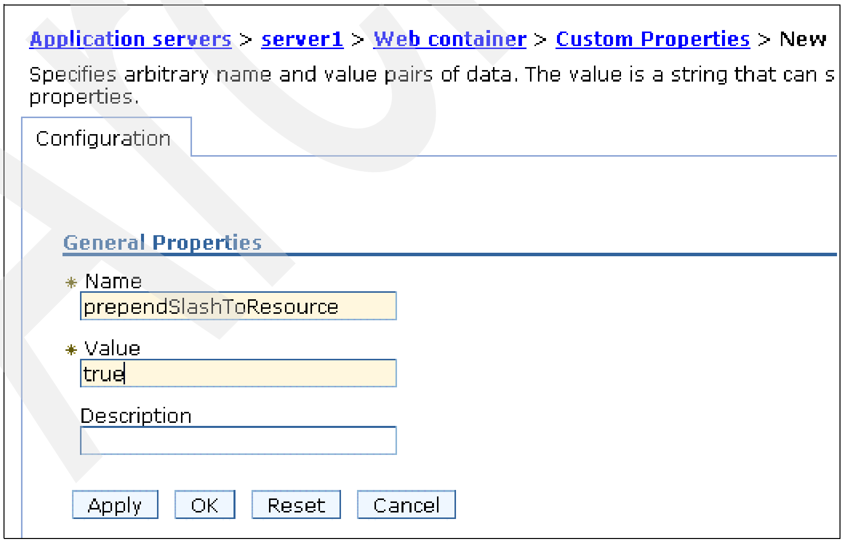
The solution requires that you set the isolation level to the value that your application expects by specifying the isolation level in a data source resource reference for the data sources that are used by the application. You can edit resource references by using the Application Assembly Toolkit that is included with the V6 product, or you can use one of the Rational Developer products. See this article for more information about how to edit resource references: *Creating or changing a resource reference*, found at:

`http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tatk_crtresref.html`

8.5.3 getResource path syntax

WebSphere Application Server Version 6 changed the manner in which it interprets the resource path for the **`getResource`** and **`getResourceAsStream`** methods. See "Slash character on the getResource path" on page 31 for background on why this behavior has changed.

The default behavior is such that many applications may receive a MalformedURLException on Version 6, whereas these applications did not get the exception on Version 5. This default behavior was changed to pass the J2EE compliance tests. You can change the behavior back to the Version 5 behavior by setting the prependSlashToResource property for the Web container on the application server. Figure 8-33 shows an example of how to set this property using the administrative console. After adding the property, remember to save the configuration and restart the server.



Application servers > server1 > Web container > Custom Properties > New

Specifies arbitrary name and value pairs of data. The value is a string that can s properties.

General Properties

* Name: prependSlashToResource

* Value: true

Description:

Figure 8-33 Setting custom property for Web container

8.6 Automatic migration utilities

This section describes utilities that are shipped with the WebSphere Application Server that can be used for migration of system configuration and applications from a previous version of WebSphere Application Server to the current one.

8.6.1 Command line utilities introduction

There are four commands shipped with WebSphere Application Server Version 6 to assist in migration of configuration and applications between WebSphere Application Server versions.

- **`clientUpgrade`** - This command is used by developers or administrators to upgrade resources in client jar files.
- **`convertScriptCompatibility`** - This command is used by administrators to convert their configuration from a mode that supports backward compatibility of 5.x administration scripts to a mode that is fully 6.0.
- **`WASPreUpgrade`** - This command is used to save WebSphere configuration and application information into a migration-specific backup directory.
- **`WASPostUpgrade`** - This command is used to process the migration-specific backup directory that was created by **`WASPreUpgrade`** and import that configuration and enterprise applications into the current WebSphere Application Server environment.

8.6.2 Configuration migration

There are two utilities that are shipped with the WebSphere Application Server runtime and used in conjunction with each other to migrate the administration configuration from a previous version of WebSphere Application Server to WebSphere Application Server Version 6. The overall migration process is to install the V6.0 product, back up the previous version configuration and necessary files, and import the configuration from the previous version into WebSphere Application Server Version 6. The migration wizard can call these utilities or you can call them manually.

These commands, as introduced previously, are:

- **`WASPreUpgrade`**
- **`WASPostUpgrade`**

Their participation in the overall migration is shown in Figure 8-34 on page 126.

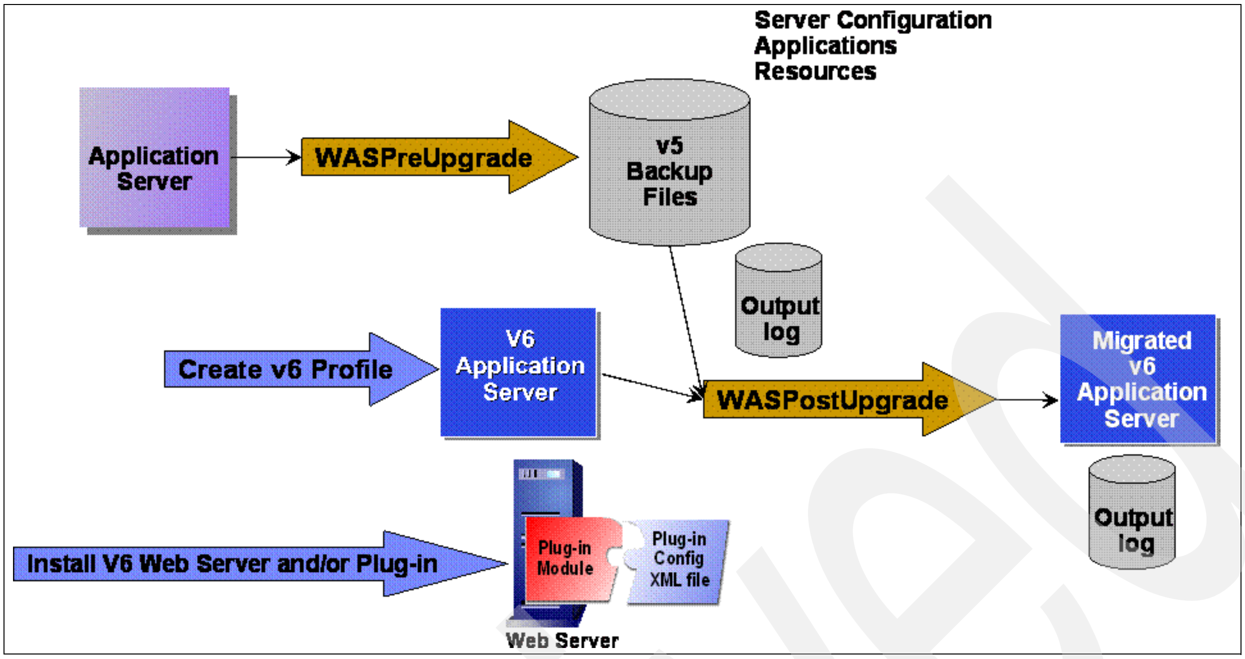


Figure 8-34 General migration process steps

WASPreUpgrade command

The **WASPreUpgrade** command is used to save the configuration from the previous version of WebSphere Application Server into a backup directory. This backup directory is later used by the **WASPostUpgrade** command to import this configuration into WebSphere Application Server Version 6. The **WASPreUpgrade** command provides the status both to the screen and to log files in the backup directory. ASCII log file names start with the text `WASPreUpgrade` and include a date timestamp.

The following files are saved in the backup directory:

- For V4.0.x
  - bin/setupCmdLine.sh (or bin/setupCmdLine.bat for Windows platforms)
  - classes (not saved for iSeries)
  - config (V4.0.x Advanced Single Server Edition only)
  - installableApps
  - installedApps
  - installedConnectors (V4.x Advanced Edition only)
  - properties
- For V5.0.x and 5.1.x
  - classes (not saved for iSeries)
  - config
  - installableApps
  - installedApps
  - properties
  - Any instances created by using the **wsinstance** command (For iSeries, **WASPreUpgrade** must be invoked for each instance.)

The **WASPreUpgrade** command exports the existing application server configuration from the repository when migrating from V4.x Advanced Edition. If you are migrating V4.x Advanced Edition, the administrative server of the existing environment must be running.

This command is also available in the migration directory on the Installation CD. This directory can be used for scenarios where you need to run **`WASPreUpgrade`** without first installing WebSphere Application Server Version 6.

How to run it

```
WASPreUpgrade backupDirectory currentWebSphereDirectory
         [administrationNodeName]
         [-import xmiDataFile]
         [-nameServiceHost host_name [-nameServicePort port_number]]
         [-traceString traceSpec [-traceFile fileName]]
```

The first two arguments are required and positional. The third argument is required and positional only when upgrading from WebSphere Application Server Version 4 Advanced Edition.

Supported parameters include:

- `backupDirectory`

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

- `currentWebSphereDirectory`

  Required positional parameter. The name of the installation root directory for the current V4.0.x, V5.0.x or V5.1.x installation.

- `administrationNodeName`

  Optional positional parameter. The 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`** command calls the **`XMLConfig`** command using this parameter. This is a required parameter only when upgrading from WebSphere Application Server Version 4 Advanced Edition.

- `-nameServiceHost -nameServicePort`

  When specified, the **`WASPreUpgrade`** command passes these optional parameters to the **`XMLConfig`** command. Use them to override the default host name and port number used by the **`XMLConfig`** command. This is only applicable when upgrading from WebSphere Application Server Version 4 Advanced Edition.

- `-import`

  The name of the XML Metadata Interchange (XMI) configuration file to process. This is optional when upgrading from WebSphere Application Server Version 4 Advanced Edition Single Server or WebSphere Application Server Version 4 Advanced Edition because the program uses the V4.0 config\server-cfg.xml file by default. When migrating from a WebSphere Application Server Version 4 Advanced Edition configuration that uses anything other than the default server-cfg.xml file, you must use the **`-import`** option along with the path to the configuration file. You also must use the **`-import`** and path option when running **`WASPostUpgrade`**, to point to the non-default xml configuration file in the directory created by **`WASPreUpgrade`**.

- `-traceString traceSpec -traceFile fileName`

  Optional parameters to gather trace information for IBM service personnel. Specify a traceSpec of `"*=all=enabled"` (with quotation marks) to gather all trace information.

WASPostUpgrade command

The **WASPostUpgrade** command imports the V4.0.x, V5.0.x, or V5.1.x configuration data. This command uses information created by the **WASPreUpgrade** command to restore the previous configuration to a V6 installation.

This command can be run multiple times with different configuration files to incrementally update the V6.0 configuration.

How to run it

```
WASPostUpgrade backupDirectory
          [-oldProfile < oldProfileName >]
          [-profileName < profileName >]
          [-import < xmiDataFile >]
          [-portBlock < portStartingBlock >]
          [-backupConfig < true | false >]
          [-replacePorts < true | false >]
          [-includeApps < true | false >]
          [-scriptCompatibility < true | false >]
          [-connectionTimeout < timeoutInMinutes >]
          [-substitute "key1=value1[;key2=value2;[...]]"]
          [[-traceString traceSpec [-traceFile fileName]]
```

The first argument is required. Supported parameters include:

- `backupDirectory`

  Required parameter. The name of the directory in which the **WASPreUpgrade** command stores the saved configuration and files, and from which the **WASPostUpgrade** command reads the configuration and files. The **WASPreUpgrade** command creates this directory if it does not already exist.

- `-oldProfile <profile name>`

  This optional parameter is used for migrating instances from previous WebSphere Application Server versions. In WebSphere Application Server Version 5, instance names are defined by "`instanceName`" and "`hostName`". These names are concatenated together to form a unique name. The instance must already exist in the migration backup directory before executing this command. Prior to this release, **`-instance`** was an undocumented parameter used by iSeries to support their instance architecture; this support continues as is. For iSeries, if not specified, the default instance is used (called "`default`").

- `-profileName <profile name>`

  This optional parameter is used for migrating to profiles in this release. This profile must have already been created by the user before calling **WASPostUpgrade**. If not specified, the default profile is used. An error is reported if no default is found.

- `-import < xmiDataFile >`

  An optional name of the XML Metadata Interchange (XMI) configuration file to process. Specify the WebSphere Application Server Version 4 Advanced Edition Single Server configuration file or an XML configuration data file that the **XMLConfig** command created. When migrating from a WebSphere Application Server Version 4 Advanced Edition Single Server configuration that uses anything other than the default server-cfg.xml file, you must use the **-import** option along with the path to the non-default XML configuration file in the directory created by **WASPreUpgrade**. If not specified, the program uses the default XML configuration file in the backup directory.

- `-portBlock <startingPortNumber>`

  This optional parameter is used to specify the starting seed value to be used when creating new ports.

- `[-backupConfig <true|false>]`

  An optional parameter used to back up the existing configuration of the current profile before adding the saved configuration from the earlier release to the current instance. The default is true, to use the backupConfig command to save a copy of the current configuration into the profile_name/temp directory. Use the restoreConfig command to restore that configuration as required.

- `-replacePorts <false|true>`

  An optional parameter used to define how to migrate virtual host and server transport ports. The default for migrations from V4.0.x is `false` (do not replace default port definitions); the default for migrations from V5.x is `true` (do replace default port definitions). Migrating adds configuration data from the previous version to the existing data in the V6.0 configuration. In some cases, existing port definitions from the earlier release are carefully set to avoid port conflicts with other products. In such cases, it is likely that you would want to migrate the settings into V6.0. Use the **`-replacePorts`** parameter to totally replace settings in the V6.0 environment with the settings from the previous version. Select `true` to replace all virtual host alias port settings during migration. If migrating from V5.0.x or later, transport settings in existing servers are replaced by the settings from the previous version.

- `-includeApps <true|false>`

  This optional parameter is used to specify whether to only migrate the configuration or to include the user enterprise applications as part of the migration. The default setting is true. System applications are migrated irrespective of this setting. Sample applications are typically not migrated. The only exception is during ND Phase1 processing.

- `-scriptCompatibility <-true|false>`

  This optional parameter is used if migration should create Transport and ProcessDef definitions in the configuration instead of Channels and ProcessDefs. This would be used if customers have existing wsadmin scripts or programs that used SM configuration APIs to create or modify Transport or ProcessDef definitions. The default is `true`.

- `-connectionTimeout` *`timeoutInMinutes`*

  This is an optional parameter to be used if a SOAP/RMI time-out occurs when performing a migration of a managed node. This parameter is specified in minutes (the default value is 10). It is possible to run into a SOAP/RMI time-out issue when migrating a very large or very complex configuration; if this happens, you should increase the amount of time before a SOAP/RMI connection time-out occurs and run migration again.

- `-substitute "key1=value1[;key2=value2;[...]]"`

  An optional argument passed to the **`XMLConfig`** command. Specify values for security variables to be substituted (for example, `-substitute "NODE_NAME=admin_node;APP_SERVER=default_server"`). In the input XML data file, each key should appear 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, be sure to add an escape character to each dollar sign ($) within the substitution string (for example, `\$semiColon\$`). This parameter is applicable for configurations saved from Versions 4.0.x Advanced Edition.

- `-traceString -traceFile`

  Optional parameters to gather trace information for IBM service personnel. Specify a traceSpec of `"*=all=enabled"` (with quotation marks) to gather all trace information.

After migration

It is always a good idea to review the results of migration by reviewing the migration logs in the backup and V6.0 profile logs directory. Some of the configuration may need to be modified based on the results of the migration.

8.6.3 Graphical wizard for configuration migration

A separate migration wizard for WebSphere Distributed platforms is also provided that is a Graphical User Interface (GUI) to the **`WASPreUpgrade`** and **`WASPostUpgrade`** commands.

In previous versions of WebSphere Application Server, the migration wizard is integrated with the installation process for WebSphere Distributed platforms. This approach has some distinct advantages and disadvantages. The advantages include ease of use and integration with the silent installation support. However, the disadvantages include the inability to apply automatic migration utility fixes before migrating and a combination of skill sets required for the installation process.

The separate migration wizard approach eliminates the disadvantages and continues to provide a easy to use interface. The ability to integrate with silent install can be replaced by calling the **`WASPreUpgrade`** and **`WASPostUpgrade`** commands directly as part of the installation tasks.

The mIgration wizard does not provide interfaces for all the options that are available by using the **`WASPostUpgrade`** command directly. However, it can be used for most migration tasks.

The migration wizard can be launched from the First steps window shown in Figure 8-35.

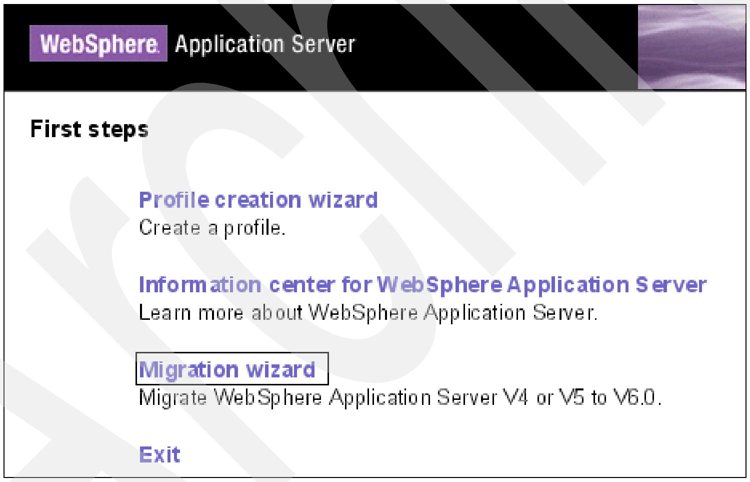
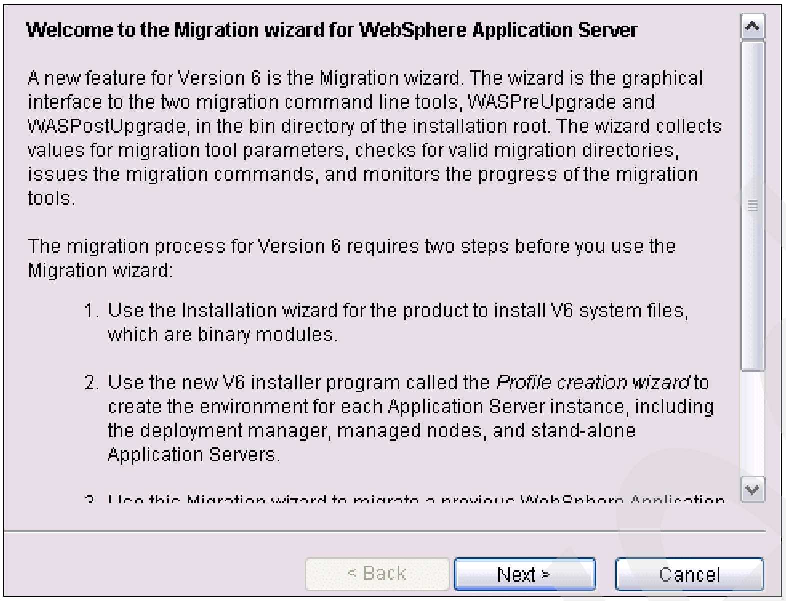

WebSphere Application Server
First steps
Profile creation wizard
Create a profile.
Information center for WebSphere Application Server
Learn more about WebSphere Application Server.
Migration wizard
Migrate WebSphere Application Server V4 or V5 to V6.0.
Exit

Figure 8-35 First steps

Figure 8-36 on page 131 shows the initial screen of the migration wizard. The initial screen lists an overview of the installation and migration steps for WebSphere Application Server Version 6.



Welcome to the Migration wizard for WebSphere Application Server

A new feature for Version 6 is the Migration wizard. The wizard is the graphical interface to the two migration command line tools, WASPreUpgrade and WASPostUpgrade, in the bin directory of the installation root. The wizard collects values for migration tool parameters, checks for valid migration directories, issues the migration commands, and monitors the progress of the migration tools.

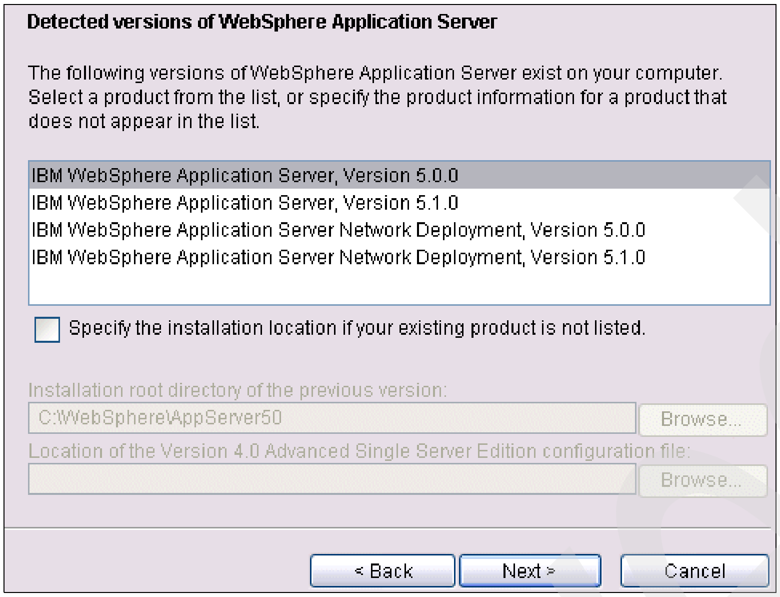
The migration process for Version 6 requires two steps before you use the Migration wizard:

1. Use the Installation wizard for the product to install V6 system files, which are binary modules.
2. Use the new V6 installer program called the *Profile creation wizard* to create the environment for each Application Server instance, including the deployment manager, managed nodes, and stand-alone Application Servers.
3. Use this Migration wizard to migrate a previous WebSphere Application

Figure 8-36 Initial screen of migration wizard

1. Install WebSphere Application Server Version 6 using the installation wizard. Note that for Base and Express installations, a default profile is created during this step.
2. Use the profile creation wizard, which is also located in the First steps window, to create any profiles that are required. For WebSphere Application Server Network Deployment no default profile is created during installation. For this environment, a *deployment manager* profile must be created. For each managed node in the existing 5.0 or 5.1 cell either a *stand-alone* profile or a *custom* profile (that is not federated during profile creation) must be created using the profile creation wizard. See 12.5.3, “V6 installation preparation” on page 217 for more details on specific node name or cell name requirements for specific migration scenarios.
3. Run the Migration wizard using one of the profiles created in the previous steps.

Figure 8-37 on page 132 shows a list of existing WebSphere Application Server Version 4.0.x, 5.0.x, and 5.1 installations that can be migrated. You can either select one from the list or supply a directory name of a different one if, for some reason, it was not detected and entered into the selection list. The last entry field is used only when migrating from WebSphere Application Server V4.0. For WebSphere Application Server Version 4 Advanced Edition, you can specify a file created by **`XMLConfig`** export instead of the one migration would use by exporting **`XMLConfig`** directly. For WebSphere Application Server Version 4 Advanced Edition Single Server, you can specify a configuration file other than the default one.



Detected versions of WebSphere Application Server

The following versions of WebSphere Application Server exist on your computer. Select a product from the list, or specify the product information for a product that does not appear in the list.

IBM WebSphere Application Server, Version 5.0.0
IBM WebSphere Application Server, Version 5.1.0
IBM WebSphere Application Server Network Deployment, Version 5.0.0
IBM WebSphere Application Server Network Deployment, Version 5.1.0

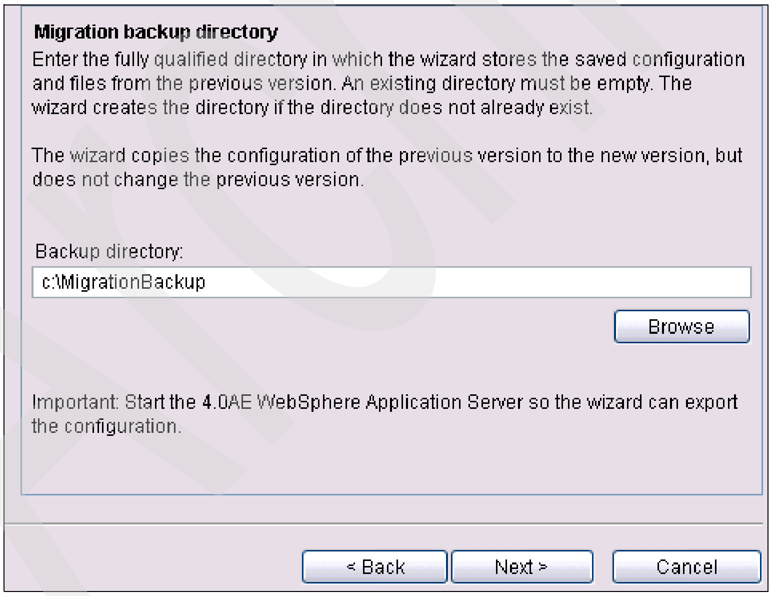
□ Specify the installation location if your existing product is not listed.

Installation root directory of the previous version:
C:\WebSphere\AppServer50

Location of the Version 4.0 Advanced Single Server Edition configuration file:

Figure 8-37 Detected versions in migration wizard

Figure 8-38 shows how to specify the location of the migration-specific backup directory that is created by **WASPreUpgrade**. You can specify any valid directory name for this field. It is advisable to specify a location that is not a subdirectory of an existing WebSphere Application Server location. It is possible that, at some point, that version of WebSphere Application Server may be uninstalled and you may unintentionally erase this migration-specific backup directory.



Migration backup directory

Enter the fully qualified directory in which the wizard stores the saved configuration and files from the previous version. An existing directory must be empty. The wizard creates the directory if the directory does not already exist.

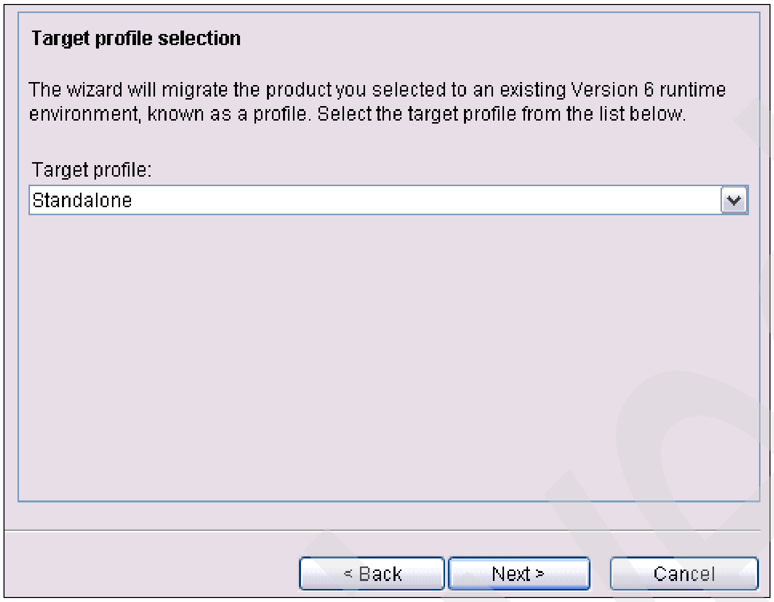
The wizard copies the configuration of the previous version to the new version, but does not change the previous version.

Backup directory:
c:\MigrationBackup

Important: Start the 4.0AE WebSphere Application Server so the wizard can export the configuration.

Figure 8-38 Specifying backup directory in migration wizard

Figure 8-39 on page 133 shows how to select the Version 6 profile that is used during migration. It has a drop-down menu of all existing profiles that are compatible (to the basic degree) with the type of migration that is being performed. For example, in this case we are migrating from WebSphere Application Server Version 5. A compatible profile is either a *stand-alone* or a *custom* profile. These types of profiles are listed in the drop-down box. However, there are some scenarios where the values used when creating these profiles are important. See 12.5.3, “V6 installation preparation” on page 217 for more details on specific node name or cell name requirements for specific migration scenarios.



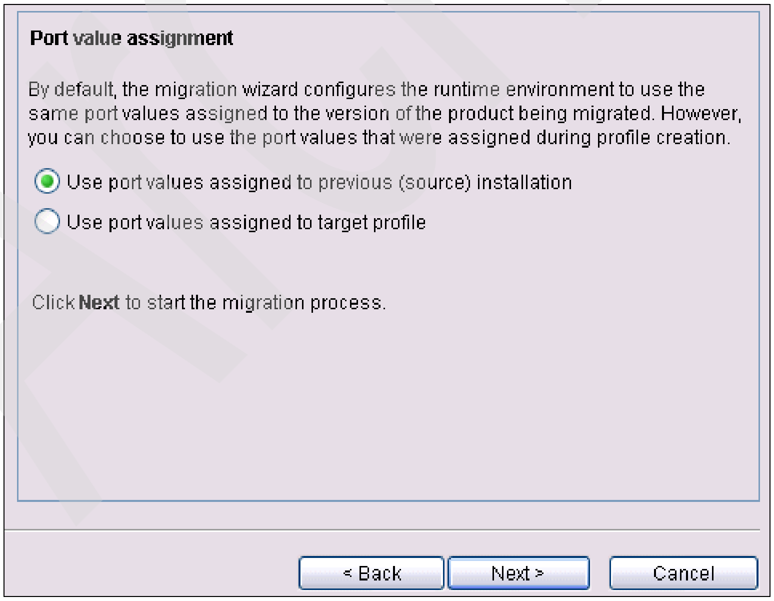
Target profile selection

The wizard will migrate the product you selected to an existing Version 6 runtime environment, known as a profile. Select the target profile from the list below.

Target profile: Standalone

Figure 8-39 Specifying target profile name in migration wizard

Figure 8-40 shows the choice for how to handle migration of port assignments. The default value is to use the same port values in the WebSphere Application Server Version 6 configuration as the existing WebSphere Application Server configuration. This assumes a replacement of the existing WebSphere Application Server installation with the latest WebSphere Application Server Version 6 configuration. Choosing this option means you cannot run WebSphere Application Server installations and the new WebSphere Application Server Version 6 installations concurrently because there would be port conflicts.



Port value assignment

By default, the migration wizard configures the runtime environment to use the same port values assigned to the version of the product being migrated. However, you can choose to use the port values that were assigned during profile creation.

● Use port values assigned to previous (source) installation

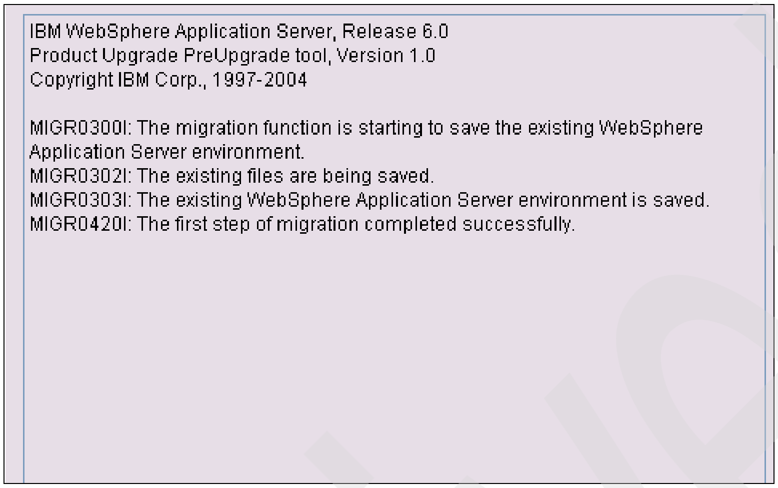
○ Use port values assigned to target profile

Click **Next** to start the migration process.

Figure 8-40 Specifying port assignments in migration wizard

You can also choose to not map the port values over during the migration process. You do this by selecting to use port values already assigned to the target profile. This allows you to run the previous WebSphere Application Server installations and the new WebSphere Application Server Version 6 installations concurrently. This radio button maps to the `-replacePorts` parameter on the **`WASPostUpgrade`** command.

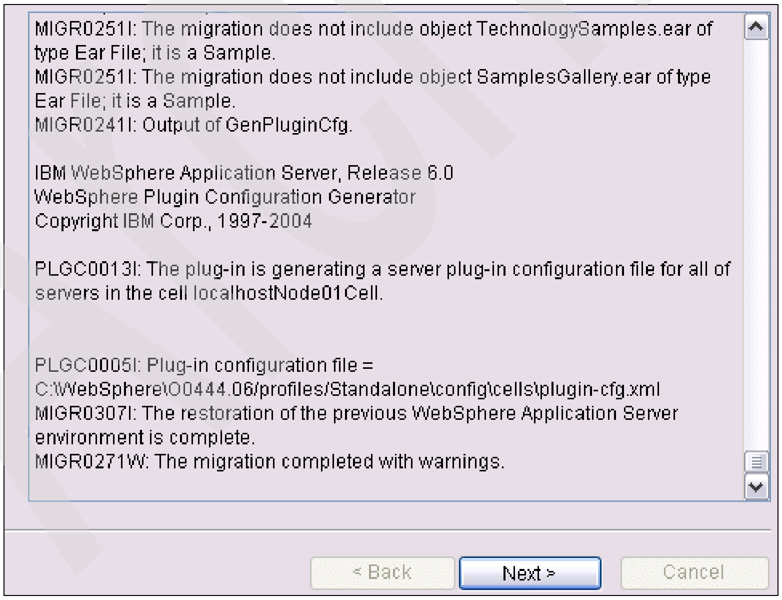
Figure 8-41 shows the start of running the **`WASPreUpgrade`** command in the migration wizard. The command takes some time to complete.



```
IBM WebSphere Application Server, Release 6.0
Product Upgrade PreUpgrade tool, Version 1.0
Copyright IBM Corp., 1997-2004

MIGR0300I: The migration function is starting to save the existing WebSphere
Application Server environment.
MIGR0302I: The existing files are being saved.
MIGR0303I: The existing WebSphere Application Server environment is saved.
MIGR0420I: The first step of migration completed successfully.
```

Figure 8-41 Executing WASPreUpgrade in migration wizard

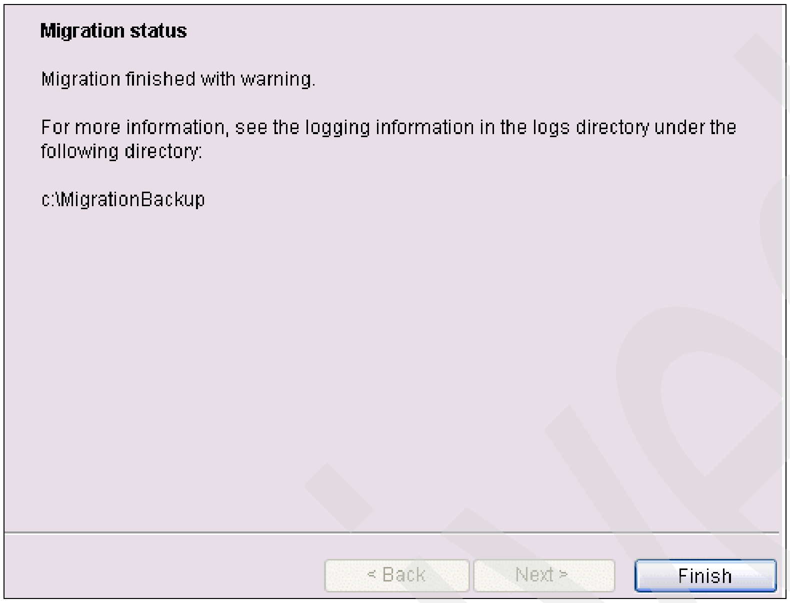
Figure 8-42 shows the completion of running the **`WASPreUpgrade`** command in the migration wizard. Verify the last message is shown, which indicates a successful first step of Migration has completed. One failure that can occur for WebSphere Application Server Version 4 Advanced Edition scenarios is when the 4.0 administration server is not running. When **`WASPreUpgrade`** runs for this scenario, it calls **`XMLConfig`** export to access the existing configuration in the repository. The 4.0 administration server must be running for this to work.



```
MIGR0251I: The migration does not include object TechnologySamples.ear of
type Ear File; it is a Sample.
MIGR0251I: The migration does not include object SamplesGallery.ear of type
Ear File; it is a Sample.
MIGR0241I: Output of GenPluginCfg.

IBM WebSphere Application Server, Release 6.0
WebSphere Plugin Configuration Generator
Copyright IBM Corp., 1997-2004

PLGC0013I: The plug-in is generating a server plug-in configuration file for all of
servers in the cell localhostNode01Cell.


PLGC0005I: Plug-in configuration file =
C:\WebSphere\O0444.06/profiles/Standalone\config\cells\plugin-cfg.xml
MIGR0307I: The restoration of the previous WebSphere Application Server
environment is complete.
MIGR0271W: The migration completed with warnings.
```

Figure 8-42 Completion of WASPreUpgrade in migration wizard

Figure 8-43 on page 135 shows the results of running the **`WASPostUpgrade`** command in the migration wizard. This is the final panel and shows a summary of the final messages from **`WASPostUpgrade`**. This can be quite a bit of output and shows the details of migrating the configuration, installing the applications and (in some scenarios) calling **`GenPlugInCfg`** to generate the plug-in configuration. It is important to view the final message for an indication of success, warning, or failure. In most cases a warning message is displayed at the bottom of the panel. You should review any messages that have a code ending in “W” so that you can see warnings about deprecations or configuration updates that need to be done.



Migration status

Migration finished with warning.

For more information, see the logging information in the logs directory under the following directory:

c:\MigrationBackup

Figure 8-43 Completion in migration wizard

8.6.4 Script compatibility

The **`convertScriptCompatibility`** command is used to convert all or portions of a configuration from script compatibility mode to a configuration that may not be compliant with existing 5.x customer scripts.

The scope of the configuration changes in the configuration are dependant on the type of profile that is being processed. For stand-alone profiles, the default is to convert all servers owned by the node in that configuration. An additional -serverName parameter can be used for more granular control. For Deployment manager profiles, the default behavior is to convert all nodes and all servers owned by those nodes. Additional parameters **`-nodeName`** and **`-serverName`** can be used for more granular control. Note that nodes are checked to verify that they are at a 6.0 level before they are processed; this is required to support mixed-node configurations. Client environments are not processed.

The conversions that take place by this command are:

1. processDef → processDefs - Objects of type processDef are converted to use processDefs as defined in the WebSphere Application Server Version 6 server.xml model. The existing processDef object remains in the configuration and is ignored by the runtime. For Distributed and iSeries platforms there can be only one occurrence of a processDefs object in a server configuration. If an existing processDefs object is found when performing this conversion then it is used and updated, otherwise a new object is created. For zSeries platforms more than one processDefs object can exist in the server. For this case the object with the same name is found and updated.
2. transports → channels - Existing transport entries in the configuration are mapped to channel support. This affects server.xml and serverindex.xml files. The values of the transport settings are used to create new channel entries.

How to run it

```
convertScriptCompatibility [-help]
         [-backupConfig true | false]
         [-profileName <profile name>]
         [-nodeName <node name>]
         [-serverName < server name>]
         [[-traceString <traceSpec> [-traceFile <fileName>]]
```

where:

- `[-help]` - Displays help for the command.
- `[-backupConfig <true | false>]` - This is used to back up the existing configuration before changes are made to the configuration by migration. If not specified, the default is `true`.
- `[-profileName <profile name>]` - If provided, this value can be used to specify a particular profile configuration in the WebSphere Application Server Version 6 environment. If not specified, the default profile is used. If the default has not been set or cannot be found, an error is returned.
- `[-nodeName <node name>]`- If provided, this can be used to specify that a particular node name should be processed instead of every node in the configuration. It is most useful when processing a Deployment manager profile but is allowed on all profile types. If not specified then all nodes in the configuration are converted.
- `[-serverName <server name>]` - If provided, this can be used to specify that a particular server name should be processed instead of every server in the configuration. It can be used on all profile types and can be used in conjunction with the **`-nodeName`** parameter when processing Deployment manager profiles. If not specified then all servers in the configuration are converted. If used in conjunction with the **`-nodeName`** parameter then all processing is limited to the specified node name.
- `[-traceString traceSpec [-traceFile fileName]]` - These optional parameters are used to gather trace information for use by IBM service personnel. The value of traceString is "`*=all=enabled`" and must be specified with quotation marks to be processed correctly.

8.6.5 J2EE application clients conversion

J2EE EAR files may exist on the client or server that have client JARs contained in them that have J2EE resources within those JARs. If these resources are at WebSphere Application Server V4.0 level they need to be updated to be WebSphere Application Server Version 6 resources. You may also want to upgrade WebSphere Application Server Versions 5.0 and 5.1 applications to ensure they are using the latest level of WebSphere Application Server 6.0 resources. The **`clientUpgrade`** command can be run on these J2EE EAR files to perform this upgrade.

This command is not run automatically as part of the installation process because the J2EE EAR files are not in a predetermined location. The customer must run this command against their J2EE EAR files themselves.

Note that copies of client JARs are not made in the J2EE EAR file. If the J2EE EAR file needs to be used with WebSphere Application Server V4.0 as well as with WebSphere Application Server Version 6, then it is necessary to make a copy of the J2EE EAR file before using **`clientUpgrade`**.

How to run it

```
clientUpgrade EAR_file
          [-clientJar client_jar ]
          [-logFileLocation logFileLocation]
          [-traceString traceSpec [-traceFile fileNSame ]]
```

where:

- `EAR_file` - The fully qualified path to the EAR file that contains client JAR files to process.
- `-clientJar` - Specifies a JAR file for processing. If not specified, the program transforms all client JAR files in the EAR file.
- `-logFileLocation` - Use this optional parameter to specify an alternate location to store the log output.
- `-traceString -traceFile` - Gathers trace information for IBM service personnel. Specify a traceSpec of `"*=all=enabled"` (with quotation marks) to gather all trace information.

9

z/OS runtime administration overview

This chapter discusses the most important aspects of how WebSphere Application Server for z/OS is administered.

9.1 Significant concept changes from V5

You should already be somewhat familiar with WebSphere Application Server and specifically, how to administer V4 or V5. Given that, the quickest way to be productive on V6 is to understand the significant administration task changes.

9.1.1 Installation changes

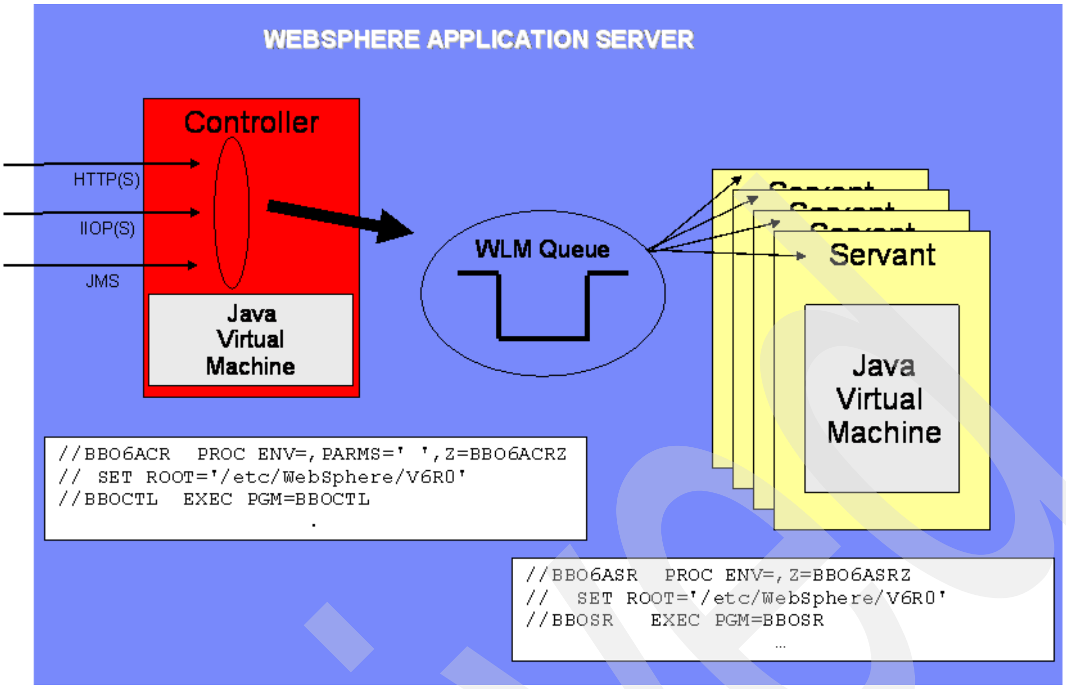
These are the significant installation changes from V5:

- The Java SDK is part of the product code. It is unpacked automatically during the APPLY process into your product directory: /usr/lpp/zWebSphere/V6R0/java. The code is certified at particular Java levels. You should not make any changes unless instructed to do so.
- System applications stay with the product binaries. The system applications include:
  - adminconsole
  - filetransfer
  - ManagementEJB.ear
  - SchedulerCalendars.ear

  In versions prior to V6, these applications are installed along with your other applications into your server and they became part of your configuration. With V6, they stay with the product binaries and a symbolic link points to them from a new systemApps directory in your configuration HFS. Some applications used for the PMEs also reside in the systemApps directory, including ManagementEJB.ear and SchedulerCalendars.ear.
- A second very large HFS contains CD images to be installed on non z/OS platforms. The CDs include Edge Components, the WebSphere Application Server Application Server Toolkit (ASTK), a trial version of Rational Application Developer, and Tivoli Access Manager. See the appropriate articles in the WebSphere Application Server Information Center for installation and use of these CDs.

9.2 Process architecture

The z/OS server is actually split among address spaces. Figure 9-1 illustrates the address space configuration. z/OS uses the idea of a *controller address space,* which acts as the network endpoint for HTTP, IIOP, and messaging traffic. The controller routes application work to a Work Load Manager (WLM) queue from which a second address space, a *servant*, selects and executes work. The controller is started with a JCL Start procedure that is created for you during customization. The servants are started and stopped by WLM depending on workload.



```
//BBO6ACR   PROC ENV=,PARMS=' ',Z=BBO6ACRZ
//  SET ROOT='/etc/WebSphere/V6R0'
//BBOCTL   EXEC PGM=BBOCTL
                             .
```

```
//BBO6ASR   PROC ENV=,Z=BBO6ASRZ
//   SET ROOT='/etc/WebSphere/V6R0'
//BBOSR    EXEC PGM=BBOSR
                    ...
```

Figure 9-1 Controller address space

The application code runs in the servant regions, isolated from the WebSphere runtime code. Notice that both the controller and servant regions contain a JVM™.

Figure 9-2 shows a new box called the *control region adjunct*, or CRA. This is a new address space in V6.0 that works in conjunction with the controller as the communications endpoint for messaging. The controller is responsible for IIOP and HTTP communications. Unlike the controller, which is always there, the CRA is only present when a server becomes a bus member and has a messaging engine created. The messaging engine runs in the CRA, communicating with the Resource Adapter (RA) shown.

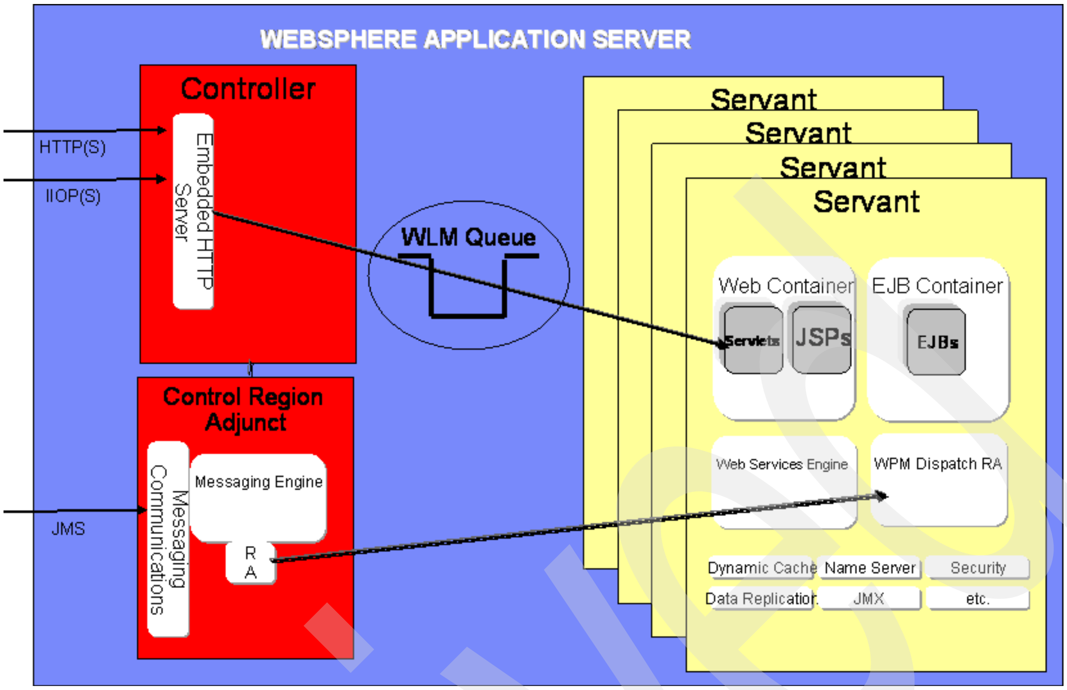


Figure 9-2 Controller region adjunct address space

Most of the components are present in the servant region, where the application code runs.

The first entity you would want to configure is a stand-alone application server. In Figure 9-3, notice a new box labeled the *daemon*. This is a single address space that serves as the Location Service Daemon. It is started automatically when your server is started. There is one daemon per cell per LPAR. The idea of a cell is not that interesting in the stand-alone application server case, but you can see where the multi-address space server fits into the picture. The administrative console application runs along with your own applications in the server.

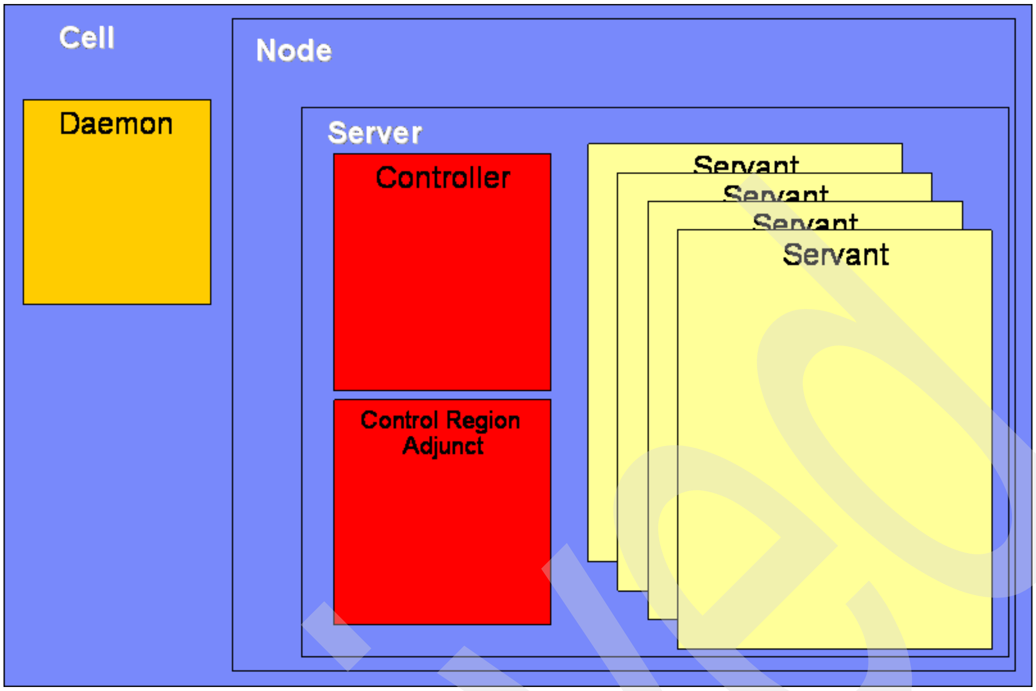


Figure 9-3 Daemon address space

Next, look at how the picture changes in the Network Deployment configuration represented by Figure 9-4. The node agent is another single address space and is used to communicate with the deployment manager to manage the application servers in the node. Notice also that there are now 2 nodes in the cell. The deployment manager node is a special purpose node where the administrative console application runs. No user applications run in the deployment manager node but it has the same server structure as the base application server node, minus the CRA address space. WLM starts servant regions as needed to service the requests coming into it from the nodes in the cell. The node on the left has been federated into the cell with the deployment manager using the addNode command.

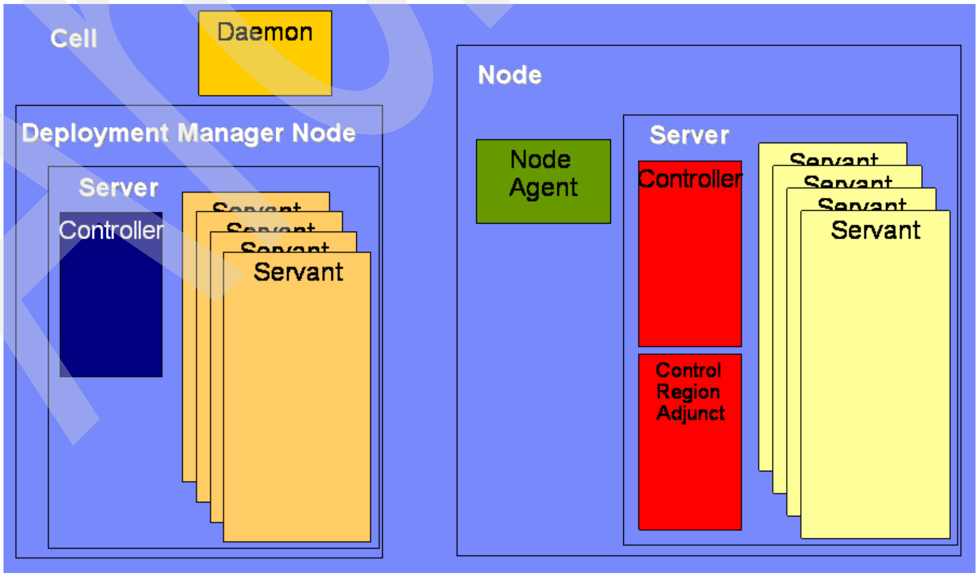


Figure 9-4 Network deployment configuration

9.3 Administrative console

The administrative console for WebSphere Application Server is virtually identical to that for the distributed operating platforms. See 8.3, “Administrative console” on page 113 for more information about the administrative console.

9.4 How to perform basic functions

These MVS™ commands are used to perform basic server process functions involving starting and stopping processes.

Starting servers

All server types use the same basic form of the MVS start command, which is of the form:

```
START server_proc_name, JOBNAME=server_short_name,
ENV=cell_short_name.node_short_name.server_short_name
```

See 13.2.6, “Start the server” on page 298 for a more detailed example of how to start an application server.

Stopping servers

All server types use the same basic form of the MVS start command, which is of the form:

```
STOP server_short_name ENV=cell_short_name.node_short_name.server_short_name
```

See “Stop and restart server” on page 301 for a more detailed example of how to stop an application server.

9.5 Changes in default settings

See 8.5, “Changes in default settings” on page 118 for this information. The default setting information in that section applies to all operating platforms.

9.6 Automatic migration utilities

This section describes utilities that are shipped with the WebSphere Application Server that can be used for migration of system configuration and applications from a previous version of WebSphere Application Server to the current one.

9.6.1 Overview of the Version 6 migration process

Migration process flow

Here's a snapshot of the process to migrate a multi-node Network Deployment configuration:

1. Take an inventory of your existing environment so you have a feel for things like mount points, userids and groups, node directory roots, and JCL start procedures.
2. Back up your source configuration HFS because the migration process makes a minor alteration to that configuration HFS. Just to be sure, you should back up the source configuration HFS using your preferred file backup tool.
3. Run through the ISPF customization dialogs that are provided with V6. The dialogs generate customized migration jobs for the node.

4. Migrate the node by running the customized job. This involves submitting the jobs and checking for a zero return code.
5. Perform post-migration work, which involves creating a few additional RACF® profiles.

Differences from V5.1 migration process

While many of the concepts are the same, some of the specifics of the V6 migration process are different. We identify them here:

- One primary migration utility

  In the V5.0-to-V5.1 migration process there are five migration utilities: BBOXMIG1 through BBOXMIG5. With the V6 migration process, there is one job responsible for doing the heavy lifting of migration. There are other jobs, but they do more mundane things like create and mount the new HFS, or copy new JCL procedures into PROCLIB.

- ISPF Dialog creates customized migration jobs

  The V5.0-to-V5.1 migration process comes packaged as a series of JCL jobs in a PDS that you have to modify. With V6, there is an option in the ISPF Customization Dialogs that creates those migration jobs for you, with all the relevant information updated in all the right spots.

- Simpler ISPF panels to define configuration being migrated

  The V5.0-to-V5.1 migration process requires you to run through the standard customization dialogs to build a skeleton configuration. That skeleton is used as input to the migration process. Much of the information entered into the skeleton configuration is irrelevant information. In other words, information is collected by the ISPF dialogs, but not used by the migration process. With V6, the ISPF panels are much simpler, capturing only that information the migration utilities really require.

- Version 6 information asked for in ISPF panels

  These utilities convert a set of V5.x servers into V6.0 servers. The architecture of the V6 server is different from V5, most notably the High Availability Manager function. The ISPF panels that are used to generate the migration jobs ask you for the host value assigned to this new function.

- Additional V6 post-migration security work needed

  V6 requires a few more security profiles compared to V5. That means that after migrating a node, some additional RACF (or other SAF interface security product) work is required. See 13.1.1, “Migrating the deployment manager node on SYSC” on page 244 for details on how this is done.

- New JCL start procedures created during migration job generation

  The V6 migration process asks for names to be used for the JCL start procedures for server controllers of the migrated node. It then generates new JCL for you and copies that JCL into your PROCLIB. The implication is that new JCL procedures are required for V6, and indeed the new JCL procedures do look quite a bit different. But there is a subtlety here that we want to stress:

  – New JCL start procedures are needed. At a minimum the SET ROOT= value that points to the configuration mount point must be different. If STEPLIB statements are used then those also need to be different from V5.

  – The names of the JCL start procedures do not need to be different. Keeping the same names as used with V5 provides a key benefit: you won't need to create new STARTED profiles.

- New controller start

  What this means is that you'll need to back up your V5 start procedures before running the job that copies in the new JCL. See “Preliminary work” on page 245 for details of how to do a backup.

- Mixed-version nodes within a cell on the same MVS image permitted (with limitations)

  One of the nice features of V6 is that it permits a deployment manager running at the V6 level of code to manage nodes still at the V5 level of code. A V6 daemon is capable of supporting V5.1 servers in a node that's part of that daemon’s cell, all on the same MVS image. This provides considerable flexibility in the migration process.

9.6.2 Discussion of previous migration operation - how migration to 5.1 works

The V5.1 migration process requires that two or more nodes from a cell on the same MVS image be migrated simultaneously. In other words, the nodes must be migrated one immediately after the other. If you have an application server node on the same MVS image as its deployment manager, you must stop both and migrate them both before you can restart either.

The issue with this requirement is due to the daemon server. Two or more nodes from the same cell on the same MVS image share the same daemon server. If the daemon server is running at the higher level of code, then the not-yet-migrated node (running at the lower level of code) would have to be compatible with the daemon. In V 5.1, a daemon is not compatible with servers in a node running at V5.0.

If the two nodes on the same MVS image were in different cells, that would be okay because they would have been supported by different daemon servers. The problem arises when two nodes from the same cell reside on the same MVS image; they consequently share the same daemon.

If the two nodes were on different MVS images, that too would be okay because that would imply different daemon servers. For a description of how a Network Deployment configuration that spans two or more MVS images could be non-disruptively migrated from V5.0 to V5.1, see white paper WP100441 at this address:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100441

But with Version 6, you have additional flexibility. When migrating from V5.1 to V6, the daemon code is compatible with both V6 and V5.1, so a configuration like this is possible:

9.6.3 Limitations of mixed version cells

The V5 node must be at V5.1, and not V5.0.x

If one of the nodes is the deployment manager node, it must be at the higher level.

If your V5 configuration is still at V5.0.x, then plan on migrating all nodes from a cell on the same MVS image immediately after one another. Migrate the deployment manager first, then the application server node, because the deployment manager needs to be up and running to migrate an application server node.

But if your nodes are at V5.1 (and W510207 at a minimum service level), then you have the flexibility to have a V6 node and V5.1 node from the same cell coexist on the same MVS image.

Coexistence of separate V5 and V6 cells on the same MVS image is permitted. There is nothing about V6 that would prevent a V5 cell and a V6 cell from coexisting on the same MVS image or the same Sysplex.

There are some limitations to this:

- V5 and V6 modules cannot both be in LPA/LNKLST at the same time.
- You cannot share JCL start procedures between the V5 cell and the V6 cell when STEPLIB statements are in the JCL.

V5 and V6 nodes can coexist if you provide sufficient separation for the following:

- Separate HFS
- Separate JCL start procedures
- Separate mount point

There are some limitations to this mixed version coexistence:

- The deployment manager must be at V6.0 to manage V5.1 nodes. The reverse is not permitted: a V5.1 deployment manager can't manage a V6.0 node.
- V5 and V6 modules cannot both be in LPA/LNKLST on the same MVS image at the same time. At a minimum one must be STEPLIBed.
- A V6.0 node cannot share JCL start procedures with a V5.1 node when STEPLIB statements are used in the JCL.

Note: There is only one way to create a mixed-version cell, which is by migrating the deployment manager via the automated migration utilities. Migrating the deployment manager to a higher level leaves the nodes down-level, thus creating a mixed-version cell. The nodes were previously members of the cell, and the nodes remain members of the same cell once the deployment manager is migrated.

It is not possible for a down-level standalone node to join a cell via federation. Any attempt to federate a down-level node to a cell will fail with an indication that such an operation is illegal.

Coexistence of V6 with V5.0 nodes in the same cell on the same MVS image is not permitted. The daemon server, which must be at the level of code equal to the highest code-level node for that cell on that MVS image, is not compatible with a V5.0 node. It is compatible with a V5.1 node.

A cell with mixed V6.0 nodes and V5.0 nodes is permitted, provided they are not mixed on the same MVS image. On another MVS image, where the V5.0 servers can be supported by a V5.0 daemon, it is okay. There are some limitations to this:

- The deployment manager must be at V6.0 to manage V5.0 nodes. The reverse is not permitted: a V5.0 deployment manager can't manage a V6.0 node.
- A V6.0 node cannot share JCL start procedures with a V5.1 node when STEPLIB statements are used in the JCL.

Down-level nodes in a cell are fairly flexible with respect to how the node can be managed. You can do all these things with a down-level node:

- You can install applications.
- You can change settings like short names and ports.

- You can add servers to any node. Make sure the deployment manager is at the 6.0.2 service level or higher, since the capability to add servers to down-level nodes in the cell was not possible prior to 6.0.2.
- You can start and stop the server from the administrative console.
- You can start and stop applications from the administrative console.

9.6.4 Minimum version, release, and maintenance levels

Automated migration to V6.0 is supported for both V5.0 and V5.1. The required maintenance levels to allow automated migration are shown in Table 9-1. A V4 configuration cannot be migrated using the automated migration tools. You must perform a manual migration if your current system is V4. See 13.2, “Manual migration: Installing a V6 stand-alone server” on page 276 for an example of a manual migration.

Table 9-1 Maintenance levels required for automated migration

Version	Maintenance level
V 5.0	W02025
V 5.1	W510207

If your nodes are not at the appropriate level of maintenance, apply the maintenance and make sure at least one server from the node runs applyPTF.sh so the node's configuration is brought up to the minimum maintenance required for migration.

The discussion of the additional flexibility of migrating V5.1 to V6.0 raises the question of whether there is any advantage to performing a two stage migration. Is there any advantage to first migrating everything to V5.1, and then migrating to V6.0. The advantage to being at V5.1 during migration is an additional degree of flexibility with regard to how the nodes are migrated. It centers around the daemon servers. A daemon migrated up to the V6 level of code is capable of managing a V5.1 node from the same cell on the same MVS image. But this is not the case for a V5.0 node. Therefore, if you have two nodes from a cell on the same MVS image and they are at V5.0, they must be migrated one right after the other. If the two nodes from a cell on the same MVS image are at V5.1, then the second one can be migrated at your leisure.

However, migrating from V5.0 to V5.1 is not a trivial undertaking. Migrating from V5.0 to V6 directly is probably the best way to go.

Sequence of migration steps

Always migrate the deployment manager node first. Version 6 is capable of managing V5 nodes, but a V5 deployment manager cannot manage V6 nodes.

After the deployment manager node is migrated, other nodes may be migrated as you please.

What can be running during migration

All servers in the node must be stopped in order to migrate a configuration; all the servers in the node are migrated at the same time.

The deployment manager must be up and running when application server nodes are migrated.

It is quite possible to have some nodes in a cell migrated to V6 while other nodes are still at V5. This provides the ability to provide a non-disruptive migration where some nodes continue to operate while other nodes are being migrated.

Further, a V6 deployment manager is capable of existing with V5.1 nodes for quite some time, so there is no need to feel as if you have to rush a migration to fit within a maintenance window. The V6 deployment manager knows that a node it is managing is at the lower level of code, and when configuration changes are made to the node the V6 deployment manager makes sure those changes have the V5 format.

Security considerations

You can run migration with global security enabled. The migration ISPF customization panels ask for the WebSphere Admin ID and password when an application server node is being migrated. This is needed so the migration utility can connect to the running V6 deployment manager and synchronize.

When you configure security profiles, you should re-use the profiles you were using for V5 before migration. By "security profiles" we mean things like (in the language of RACF) userids and groups, STARTED profiles, SERVER and CBIND profiles, keyrings, and certificates.

The relationship between the configuration and the underlying security profiles is so tight that trying to map the migrated configuration to a new set of profiles would be extremely challenging. It is far better to use the same profiles.

The post-migration security work mentioned earlier involves creating a few new profiles needed by a V6 server, regardless of whether global security is enabled or disabled.

TCP/IP ports

The migration carries over the ports you had assigned to your servers in V5. However, a V6 application server has six new ports in addition to the six original V6 ports, so the total number of ports per application server is now 12. The migration process propagates the six V5 ports, and assigns default values for the six new ports. You will probably want to re-map these new ports so they adhere to your port allocation scheme.

Attempting to migrate several nodes concurrently

We recommend that you do not attempt to migrate more than one node at a time. In general, doing this does not work due to conflicting use of the temporary directory. The BBOWMG3* job tries to use the /tmp/migrate directory for work space. The job attempts to copy the file bbomigrt2.sh into the directory. Depending on whether that file already exists, and what the permissions are on it, a migrate job either operates or fails. Failure is indicated by an RC=256 on the first step (`SETUP`) of the BBOWMG3* job.

PRR processing

When migrating an application server node, two migration utilities are generated: BBOWMG1F and BBOWMG2F. They perform the Peer Resource Recovery (PRR) processing for the application servers in a node. You must run these jobs if you have XA connectors installed in your application servers. Running those jobs even if you do not have XA connectors installed will not hurt anything. So if you are not sure, then run the jobs.

10

Migration tasks

This chapter discusses major migration tasks for specific WebSphere components.

10.1 Migration utilities supported versions

Table 10-1 shows the products that the automatic migration utilities support as a starting point.

Table 10-1 Supported WebSphere releases for migration starting points

Version 4.0	Version 5.0	Version 5.1
WebSphere Application Server Advanced Edition	WebSphere Application Server	WebSphere Application Server
WebSphere Application Server Advanced Edition Single Server	WebSphere Application Server Express	WebSphere Application Server Express
WebSphere Application Server Enterprise Edition	WebSphere Application Server Network Deployment	WebSphere Application Server Network Deployment
	WebSphere Application Server Enterprise	WebSphere Business Integration Server Foundation
	WebSphere Application Server for z/OS	WebSphere Application Server for z/OS

Each of these products can be migrated using automatic migration utilities. The automatic migration utilities transform an operational WebSphere configuration on one of the versions listed in Table 10-1 into a new operational configuration for V6. The automatic migration utilities save you a lot of time by eliminating the need to create the new configuration manually. If your target version is not listed in Table 10-1, then you must perform a manual migration. See 8.6, “Automatic migration utilities” on page 125 for an overview of the automatic migration utilities for distributed platforms. See 8.2.3, “Profiles” on page 105 for an overview of profiles.

10.2 Migrating nodes on distributed platforms

Node migration on distributed platforms is either of a single stand-alone node, or of a collection of nodes. Stand-alone nodes migrate to stand-alone nodes. Collections of nodes have different terminology and architecture depending on whether you are migrating from V4 or V5. However, regardless of version, collections of nodes migrate to a V6 cell.

10.2.1 Migrating single stand-alone nodes

Any of the products in Table 10-1 can be configured to operate as a single stand-alone server. Even though we use the term *stand-alone server*, we also extend the term to mean a stand-alone node that may have multiple application servers defined. We address Network Deployment cell configurations in 10.2.3, “Multiple nodes for V5” on page 156. However, the migration steps described for a stand-alone node also apply to each of the nodes in a Network Deployment cell.

See 12.2, “Automatic migration: V4 AE Single Server to V6” on page 187 and 12.3, “Automatic migration: Application Server V5 to Express V6” on page 191 for complete examples of migrating stand-alone nodes.

The migration steps for stand-alone servers is conceptually simple. See Figure 10-1 on page 153 for a diagram of the steps.

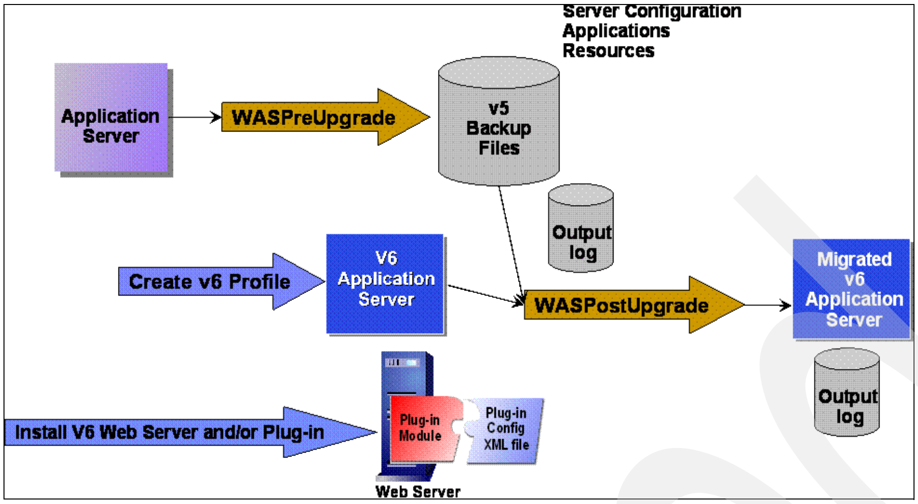


Figure 10-1 Migrating application servers from previous versions

1. Complete a V6 installation and create an application server type profile. When you create the profile and specify the node name, be aware of your choice of node name. The node name of the target V6 node must match the node name of the source application server configuration for these configurations, where the source system is:

   – WebSphere Application Server Version 4 Advanced Edition

   – WebSphere Application Server V5 node that is a member of a Network Deployment cell

   For other source configurations, the node name is not crucial.

   See 8.2, “Installation” on page 100 for more information about installation and profile creation.

2. Before WASPreUpgrade is run, you may want to stop the server, but stopping at this point is optional. One exception is migrating WebSphere Application Server Version 4 Advanced Edition. The V4 administration server must be active so that WASPreUpgrade can contact the administration server to extract the configuration information.

3. WASPreUpgrade is run on the source system to copy the server and application into a backup directory. WASPreUpgrade can be run directly from the installation media, thus eliminating the need to install Version 6 on the source system. Running directly from the installation media allows you to migrate from one system to a different system. If you do want to migrate to a different physical system, you need to transfer this directory in its entirety to the target system in order to perform the next step.

   Once WASPreUpgrade completes consult the output log, which is placed in the backup directory that it creates. You should evaluate any warnings or errors and take appropriate action.

   If you chose to use the migration wizard, you will skip the next step. Also note that since the migration wizard calls WASPreUpgrade and WASPostUpgrade in a seamless step, you cannot use the migration wizard in a situation where you are migrating to a different physical system. See 8.6.3, “Graphical wizard for configuration migration” on page 130 for more information about running the migration wizard.

4. WASPostUpgrade is run on the backup directory. Examine the output log for any warnings or errors.

5. Once the WASPostUpgrade completes, the new node can be tested while the old node continues to run. If the old node and the new node reside on the same physical system,

you may have to stop the old node and all the servers if you did not take steps to ensure that there are no port conflicts. When the new node is finished testing, the old node can be permanently taken down and the new node switched online.

6. If your application server works in conjunction with a Web server, you must upgrade your Web server installation to a supported level. You must also upgrade the WebSphere plug-in installation so that the Web server is able to route requests to the V6 application server. This upgrade is a manual installation of these two components. See “Web server” on page 104 for an overview of the installation steps for Web servers and plug-ins.

There are several situations that may require you to make manual modifications to the target configuration once migration is complete. You can either edit files in the backup directory or change the configuration items in the target profile with the administrative console. Editing the files in the backup directory may be best for you if you are comfortable with editing XML files or you have some sort of automation available to make changes. If you do make changes to the backup directory, make sure you make a backup copy of any file you change. If you are not comfortable changing XML files, you should make your changes with the administrative console.

- If you are migrating from one system to a different physical system, the new system may not be exactly identical. Some items to examine and evaluate for possible changes are:
  - Users and passwords for security settings
  - Fixed file system paths that may be part of a classpath for resources such as JDBC providers or shared libraries
  - Database specifications such as database names, database hosts, or database users
- Port conflicts may arise after migration. The remedy is to re-specify the port assignments in a new profile and re-run the migration steps. The migration wizard can help you identify port conflicts. See the Infocenter references already mentioned for more details on fixing port conflicts.

10.2.2 Multiple nodes for V4

WebSphere Application Server Version 4 Advanced Edition has the capability of managing multiple servers through a single point called the administration server. WebSphere Application Server Network Deployment Version 6 is the equivalent product that has the same capability of managing several servers on different systems from a single point of control. The Version 6 Network Deployment manager performs the same function as the V4 administration server in that it acts as a single control point for managing multiple servers.

Here we describe the general steps to migrate a WebSphere Application Server Version 4 Advanced Edition installation to a WebSphere Application Server Network Deployment Version 6, with all of the node configuration preserved as well as the applications.

See 12.4, “Automatic migration: V4 AE Server Group to a V6 Network Deployment Cluster” on page 205 for a detailed example of the following steps.

1. Install WebSphere Application Server Network Deployment Version 6 and create a profile with the type of deployment manager. See 8.2.3, “Profiles” on page 105 for detailed information about creating profiles.
2. Migrate the global configuration information from the V4 installation to the deployment manager profile. Figure 10-2 on page 156 illustrates this step.
3. Start the deployment manager. See “Starting a deployment manager” on page 116 for details about starting the deployment manager.
4. For each node in the V4 installation that you wish to migrate:

a. Migrate the node from the V4 installation to a V6 application server node according to the detailed steps outlined for a single node in 10.2.1, “Migrating single stand-alone nodes” on page 152. Ensure that the node name in the V6 profile matches the node name of the V4 node that you are attempting to migrate. Figure 10-3 on page 156 illustrates this step.

b. Add the node to the cell using the **addNode** command. The action of adding a node to a cell is also called *federating a node*. Determine whether you want the applications installed on that node to remain on the node once it joins the cell. If you do want to keep the applications, make sure you use the **-includeapps** option on the **addNode** command.

5. After all the nodes are migrated, install any applications that are assigned to server groups in the V4 installation. If there are such applications, they are placed in the installableApps directory under the application server profile. Installation is accomplished by installing an Enterprise Application via the administrative console on the deployment manager.

Note that the difference between migrating the deployment manager and migrating a node is only the match on the node name. For the deployment manager, the V6 node name must not match any of the V4 node names and thus only global information is copied to the deployment manager profile. In other words, global information is intended to be applied to every node of the cell. For an application server node migration, the V6 node name of the profile must match the intended V4 node name; the node name is used as an implicit selection for the intended node. If the node does not match, and the target is an application server type profile, the migration utilities do not migrate the intended information, namely the node configuration and applications.

The applications that are migrated for a node migration fall into two categories.

- The applications that are assigned to specific servers in the V4 domain are migrated to the V6 application server profile in an installed state. This means the application is installed, configured, and ready to go. When you start the application server, these installed applications are loaded and ready to execute.

  When you add the node to the cell, you have to determine whether you want the applications that are automatically installed by the automatic migration utilities to remain installed after joining the cell. If you do want the applications to remain, remember to specify **-includeApps** on the **addNode** command.

  ```
  addNode dmgr_host -includeApps
  ```

- Applications that are assigned to server groups, or clusters, are not automatically installed in the node. These applications are stored, in EAR format, in the installableApps directory in the V6 profile. It is your responsibility to install these manually. We recommend that you wait until all the nodes are joined into the cell. At that time, you would install the clustered applications into the cell via the deployment manager administrative console.

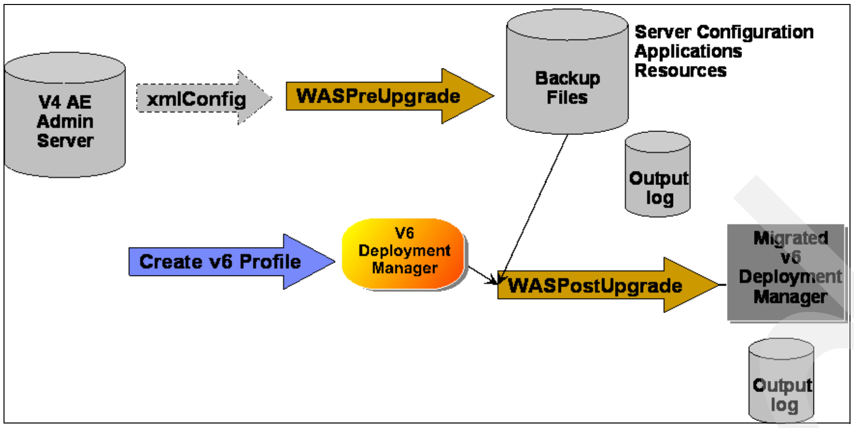


Figure 10-2 Migrating deployment manager from V4 configuration

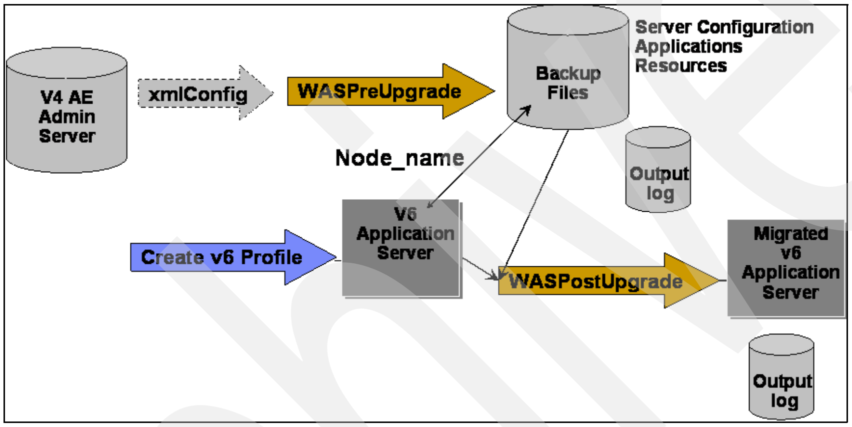


Figure 10-3 Migrating application server from V4 configuration

10.2.3 Multiple nodes for V5

WebSphere Application Server Network Deployment Version 6 has a similar architectural structure to that of WebSphere Application Server Network Deployment Version 5. Therefore, a V5 Network Deployment cell can be migrated to a V6 cell with a great deal of automation. Very few manual steps are needed. Moreover, each of the nodes of the cell can be migrated in an incremental fashion such that some of the nodes are not yet migrated, while some are, and the cell continues to operate. This concept of incremental migration, whereby you can operate a cell with nodes at different versions, is called a *mixed version cell*. A mixed version cell is flexible enough that nodes in a cooperating server cluster may be at different versions.

See 12.5, “Automatic migration: V5 Network Deployment Cluster to V6 Network Deployment Cluster” on page 214 for a detailed example of the following steps.

1. Install WebSphere Application Server Network Deployment Version 6 and create a profile with the type of deployment manager. See 8.2.3, “Profiles” on page 105 for detailed information about creating profiles.
2. Migrate the deployment manager from the V5 Network Deployment installation to the V6 deployment manager profile. When you create the profile, make sure you specify the cell name such that it matches the cell name of the V5 cell. You should also ensure that the deployment manager node name matches that of V5. Figure 10-4 on page 157 illustrates this step.

3. Start the deployment manager. See “Starting a deployment manager” on page 116 for details about how to start the deployment manager.

   When this step is complete, the cell is fully operational. The V6 deployment manager is managing the cell consisting entirely of V5 nodes.

4. If your configuration is using an external Web server, upgrade the Web server and plug-ins manually. Figure 10-5 on page 158 shows this step. At the conclusion of this step, the cell is fully operational. The upgraded Web server routes HTTP requests to V5 application servers through V6 plug-ins.

   a. Manually upgrading a Web server is a matter of installing the proper version of Web server that is approved for use with WebSphere Application Server Version 6. See Appendix A, “Prerequisite software” on page 335 for a list of approved Web servers. See 8.2, “Installation” on page 100 for an overview of installing IBM HTTP Server Version 6.

   b. Manually upgrading the Web server plug-ins is a matter of installing the WebSphere Application Server Version 6 Web Server Plug-ins as a separate install procedure. See 8.2, “Installation” on page 100 for an overview of installing Web server plug-ins.

5. Migrate each node from the V5 installation to a V6 application server node according to the detailed steps outlined for a Single Server Node in 10.2.1, “Migrating single stand-alone nodes” on page 152.

   Ensure that the node name in the V6 profile matches the node name of the V5 node that you are attempting to migrate.

   After migrating the node, start the node agent with the **startNode** command before attempting to start any of the application servers.

   Figure 10-6 on page 158 illustrates this step. The cell now consists of one or more V5 nodes, one or more V6 nodes, and a Web server routing HTTP requests to those nodes via a V6 Web server plug-in.

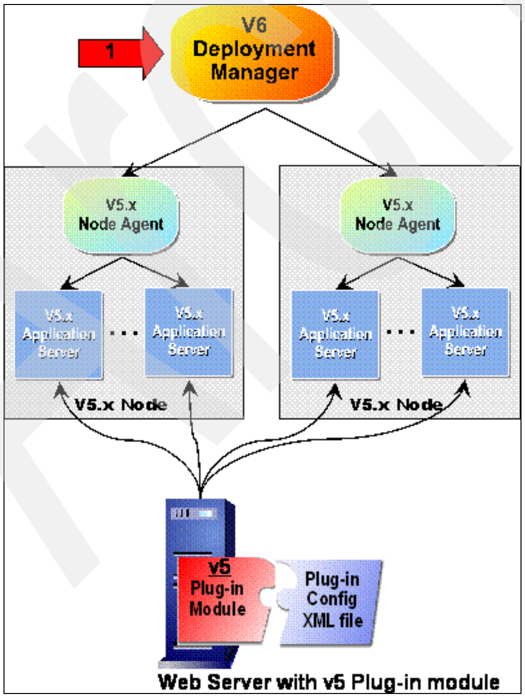


Figure 10-4 Version 5 cell migration - deployment manager

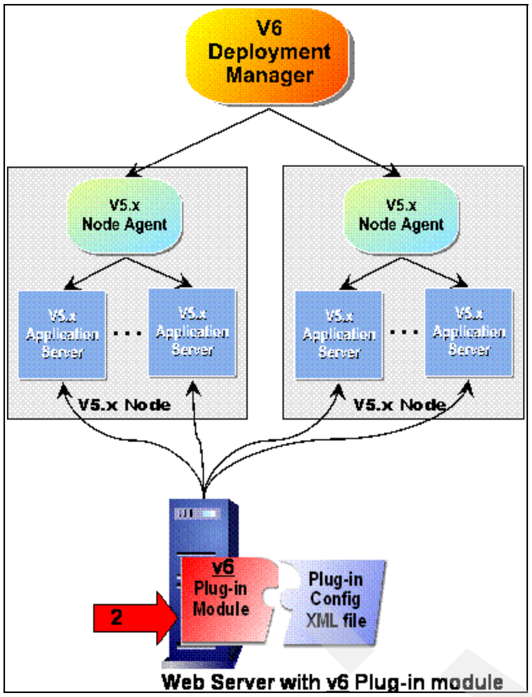


Figure 10-5 Version 5 cell migration - Web server and Plug-in

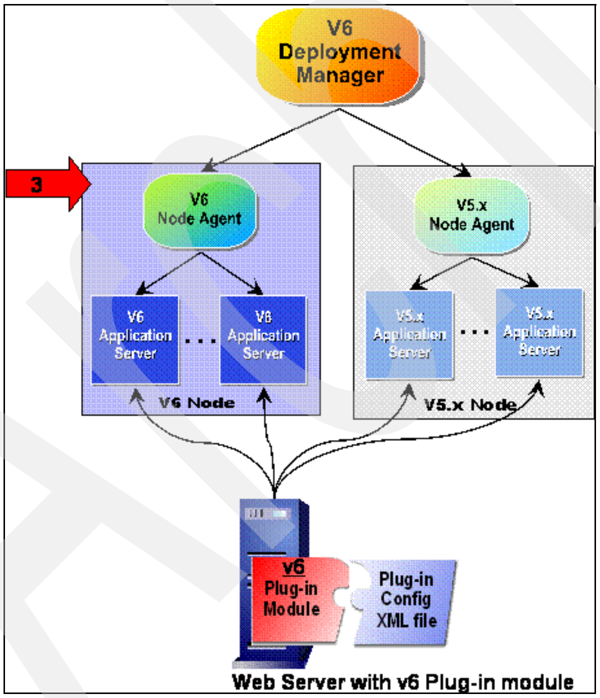


Figure 10-6 Version 5 cell migration - application server node

There are a few restrictions regarding expanding the capability of a mixed version cell. Table 10-2 summarizes the restrictions of adding resources to a cell if one or more of the nodes are still at the V5 level. The column labeled “Supported in V6.0” shows that most expansion capabilities will not work. The only instance that works is adding a V6 cluster member to a cluster that already has at least one V6 member. Note it is not possible to add a V6 cluster member to a cluster that has only V5 members. To remedy that situation, you would first migrate one of the V5 members, by migrating its node. Once a single node has been migrated, you could then manually add v6 cluster members.

When you apply the V6.0.2 refresh pack some the restrictions are relieved. The column labeled “Supported in V6.0.2” shows that which of the restrictions have been lifted. You are still not able to federate a V5 node into a mixed version cell. This also implies that you cannot federate a V5 node into a V6-only cell.

Table 10-2 Mixed version restrictions when adding resources

	New resource to add	**Supported in V6.0**	**Supported in V6.0.2**
Adding new nodes via addnode	Federate V5 node	No	No
Adding new servers	Add server to V5 node	No	Yes
Adding new cluster members	V5 server to V5-only cluster	No	Yes
	V5 server to V6-only cluster	No	No
	V6 server to V5-only cluster	No	No
	V5 server to mixed cluster	No	Yes
	V6 server to mixed cluster	Yes	Yes

10.3 Migrating nodes on z/OS platforms

WebSphere Application Server for z/OS Version 6 configurations can consist of either stand-alone nodes or full network deployment cells. Table 10-1 on page 152 shows that only nodes that are at V5.0 or V5.1 can use the automatic migration utilities. If your node is V4, then you will have to perform a manual migration, which consists of installing the new node from scratch. See Section 13.2, “Manual migration: Installing a V6 stand-alone server” on page 276 for a detailed example of manual migration of a stand-alone node.

In order to perform a successful automatic migration using the migration utilities, your V5 configuration should first be upgraded to the service levels shown in Table 10-3.

Table 10-3 Recommended service levels before migrating

Version	**Service level**
V5.0	W502025
V5.1	W510207

Be aware of the restriction that only one version of the WebSphere Application Server load modules can reside on the LPA/LNKLST at any one time because the load module names are the same between different versions. This means that if your V5.0 or V5.1 modules reside in LPA/LNKLST, then you must specify STEPLIB on the migration dialogs when specifying the V6.0 target.

10.3.1 Migrating single stand-alone nodes

The migration of a stand-alone node for the z/OS platform is conceptually very similar to that for the distributed platforms; however, the details differ. Compare the diagram in Figure 10-1 on page 153 with that in Figure 10-7. The migration utilities consist of a collection of jobs that you create using ISPF dialogs that prompt you to enter all the pertinent information.

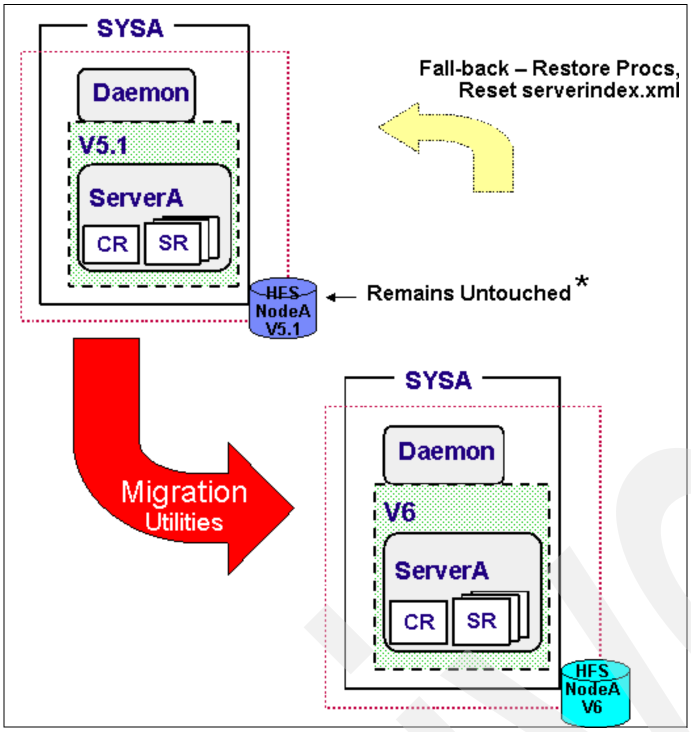


Figure 10-7 Stand-alone node migration on z/OS

The general sequence of steps that you perform to migrate a stand-alone node are:

1. Perform this preliminary work:
   a. Back up the source HFS.
   b. Back up the JCL start procedures.
   c. Check permissions of /tmp/migrate so that the migration jobs can run successfully later.

   The preliminary work you should perform is to make backups of the original configuration. The original configuration is typically not modified when the migration jobs execute successfully. However, a mistake in entering information could conceivably cause the original configuration to be unintentionally modified. Therefore, play it safe and make a backup.
2. Sequence through ISPF dialogs and enter information. Some of the key information you must enter is:
   a. Names and allocations of target data sets that hold the generated jobs
   b. Names and locations where the new configuration resides
   c. Location where the original configuration resides
   d. Procedure names
3. Stop the original node.
4. Submit jobs created by the ISPF dialogs.
5. Perform post-migration RACF work:
   a. Create new STARTED profiles if you use new JCL start procedures.
   b. Create a new keyring for the servant ID if the servant ID differs from the controller ID.
6. Start the new node.

Details on exactly how to execute the preceding steps are in Section 13.1.1, “Migrating the deployment manager node on SYSC” on page 244. Migration of a stand-alone node and that of a deployment manager node are basically the same.

10.3.2 Migrating network deployment cells

Now that you know how to migrate a stand-alone server node, migrating a network deployment cell is just a matter of migrating a collection of nodes in the proper order. The proper order of migration is:

- Migrate the deployment manager first. Details about migrating the deployment manager are in Section 13.1.1, “Migrating the deployment manager node on SYSC” on page 244.
- Migrate each application server node after the deployment manager. Details about migrating an application server node are in Section 13.1.2, “Migrating the application server node on SYSD” on page 259.

If your cell is at V5.0, then be aware of the restriction that all V5.0 nodes on the same LPAR must be migrated at the same time. To state this another way, the cell cannot operate correctly if the deployment manager is at V6.0 and any nodes are at V5.0 when they are on the same LPAR. If you have this situation, you must shut down the deployment manager and all V5.0 nodes. You would then migrate each node individually, starting with the deployment manager. When all nodes on that LPAR are done, you would then start the deployment manager and the nodes. You would then have that portion of the cell on that LPAR running all at V6.0.

If your cell is at V5.1, then this LPAR restriction does not apply. You can migrate the deployment manager and the nodes incrementally, achieving a mixed-version cell. Also, any V5.0 nodes running on an LPAR different from the deployment manager can stay at V5.0 and the cell will continue to operate correctly.

10.4 Web servers and plug-ins

Migration of Web servers and the Web server plug-in modules is a manual task that consists of a new installation of these components. You should install both of these components in order to have a correct relationship with the V6 application server.

10.4.1 IBM HTTP Server Version 6

If you are using IBM HTTP Server in your V4 or V5 WebSphere installation, you may want to upgrade to IBM HTTP Server Version 6. You have the flexibility to install IBM HTTP Server Version 6 in either of these ways:

- Upgrade your previous version in place if you no longer need the previous version. However, you should edit the httpd.conf configuration file to remove the plug-in customization lines.
- Install in a new directory if you need to run the previous version Web server at the same time. Make sure you choose port numbers that are different from the version you already have installed.

See the IBM Education assistant for more information about installing IBM HTTP Server V6, at this address:

ftp://ftp.software.ibm.com/software/eod/WAS_6-0/Install_Migration/index.html

10.4.2 Other Web server brands

If you are using a non-IBM brand of Web server, you should verify that the version you have is approved for use with WebSphere V6. See Appendix A, “Prerequisite software” on page 335 for information about Web server compatibility. Contact your Web server vendor for information about installing non-IBM Web servers.

10.4.3 Web Server Plug-ins

WebSphere V6 Web Server Plug-ins are no longer packaged with the application server installation. Web Server Plug-ins are installed with a separate installer in a separate directory. Web Server Plug-ins should be installed after installing a V6 compatible Web server.

See the IBM Education assistant for more information about installing plug-ins, at this address:

ftp://ftp.software.ibm.com/software/eod/WAS_6-0/Install_Migration/index.html

10.5 Messaging migration

If you use V5 JMS messaging resources, be aware of several migration topics that may affect you.

If you use WebSphere MQ as a JMS provider, you must update your WebSphere MQ installation to Fix Pack 8 or later. You can obtain more information about the latest fix packs at:

http://www-306.ibm.com/software/integration/mqfamily/support/summary/

If you use wsadmin administration scripts for V5 to control or create JMS resources, you may have to modify your scripts to run on V6. See 11.4, “Migrating from V5 embedded messaging” on page 170 for more details about which scripting commands are affected.

JMS resources on V6 are configured differently from V5 in the administrative console. See 8.3.2, “Messaging components” on page 113 for an overview of these changes. See 12.6, “Manual migration: Installation of J2EE 1.3 Enterprise Application on V6 Application Server” on page 226 for a detailed example of how to configure these resources.

10.6 Web Services

There are a few other considerations to take into account if you are using some Web Services in releases prior to WebSphere Application Server Version 6 and you plan to migrate their usage. These considerations are primarily in the area of migrating your environment. The scenarios described in this section are not automatically migrated. They require planning and some manual steps.

10.6.1 Migration of Web Services Gateway configuration

The configuration of Web Services Gateway has changed in WebSphere Application Server Version 6. In previous versions, it was contained outside of the WebSphere Application Server configuration and managed by a program that was not part of the administrative console. The Web Services Gateway configuration has been merged into the administrative console and profile structure in V6.0. This is one of the migration considerations when using Web Services Gateway in V6.0.

There are a couple of alternative migration strategies that can be chosen for this scenario, depending on your requirements. Coexistence is supported. This means that you can run both Version 5 and Version 6 of Web Services Gateway concurrently. If coexistence is not required then you can migrate directly from WebSphere Application Server Version 5 to WebSphere Application Server Version 6.

Coexistence support

If you have Web Services Gateways running on WebSphere Application Server Version 5, they can, subject to certain restrictions, co-exist with WebSphere Application Server Version 6 gateway instances.

Before you choose to use a mixture of V5 and V6 gateways, you should be aware of the following restrictions to co-existence:

- The V5 Web Services Gateway application is not supported on V6.0 or later application servers.
- The service integration technologies endpoint listener applications are not supported if installed on V5 application servers.

The V6 administrative console and the service integration technologies administrative commands can only be used to change the configuration of gateways running on V6 application servers. To change the configuration of a gateway running on a V5 application server, you use a Web browser to access the V5 gateway user interface.

Steps for this task

1. Save the V5 gateway configurations.
2. Migrate the cell to V6 without migrating the application servers.
3. Restore the V5 gateway configurations to the V5 application servers within the V6 cell.

Migrating a complete gateway configuration

Use the command **`migratewsgw`** to migrate an existing gateway configuration from a V5 application server to the new gateway capability on a V6 application server.

The migration procedure takes an existing gateway application whose configuration has been exported to an XML file and uses the exported XML file to configure the same gateway functionality on a V6 application server. You can migrate a V5 gateway that is in production use without stopping the gateway; requester applications can then switch over to using the new gateway configuration while the existing V5 gateway continues to run.

The considerations and number of steps to perform this migration are significant and it is best to see the following WebSphere Application Server Version 6 InfoCenter article for more information: *Migrating a complete gateway configuration*, found at:

`http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.pmc.nd.doc/tasks/twsg_coex_migrate.html`

10.6.2 Migration of Web Services Gateway applications using WS-Security

Web Services is an evolving set of specifications. The security aspect of this is called WS-Security and it is also evolving. WebSphere Application Server V5.0.2 and V5.1 support SOAP Message Security Draft 13 (formerly Web Services Security Core Specification).

WebSphere Application Server Version 6 supports SOAP Message Security Draft 13 as well as SOAP Message Security V1.0, which became the V1.0 OASIS standard. However, Web Services Gateway only supports SOAP Message Security V1.0 in the initial release of

WebSphere Application Server Version 6. All WebSphere Application Server V5.0.2 and V5.1 Web Services Gateway applications using WS-Security must be changed to use this latest version of the specification.

Note that this is considered a temporary restriction and support for SOAP Message Security Draft 13 is currently planned to be provided in a future WebSphere Application Server Version 6 maintenance release.

10.6.3 Migration to UDDI V3

The supported version of UDDI in WebSphere Application Server Version 6 is UDDI V3. This is different than prior releases of WebSphere Application Server and migration of the UDDI data is required.

You can use the process described in this section to migrate a UDDI Registry to UDDI V3, running in WebSphere Application Server Version 6, subject to the following constraints:

- Your existing registry uses a DB2 database
- Your existing registry runs in WebSphere Application Server Version 5 or later

If you are migrating from the UDDI V1.1 or 1.1.1, which runs on WebSphere Application Server V4.x, you should first migrate to UDDI V2 running on WebSphere Application Server Version 5 as described in the WebSphere Application Server V5 Information Center.

You can only run the V2 UDDI Registry, supplied in WebSphere Application Server Version 5, in a WebSphere Application Server Version 6 server if you are running in a mixed version cell migration mode. In this configuration, the WebSphere Application Server Version 6 deployment manager manages the Version 5.x nodes, and the V2 UDDI Registry is supported only for migration purposes; it is not supported for normal execution.

If you have UDDI deployed in a clustered application server and you migrate to WebSphere Application Server Version 6, you cannot run UDDI across the mixed version cluster. You can continue to run the UDDI Registry on the servers that remain at Version 5, but you cannot run the UDDI Registry in the V6 servers until all nodes in the cluster have been migrated to Version 6. This is because the UDDI data needs to be migrated from the UDDI V2 format to the UDDI V3 format.

If you are migrating the UDDI Registry from a WebSphere Application Server Network Deployment Version 5 configuration, or from a WebSphere Application Server Version 5 stand-alone application server, the steps are very similar. For a migration to a WebSphere Application Server Network Deployment configuration, one suggested option is to perform incremental migration such that you configure a mixed version cell. See 10.2.3, “Multiple nodes for V5” on page 156 for more information about configuring mixed version cells. In this way, individual application servers can be migrated when convenient in a incremental manner.

The considerations and number of steps to perform this migration are significant and it is best to see the following WebSphere Application Server Version 6 InfoCenter article for more information: *Migrating to V3 of the UDDI Registry*, found at:

```
http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.nd.doc
/info/ae/ae/twsu_migrate.html
```

10.7 Multi-broker migration

The implementation of the Data Replication Services has fundamentally changed in WebSphere Application Server Version 6, and the V5 multi-broker configuration no longer applies to the V6.0 runtime. The V5 multi-broker configuration can continue to be used for V5 nodes, and in fact is maintained by migration logic when the deployment manager is migrated from V5.x to V6.0. However, after you upgrade your deployment manager to WebSphere Application Server Version 6, you can only create new data replication domains. Any multi-broker domains that you create with a previous version of WebSphere Application Server are still functional; however, you cannot create new multi-broker domains or replicators using WebSphere Application Server Version 6 administration tools.

The different versions of application servers cannot communicate with each other. When migrating your servers to the current version of WebSphere Application Server, keep at least two application servers running on the previous version so that replication remains functional.

The considerations and number of steps to perform this migration are significant and it is best to see the following WebSphere Application Server Version 6 InfoCenter article for more information: *Migrating V6.0 servers from multi-broker replication domains to data replication domains*, found at:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_drs_migrate.html

11

Script compatibility

This chapter discusses administrative scripting compatibility issues.

11.1 Introduction

Migration of wsadmin administration scripts that use JACL and Jython from WebSphere Application Server V5 to WebSphere Application Server V6.0 should be relatively straightforward since there is a high degree of compatibility between versions. However, there are a few noted exceptions that are discussed in detail in this chapter. Migration may also be required as a result of changes to the scripting language itself since WebSphere Application Server V6.0 uses a JACL 1.3.1 engine.

For administrative scripts using **wscp** and **XMLConfig** prior to WebSphere Application Server 5.0, it is recommended that all scripts first be migrated to wsadmin syntax. See 11.6, "Version 4 wscp migration" on page 173 for information about wscp scripts.

Conceptually, the old scripts can operate under compatible support so that they continue to function. Compatibility with the old scripts is achieved by automatic migration utilities in WebSphere Application Server 6.0 that make the data locations and data objects that they operate on WebSphere Application Server 5.0 compatible, so no changes would be required to the scripts themselves. That is not to say that they are entirely forward compatible; there might be cases which require direct modification to the scripts for things to continue working smoothly. Scripts may also have to be modified at some later point in time to deal with deprecated functionality and make use of the new way of performing that same function. Learn more about how to transform your scripts by reading about the **convertScriptCompatibility** command in 8.6.4, "Script compatibility" on page 135.

11.2 Compatibility of data location

In order to remain compatible with previous versions of scripts, WebSphere Application Server 6.0 offers accessibility of data from multiple locations. The old V5 locations and the new V6.0 locations are concurrently accessible. Dual location accessibility can be accessed directly without the assistance of script compatibility support provided by the automatic migration utilities. As long as the new location is not updated, the data is accessed from the old location. Once the new location is updated, it becomes the current data and is used for further accesses and updates. Warning messages are logged when the old location is still being used.

11.2.1 Transaction logs

The location of the transaction log directory attribute has changed from the ApplicationServer::TransactionService to the ServerEntry::recoveryLogs. As long as the new location is not used, the value from the old location is used. Scripts that modify the old location can still be used; that value takes effect until a value in the new location is set. This example shows how to change the transaction log location.

JACL language example:

- Old location:

```
set transService [$AdminConfig list TransactionService $server1]
$AdminConfig showAttribute $transService transactionLogDirectory
```

- New location:

```
$AdminConfig list ServerEntry $node
set serverEntry <select one of the ServerEntry from output of above command>
set recoveryLog [$AdminConfig showAttribute $serverEntry recoveryLog]
$AdminConfig showAttribute $recoveryLog transactionLogDirectory
```

11.3 Compatibility with new object types

Some configuration objects and configuration data for WebSphere Application Server V5 may need to be migrated in such a way that they continue to be compatible with the WebSphere Application Server V5 wsadmin scripts. These changes are assisted by the compatibility mode provided by the **WASPostUpgrade** command. During migration, the default is to migrate using the compatibility mode. If this option is taken, then the old object types are migrated into the new configuration; all existing scripts do run unchanged.

See 8.6.4, "Script compatibility" on page 135 for more information about how to use script compatibility.

11.3.1 HTTP transports

The new architecture for V6 uses the new channel framework. HTTP definitions are mapped on top of this support. When the compatibility mode is chosen, the old HTTPTransport objects are migrated and mapped onto the channel architecture. Existing scripts can modify these objects and do run unchanged.

Here is an example of how to write this:

- JACL language example:

```
set server [$AdminConfig getid /Node:mynode/Server:myserv/]
set web_container [$AdminConfig list WebContainer $server]
$AdminConfig modify $web_container {{transports:HTTPTransport {{{sslEnabled true}
{sslConfig DefaultSSLSettings} {address {{host *} {port 9080}}}}}}}
$AdminConfig save
```

- Jython language example:

```
server = AdminConfig.getid('/Node:mynode/Server:myserv/')
web_container = AdminConfig.list('WebContainer', server)
AdminConfig.modify(web_container, [['transports:HTTPTransport', [[['sslEnabled',
'true'], ['sslConfig', 'DefaultSSLSettings'], ['address', [['host', '*'], ['port',
9080]]]]]]]])
AdminConfig.save()
```

11.3.2 Process definition

The name of the process definition object is changed from processDef to processDefs. You can mitigate this change by using the compatibility mode mapping provided by the automatic migration utilities. The change to scripts to use the new location is as follows:

- JACL language:
  - Old example:

```
set processDef [$AdminConfig list JavaProcessDef $server1]
set processDef [$AdminConfig showAttribute $server1 processDefinition]
```

  - New example:

```
set processDefs [$AdminConfig list JavaProcessDef $server1]
set processDefs [$AdminConfig showAttribute $server1 processDefinitions]
```

- Jython language:
  - Old example:

```
processDef = AdminConfig.list('JavaProcessDef', server1)
print processDef
```

– New example:

```
processDefs = AdminConfig.list('JavaProcessDef', server1)
print processDefs
```

11.4 Migrating from V5 embedded messaging

WebSphere Application Server 6.0 introduces a new embedded JMS engine as part of the new System Integration Bus model, the SIBJMS. Most scripts that confine themselves to acting on JMS resources do not need to be changed immediately and should continue to work unmodified with one noted exception. The exception is the V5 embedded messaging that provides the DIRECT port for publish/subscribe messaging, as set on the topic connection factory. If any Version 5 topic connection factory has the Port property set to `DIRECT`, change it to `QUEUED` before use with the Version 6 default messaging provider.

It is important to separate the JMS providers and their JMS resources from the underlying messaging system. The same script may deal with both JMS resources and the JMS server. However, concepts can be better discussed with this separation of concerns. This chapter deals only with changes to administration scripts for the V5 embedded messaging system. The migration situation for WebSphere MQ as the external messaging system or a third-party external messaging system is beyond the scope of this chapter.

When using the automatic migration utilities provided with WebSphere Application Server 6.0, almost everything is copied and continues to work without changes to the application, the runtime, and the associated administrative scripts. The automatic migration utilities take the Version 5 application server and migrate it to a Version 6 application server with the same name. This server is added as a member of a service integration bus that is named for the node on which the server is located. A messaging engine is created automatically on that bus for the application server. A default WebSphere MQ client link, called Default.MQClientLink, is created automatically for the node and assigned to the messaging engine for the application server. For each JMS queue defined on the Version 5 server, the automatic migration utilities create a new bus queue with the same name as the Version 5 JMS queue, and create a message point assigned to the messaging engine. Messages sent to the JMS queues are stored and processed at the message point. After migrating the node, the basic single-node scenario becomes:

1. The JMS application can continue to access the Version 5 JMS resources, which are now managed as V5 default messaging JMS resources implemented by the WebSphere Application Server Version 6 default messaging provider.
2. The JMS application communicates with the Version 5 JMS resources through the WebSphere MQ client link and the messaging engine. This is transparent to the JMS application.
3. The JMS resources, a JMS queue connection factory, and a JMS queue are managed as Version 5 default messaging JMS resources.
4. The new bus queue is managed as a resource of the service integration bus. Messages for JMS Q (V5) are stored and processed by the message point for the associated bus destination.
5. The WebSphere MQ client link presents itself as a queue manager and transforms between the WebSphere MQ client protocols used by Version 5 JMS applications and the WebSphere Application Server Version 6 protocols used by messaging engines. The MQ client link provides a Version 5 emulation mode within the Version 6 SI Bus framework.

11.4.1 Migration of scripts to SIB JMS

In general, it is recommended to replace the Version 5 default messaging JMS resources with equivalent Version 6 default messaging provider JMS resources as soon as it is conveniently possible (after all JMS applications using those resources have been moved onto WebSphere Application Server Version 6). You should define any new JMS resources as Version 6 resources, for example WASQueue should become SIBJMSQueue. Note that V5 requires scripts to locate the correct JMS provider, but the V6 commands hide this. In V5, the scripts use AdminConfig to directly manipulate objects. The SIBJMS objects are not intended for direct access but rather for access via **`$AdminTask`** commands.

For example:

```
$AdminConfig create WASQueue
```

becomes

```
$AdminTask createSIBJMSQueue
```

In most cases, the changes to the script will be more than a minor syntactic update since there is no direct mapping between the attributes of these objects. For details on how to achieve equivalent function, see the InfoCenter article: *AdminTask object for scripted administration*, available at:

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

Version 5.0 JMS applications accessed JMS resources through a JMS Server, which acted as one or more queue managers. The WebSphere Application Server Version 6 default messaging provider uses a fully integrated message bus and therefore needs no JMS server. In this case, we configure the Service Integration Bus to replace the logical JMS server. Note that not every bus emulates the V5 embedded messaging system but rather that it relies on specific configuration of the MQClientLink and on a particular naming convention for bus destinations.

In considering the changes that are required to migrate the scripts, the following two steps are recommended.

1. All scripts should be modified to test for the existence of the default SI Bus and create one if necessary. You cannot assume there is always an SI Bus present, as you could formerly assume that there was a JMS server present.

   If you use the automatic migration utilities, then the SI Bus is created automatically. On a stand-alone node, the JMS server runs as the jmsserver service of an application server. If you migrate a WebSphere Application Server Version 5 stand-alone node to Version 6, the Version 5 application server is migrated to a Version 6 application server with the same name. The server is added as a member of a service integration bus that is named for the node on which the server is located. A messaging engine is created automatically on that bus for the application server. In a deployment manager cell, each managed node has at most one JMS server. If you migrate a WebSphere Application Server Version 5 managed node to Version 6, the JMS server is migrated to an application server, called jmsserver, and added as a member of a service integration bus that has the same name as the node. A messaging engine is created automatically on that bus for the application server. There is only one such application server and bus for each Version 5 node. For each JMS queue defined on the JMS server, the automatic migration utilities create a new bus queue with the same name as the Version 5 JMS queue, and then create a message point assigned to the messaging engine. Messages sent to the JMS queues are stored and processed at the message point.

2. Scripts accessing the JMS server must be changed. All scripts that modify the V5 JMS Server queue lists must be changed to transform queue lists to SI Bus destinations both for a post-migration bus or any other bus that you create.

The automatic migration utilities do not migrate the data on WebSphere Application Server Version 5 queues. Any messages and knowledge of durable subscriptions held by the JMS server should be consumed before migrating the node to WebSphere Application Server Version 6. If it is not possible to have the application consume the messages, the only alternative is to write an application to run on a Version 6.0 server. This application needs to access the Version 5 JMS server, get the Version 5 messages, and resend them to applications on WebSphere Application Server V6.0.

11.4.2 Wildcard syntax conversion

Existing WebSphere Application Server Version 5 client applications using the Version 5 connection factory and destination definitions use the WebSphere MQ Integrator wildcard convention. Such applications can connect to the default messaging provider and service integration bus, and automatically have their wildcard syntax mapped to the XPath convention when subscriptions are created. Any display of these subscriptions through a Version 6 administrative interface shows the XPath syntax.

11.4.3 Replacing MDB listener ports with JMS activation specifications

A JMS application that uses a message-driven bean and its listener port in WebSphere Application Server Version 5 can continue to use the listener port without change in WebSphere Application Server Version 5. However, the message listener service uses the Application Server Facilities (ASF), which is an optional part of the JMS specification. Also, ASF is not supported by the service integration technologies on which the Version 6 default messaging provider is implemented.

The Version 6 default messaging provider is implemented as a Java Connection Architecture (JCA) resource adapter, for which inbound connectivity is configured as an activation specification.

Therefore, as soon as is conveniently possible, you should replace any listener port with a JMS activation specification for use by MDB applications with the Version 6 default messaging provider.

If you used the listener port retry count in WebSphere Application Server Version 5, there is one extra consideration. The Java Connection Architecture has no concept of a listener port retry count, so this is not supported by the Version 6 default messaging provider. This should not present a problem because the Version 6 default messaging provides destinations with a “Maximum failed deliveries” setting. This defines the maximum number of times that the service tries to deliver a message to the destination before forwarding it to the exception destination. Although applications do not need to be changed, any wsadmin or JMX scripts that make use of the listener port retry count need to be changed to make use of the “Maximum failed deliveries” setting for use in WebSphere Application Server Version 6.

11.5 JACL 1.2.6 and JACL 1.3.1 differences

A new version of JACL (1.3.1) is shipped with WebSphere Application Server Version 6. With this JACL version, the **regexp** command supports only tcl 8.0 **regexp** command syntax. If your existing V5.x JACL script uses the **regexp** command syntax that is supported in JACL 1.2.6

but not in JACL 1.3.1, you may not get a match, or you may get a compile error for your **regexp** command similar to the following:

```
com.ibm.bsf.BSFException: error while eval'ing Jacl expression:
couldn't compile regular expression pattern: ?+* follows nothing
     while executing
"regexp {(?x)
     ..."
     ("if" test expression)
     invoked from within
"if {[regexp {(?x)
     ..."
     (file testregexp.jacl line 2)
     (file line 2)
     invoked from within
"source testregexp.jacl"
```

There is no workaround for this regression problem. The JACL developers have indicated that this is by design and there is no simple patch to fix this design decision.

11.6 Version 4 wscp migration

This section discusses migrating V4 wscp scripts to the V6 wsadmin scripting language.

For the most up-to-date information about migrating wscp scripts, consult the InfoCenter article *Migrating V4.0.x administrative scripts to V6 wsadmin*, found at:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/txml_migrate4to6.html

11.6.1 Migration steps

Migration of scripts based on the earlier versions of WebSphere to wsadmin is necessary since the elements involved are very different. For similar reasons, automation would not work very well either. The good news is that wsadmin can do all that **wscp** and **xmlConfig** did, and more. The following steps help to streamline migration of scripts further:

1. Isolate the **wscp** or **XMLConfig** commands in your scripts.
2. Separate them into configuration and operational commands.

   This paves the way for better understanding of the **wsadmin** commands, since in WebSphere V6 there is a very clear distinction between runtime and configuration objects.

   Configuration commands include:

   - **Create**, **modify**, **show**, **showAll**, **install**, **uninstall**, **list**, **remove**
   - **CreateClone**, **removeClone**
   - All security configurations
   - All security role assignments

   Operational commands include:

   - **start**, **stop**, **show**, **testConnection**
   - All **DrAdmin** commands
   - **regenPluginCfg**
3. Identify corresponding configuration objects in WebSphere V6, as listed in Table 11-1 on page 174.

Table 11-1 Corresponding configuration objects in WebSphere V6.0

wscp 4.0 command	**wsadmin 6 configuration type**
ApplicationServer	Server
Context	Not applicable
DataSource	WAS40DataSource, DataSource
Domain	Not applicable
EnterpriseApp	ApplicationDeployment; also, use AdminApp
GenericServer	Server
J2CConnectionFactory	J2CConectionFactory
J2CResourceAdapter	J2CResourceAdapter
JDBCDriver	JDBCProvider
JMSConnectionFactory	JMSConnectionFactory
JMSDestination	JMSDestination
JMSProvider	JMSProvider
MailSession	MailSession
Module	ModuleDeployment; also, use AdminApp
Node	Node
ServerGroup	ServerCluster
URL	URL
URLProvider	URLProvider
VirtualHost	VirtualHost

4. Identify wsadmin attribute names.

   Attribute names have changed in wsadmin from the previous versions. These, however, are easy to map using the help functions. Get a list of the attributes that are used in the scripts and map them to the new names.

5. Convert application install commands.

   Use AdminApp installInteractive to complete one install. Then you can use the wsadmin.traceout log file to construct an install command for your script. The message WASX7278I points to all the data.

6. Convert operational commands.

   Use the information in the next section for specific version mappings. A number of common tasks have been provided as examples. These should help in finding the right operation map.

11.6.2 Migration from V4.0 to V6

Migrating scripts from V4 to V6 involves translating the script syntax to a new object command structure. The command interpreter has been renamed.

wscp

The `wscp` tool is invoked using the wscp.bat or wscp.sh program. You can specify the `-f` option to run a script file, or the `-c` option to run a command. If you specify neither, an interactive shell appears.

The operation of `wscp` is affected by a .wscprc properties file that is placed in the user's home directory. There is no default properties file shipped with the product. An alternate properties file can be specified on the command line by using the `-p` option.

wsadmin

The wsadmin tool is invoked using the wsadmin.bat or wsadmin.sh program. You can specify the **-f** option to run a script file, or the **-c** option to run a command. If you specify neither, an interactive shell appears. Multiple **-c** commands can be specified on the command line, and there are additional options that can be specified as well. For more details, refer to the InfoCenter.

Table 11-2 Mapping of wscp 4.0 to wsadmin 6.0

wscp 4.0	**wsadmin 6.0**		
action	**Object and command**	**Mbean, if any**	**Operation, if any**
server start	AdminControl startServer		
server stop	AdminControl stopServer		
servergroup start	AdminControl invoke	Cluster	start
servergroup stop	AdminControl invoke	Cluster	stop
application start	AdminControl invoke	ApplicationManager	startApplication
application stop	AdminControl invoke	ApplicationManager	stopApplication
node stop	AdminControl invoke	NodeAgent	stop
check runtime attributes	For "state," see if MBean is running; for others, AdminControl getAttribute	<any mbean>	<attributeName>
regenPluginCfg	AdminControl invoke	PluginCfgGenerator	generate
testConnection	AdminControl testConnection	DataSource	
enable security	security on command in securityProcs.jacl		
disable security	securityoff command in securityProcs.jacl		

Task-based scenarios

This section provides examples of wscp-to-wsadmin migration.

Server administration

- Start an application server:
  - wscp 4.0

    ```
    ApplicationServer start
    /Node:mynode/ApplicationServer:myserv/
    ```

  - wsadmin V6

    This command only makes sense in a Network Deployment installation. In a base installation, if you are connected to a server, you are connected to the only server and you cannot request that another be started. But in a Network Deployment installation, you can use:

    ```
    $AdminControl startServer myserv mynode
    ```

    If `mynode` is the node the client is connected to, the node name can be omitted:

    ```
    $AdminControl startServer myserv
    ```

- Stop an application server:

  This is an operational command. wscp 4.0 requires that you know the hierarchical name of the application server in question: the node name and server name. You need the same information in V6, and you also need to understand that what you are stopping is an *object* called a *Server*, not an ApplicationServer. Servers represent logical processes on many platforms (for example, Windows and AIX) and are the entities that are stopped and started. ApplicationServers are contained within Servers.

  - wscp 4.0

    ```
    ApplicationServer stop
    {/Node:mynode/ApplicationServer:Default Server/}
    ```

  - wsadmin V6

    There is more than one way to stop a server using wsadmin. The simplest method is by using the **stopServer** command on the AdminControl object:

    ```
    $AdminControl stopServer server1 mynode
    ```

    If `mynode` is the node the client is connected to, the node name may be omitted:

    ```
    $AdminControl stopServer server1
    ```

- Create a new application server:

  This is a configuration command. wscp 4.0 requires that you know the hierarchical name of the application server in question: the node name and server name. You need the same information in V6, and you also need to understand that what you are creating is an object called a Server, not an ApplicationServer.

  - wscp 4.0

    ```
    ApplicationServer create
    /Node:mynode/ApplicationServer:myserv/ -attribute
    {{Stdout myfile.out}}
    ```

  - wsadmin V6

    This can be done using a single command in wsadmin, but it is clearer to show it using two. Server objects are contained within nodes. The 4.0 ApplicationServer attribute called Stdout is replaced by the fileName attribute embedded within the outputStreamRedirect attribute of the Server.

```
wsadmin>set node [$AdminConfig getid
/Node:mynode/]
wsadmin>$AdminConfig create Server $node {{name myserv} {outputStreamRedirect
{{fileName myfile.out}}}}]
wsadmin>$AdminConfig save
```

- Remove an application server:

  This is a configuration command.

  - wscp 4.0

    ```
    ApplicationServer remove /Node:mynode/ApplicationServer:myserv/
    ```

  - wsadmin V6

    This can be done with a single command in wsadmin, but it is clearer to show it using two.

    ```
    wsadmin>set serv [$AdminConfig getid
    /Node:mynode/Server:myserv/]

    wsadmin>$AdminConfig remove $serv
    wsadmin>$AdminConfig save
    ```

- Connect to remote server:

  - wscp 4.0

    Set the wscp.host name in the default.wscprc file:

    ```
    wscp.hostname = myhost
    ```

  - wsadmin V6

    Launch the scripting program by passing in the host parameter:

    ```
    wsadmin -host myhost.austin.ibm.com
    ```

Application management

- Enterprise Application - install:

  In 4.0, wscp scripts used the **`EnterpriseApp`** and **`Module`** commands to install and uninstall applications. WebSphere V6 wsadmin scripts use the **`AdminApp`** command to perform similar configuration actions.

  - wscp 4.0

    The **`EnterpriseApp`** install command has a complex syntax, so we build up some pieces of the command beforehand:

    - **`construct -modvirtualhosts`** option

      ```
      wscp> set modhost1 [list mtcomps.war default_host]
      wscp> set modhosts  [list $modhost1]
      ```

    - **`construct -resourcereferences` option**

      ```
      wscp> set resref1 [list mtcomps.war::mail/MailSession9 mail/DefaultMailSession]

      wscp> set resref2 [list deplmtest.jar::MailEJBObject::mail/MailSession9
      mail/DefaultMailSession]

      wscp> set resrefs [list $resref1 $resref2]
      ```

- **invoke install**

  ```
  wscp> EnterpriseApp install /Node:mynode/
  c:/WebSphere/AppServer/installableApps/jmsample.ear -appname MailSampleApp
  -defappserver /Node:$mynode/ApplicationServer:myserv/ -modvirtualhosts $modhosts
  -resourcereferences $resrefs
  ```

– wsadmin V6

As with 4.0, the application install commands can have a very involved syntax. The command sequence given here accomplishes approximately the same thing as the 4.0 commands, but there are simpler ways to do this.

- **construct -MapWebModToVH** option

  ```
  wsadmin> set modtovh1 [list "JavaMail Sample WebApp" mtcomps.war,WEB-INF/web.xml
  default_host]

  wsadmin> set modtovh [list $modtovh1]
  ```

- **construct -MapResRefToEJB** option

  ```
  wsadmin> set resreftoejb1 [list deplmtest.jar  MailEJBObject
  deplmtest.jar,META-INF/ejb-jar.xml mail/MailSession9  javax.mail.Session
  mail/DefaultMailSession]

  wsadmin> set resreftoejb2 [list "JavaMail Sample WebApp" ""
  mtcomps.war,WEB-INF/web.xml mail/MailSession9  javax.mail.Session mail/bozo]

  wsadmin> set resreftoejb [list $resreftoejb1 $resreftoejb2]
  ```

- Construct attribute string

  ```
  wsadmin> set attrs [list -MapWebModToVH $modtovh -MapResRefToEJB $resreftoejb
  -node mynode -server myserv -appname MailSampleApp]

  #     invoke install
  wsadmin> $AdminApp install c:/WebSphere/AppServer/installableApps/jmsample.ear
  $attrs
  ```

- Install default/standard applications

  The standard EAR files that ship with the product can be installed (in the federation of the node into a cell, where the apps have to be installed specifically) as follows:

  ```
  $AdminApp install
  \"c:/<WAS_HOME>/samples/lib/TechnologySamples/TechnologySamples.ear\" {-appname
  TechnologySamples -cell node1Network -node node1 -usedefaultbindings"
  ```

The Help utility can be used to get more information for these commands. In this case, there are other commands that offer more information about the options. The **AdminApp**, **taskInfo** and **options** commands are particularly helpful in listing all the options and information about them. You can also use the **AdminApp installInteractive** command (with the same options as the **install** command) to be prompted for each option and step through the install, setting your values. This command with all its options is logged in the wsadmin.traceout file in the logs directory, under message WASX7278I.

- Enterprise Application - uninstall:

  – wscp 4.0

  ```
  wscp> EnterpriseApp remove {/EnterpriseApp:Sample Application/}
  ```

  – wsadmin V6

  ```
  wsadmin> $AdminApp uninstall TechnologySamples
  ```

- Enterprise Application - edit:
  - wscp 4.0

    ```
    wscp>  EnterpriseApp modify {/EnterpriseApp:Trade Sample/} -attribute {{Name
    changedName}}
    ```

  - wsadmin V6

    ```
    wsadmin> $AdminApp edit DefaultApplication {-BindJndiForEJBNonMessageBinding
    {{"Increment Enterprise Java Bean" "Increment" "Increment.jar,META-INF/ejb-jar.xml"
    "Increment1"}}}
    ```

While testing, it is easier to use **`editInteractive`** and get a list of the options and the variables that can be changed. Then use those with the **`edit`** command in your scripts. In the previous command, -**`BindJndiForEJBNonMessageBinding`** is the option we are using to change the JNDI name of the ejb Increment, with the URI Increment.jar,META-INF/ejb.jar.xml.

Administration of runtime objects: query, change

This change is temporary and reverts back to the old setting in the event of server stoppage. To permanently change it, the configured object has to be changed. However, this setting can be used to get trace outputs.

```
wsadmin> set trace [lindex [$AdminControl queryNames type=TraceService,*],0]
wsadmin> $AdminControl setAttribute $trace traceSpecification com.ibm.*=all=enabled
```

Administration of configured objects

- Modifying attributes of application servers, enterprise applications, and other configured objects:
  - wscp 4.0

    ```
    wscp> ApplicationServer modify /Node:dev-pc/ApplicationServer:myServer/ \
    -attribute {{PingInterval 120}}
    ```

  - wsadmin V6

    ```
    wsadmin> set trace [$AdminConfig getid /TraceServer:/
    wsadmin> $AdminConfig modify trace
    {{startupTraceSpecification=com.ibm.ws.*=all=enabled}}
    wsadmin> $AdminConfig save
    ```

  The change in the configuration is seen only when the server is restarted.

  If you made changes and do not want to save them, then you can call the **`reset`** command:

  ```
  wsadmin> $AdminConfig reset
  ```

  There are some other useful commands, such as:

  - `wsadmin> $AdminConfig hasChanges -`

    This can be invoked before a save to check for changes to the configuration.

  - `wsadmin> $AdminConfig setSaveMode rollbackOnConflict -`

    This is called to set the save mode.

  Refer to the InfoCenter for more details about these commands.

- List the configured server groups:

  This is a configuration command in V6.

  - wscp 4.0

    ```
    ServerGroup list
    ```

You could put the result of this command into a JACL list and invoke other operations such as **show** or **modify** on the members of the list.

- wsadmin V6

  Again, in V6 there are server clusters instead of server groups.

  ```
  $AdminConfig list ServerCluster
  ```

  Again, you can put the result of this command into a JACL list and invoke other config commands such as **show** or **modify** on the members of the list. In order to invoke operational commands such as **stop**, you need to do a little more work:

  - Get the config ID of the cluster of interest:

    ```
    set myclust [$AdminConfig getid /ServerCluster:mycluster/]
    ```

  - The name returned can be used to get the ObjectName of the running Cluster MBean:

    ```
    set runningCluster = [$AdminConfig getObjectName $myclust]
    ```

  At this point, runningCluster has the object name for the running instance of the ServerCluster (or it is empty if it is not running). This object name can be used for control purposes:

  ```
  $AdminControl invoke $runningCluster stop
  ```

- Display help:

  - wscp 4.0

    To obtain general help:

    ```
    wscp> help
    ```

    To obtain help on a particular object:

    ```
    wscp> object_type help
    ```

    or:

    ```
    wscp> ApplicationServer help
    ```

  - wsadmin V6

    Displaying general help is done using the $Help object:

    ```
    wsadmin>$Help help
    ```

    This lists all the options available. In general, attributes and operations when passing in an Mbean are the most useful. To get the Mbean associated, you have to use the following command:

    ```
    wsadmin>$AdminControl queryNames type=Server,*
    ```

- List actions available on configured objects:

  - wscp 4.0

    Here, with any object, you can get help by specifying:

    ```
     "<objName>  help operations"

    wscp> ApplicationServer help operations
    ```

  - wsadmin V6

    Since we are dealing with configured objects here, the wsadmin object to be used is AdminConfig.

    ```
    $AdminConfig attributes Server
    ```

Cluster administration

- Create a server group:

  WebSphere 4.0 server groups are no more; we now have server clusters in their place. The members of a cluster are not “clones” so much as they are servers with identical application configurations.

  – wscp 4.0

    This example assumes that there already exists an application server named as1 that is to be used as the first clone in a server group.

    ```
    wscp>ServerGroup create /ServerGroup:sg1/ -baseInstance
    /Node:mynode/ApplicationServer:as1/
    -serverGroupAttrs {{EJBServerAttributes
    {{SelectionPolicy roundrobin}}}}
    ```

  – wsadmin V6

    ```
    wsadmin>set serverid [$AdminConfig getid /Node:mynode/Server:as1/]
    wsadmin>$AdminConfig convertToCluster $serverid MyCluster
    ```

- Clone a ServerGroup:

  – wscp 4.0

    ```
    wscp>ServerGroup clone /ServerGroup:sg1/ -cloneAttrs
    {{Name newServer}} -node /Node:mynode/
    ```

  – wsadmin V6

    The following three commands could be combined into a single nested command, but it is more clear to show them as three separately. The first gets the cluster ID, the second gets the node ID, and the last command creates a new member of an existing cluster.

    ```
     wsadmin>set mycluster [$AdminConfig getid
    /ServerCluster:mycluster/]

    wsadmin>set mynode [$AdminConfig getid
    /Node:mynode/]

    wsadmin>$AdminConfig createClusterMember $mycluster
    $mynode {{memberName newServer}}
    ```

- Start a ServerGroup:

  This is an operational command.

  – wscp 4.0

    ```
    ServerGroup start /ServerGroup:cluster1/
    ```

  – wsadmin V6

    Again, these two commands could be combined into one. We look up the Cluster MBean, then invoke **start** on it:

    ```
    wsadmin>set cl1 [$AdminControl completeObjectName type=Cluster,name=cluster1,*]

    wsadmin>$AdminControl invoke $cl1 start
    ```

    It is possible that there is no running MBean for cluster1 if the cluster has not been loaded first. If this is the case, use the ClusterMgr MBean to retrieve the cluster:

    ```
    wsadmin>set clustermgr [$AdminControl
    completeObjectName type=ClusterMgr,*]
    ```

```
wsadmin>$AdminControl invoke $clustermgr
retrieveCluster cluster1
```

- Stop a ServerGroup:

  This is an operational command.

  – wscp 4.0

  ```
  wscp>ServerGroup stop /ServerGroup:cluster1/
  ```

  – wsadmin V6

  Again, these two commands could be combined into one. We look up the Cluster MBean, then invoke **start** on it:

  ```
  wsadmin>set cl1 [$AdminControl completeObjectName type=Cluster,name=cluster1,*]
  wsadmin>$AdminControl invoke $cl1 stop
  ```

- Remove a clone:

  There is no command for removing a server from a cluster once it has been added. You must delete the server.

Other tasks

- Set the Trace specification:

  – wscp 4.0

  ```
  wscp> DrAdmin remote <portno> -setTrace com.ibm.ejs.*=all=enabled
  ```

  – wsadmin V6

  There are two ways to set tracing with wsadmin in WebSphere V6. One takes immediate effect, but is temporary and is set on the runtime, using **AdminControl**.

  ```
  wsadmin> set ts [$AdminControl queryNames
  type=TraceService,node=<nodeName>,process=<serverName>,*]
  wsadmin> $AdminControl setAttribute $ts traceSpecification com.ibm.*=all=enabled
  ```

  However, if you want your change to persist, then you should change the configuration by using **AdminConfig**.

  ```
  wsadmin> set svr [$AdminConfig getid /Node:<nodeName>/Server:<serverName>/]
  wsadmin> set ts [$AdminConfig list TraceService $svr]
  wsadmin> $AdminConfig modify $ts {{startupTraceSpecification com.ibm.*=all=enabled}}
  ```

  There is also a way to change the TraceLog specifications:

  ```
  wsadmin> set svr [$AdminConfig getid /Node:<nodeName>/Server:<serverName>/]
  wsadmin> set ts [$AdminConfig list TraceService $svr]
  wsadmin> set trlog [$AdminConfig list TraceLog $ts]
  wsadmin> $AdminConfig modify $trlog {{fileName myFile.log} {maxNumberOfBackupFiles
  10} {rolloverSize 2}}
  ```

- Enable security:

  – wscp 4.0

  ```
  wscp>SecurityConfig setAuthenticationMechanism LOCALOS -userid {me secret}
  wscp>SecurityConfig enableSecurity
  ```

  – wsadmin V6

  V6 ships a profile that is referenced in the default wsadmin.properties file so that it runs by default. This is securityProcs.jacl, and it implements various security configuration functions. As of this writing, it only supports two functions: securityon and securityoff.

  ```
  securityon [user [password]]
  ```

This turns on local security. If a user name is supplied, it is set in the security config. If both a user name and password are supplied, the securityon function checks the validity of the user/password combination, and fails the function if the combination is not valid.

So the equivalent V6 command is:

```
securityon userName pwd
```

- Disable local OS security:
  - wscp 4.0

    ```
    SecurityConfig disableSecurity
    ```

  - wsadmin V6

    V6 ships a profile that is referenced in the default wsadmin.properties file so that it runs by default. This is securityProcs.jacl, and it implements various security configuration functions. It only supports two functions: securityon and securityoff.

    ```
    securityoff
    ```

- Install JDBCDrivers:
  - wscp 4.0

    ```
    wscp>JDBCDriver create /JDBCDriver:mydriver/ -attribute {{ImplClass
    COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource}}
    wscp>JDBCDriver install /JDBCDriver:mydriver/ -node /Node:mynode/ -jarFile
    c:/SQLLIB/java/db2java.zip
    ```

  - wsadmin V6

    There is no separate install step in WebSphere V6. The JAR file name required in the V4.0 install step is replaced by the classpath attribute on the V6 JDBCProvider object.

    In WebSphere V6, resources can exist at the server, node, or cell level of the configuration. The V6 equivalent of providing the “node” in the 4.0 JDBCDriver install command is that we create our V6 JDBCProvider at the “node” level.

    ```
    wsadmin>set node [$AdminConfig getid /Node:mynode/]

    wsadmin>$AdminConfig create JDBCProvider $node {{classpath
    c:/SQLLIB/java/db2java.zip} {implementationClassName
    COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource} {name
     mydriver}}
    ```

- Ping node and AppServers for current status:
  - wscp 4.0

    The point of this task is generally to determine if a given server is running:

    ```
    wscp>set servers [ApplicationServer list]
    wscp>foreach server $servers {
       set result ApplicationServer show $server -attribute
    {CurrentState}
       puts "state for server $server: $result"
    }
    ```

  - wsadmin V6

    This task is a little more involved in WebSphere V6 because configuration and control commands are separated. When we get a list of all the servers in the first statement of the following code, we are getting a list of all the servers defined in the configuration. This is just configuration data; we cannot interrogate the configuration data to find out what servers are running because the config data only contains configuration information. The running state of a server is not configuration information. So for each server defined, we ask for the ObjectName of any running instance of the server. If the server is not running, nothing is returned from the **getObjectName** command on the AdminConfig object. If the server is running, we ask for its state attribute. This may appear superfluous; if the Mbean is there, the server is running and the state should be `STARTED`. It is possible, however, for the state to be something other than `STARTED`, for example, `STOPPING`.

```
set servers [$AdminConfig list Server]
foreach server $servers {
   set objName [$AdminConfig getObjectName $server]
   if {[llength $objName] == 0} {
      puts "server $server is not running"
   } else {
      set result [$AdminControl getAttribute $objName state]
      puts "state for server $server: $result"
   }
}
```

- Test connections to data sources:

  WebSphere 4.0 introduced a means for testing the connection to a DataSource object; this function is available in WebSphere V6 as well.

  – wscp 4.0

  ```
  wscp>set myds /JDBCDriver:mydriver/DataSource:myds/
  wscp>DataSource testConnection $myds
  ```

  – wsadmin V6

  In WebSphere V6, the **testConnection** command is part of the AdminControl object because it is an operational command. But this particular operational command takes a configuration ID as an argument, so we invoke the **getid** command on the AdminConfig object to get it:

  ```
  wsadmin>set myds [$AdminConfig getid /JDBCProvider:mydriver/DataSource:myds/]
  wsadmin>$AdminControl testConnection $myds
  ```

  In many cases, a user ID and password (or other properties) are required to complete the test connection. If this is the case, you get message WASX7216E describing the missing properties:

  ```
  WASX7216E: 2 attributes required for testConnection are missing: "[user, password]"
  To complete this operation, please supply the missing attributes as an option,
  following this example: {{user user_val} {password password_val}}
  ```

  In this case, you need to issue the commands as follows:

  ```
  set myds [$AdminConfig getid /JDBCProvider:mydriver/DataSource:myds/]
  $AdminControl testConnection $myds {{user myuser} {password secret}}
  ```

- Regenerate the plug-in configuration:

  – wscp 4.0

  ```
  wscp> Node regenPluginCfg /Node:myNode/
  ```

  – wsadmin V6

  ```
  wsadmin>set cfg [$AdminControl queryNames type=PluginCfgGenerator,*]
  wsadmin>$AdminControl invoke $cfg generate "C:/was50/WebSphere/AppServer
  C:/was50/WebSphere/AppServer/config node1Network node1 server1 genPlugin.txt"
  ```

12

Distributed runtime application migration examples

This chapter show examples of migrating applications and server configurations for the distributed platforms.

12.1 Overview of migration examples

The migration examples shown in this chapter are simple ones that are intended to illustrate the detailed steps of moving existing applications, application resources, and the server environment settings to a V6 WebSphere installation.

The applications are simple examples that either are used in our internal education classes or ship with the WebSphere system. The application design is not important in itself. However, we wish to show how data sources and JMS sources are migrated. We chose a simple EJB banking application that also uses an MDB interface. For the cases where the starting configuration is V4, we chose the same application but without the MDB interface. We do not show the details of configuring the database back end of the data source since database configuration is not relevant to this book. We also do not show the detailed operation of the sample application at the end of the exercise because these details are not important. However, we did test the application at the end of the exercise to prove that the procedure was correct. The starting point is a working application on a source environment. The ending point is a working application on a target environment running on WebSphere V6.

Four of the five examples illustrate the use of the automatic migration utilities discussed in 8.6, “Automatic migration utilities” on page 125. These examples show the detailed use of the automatic migration utilities and then examine the migrated artifacts on the target V6 environment.

The fifth example shows a case of manual migration. The automatic migration utilities are not used. Instead, the detailed steps of installing and configuring the application using the administrative console are shown.

These are the examples:

1. Automatic migration: V4 AE Single Server to V6 Application Server

   This example starts with the J2EE 1.2 application consisting of a Web module and a EJB 1.1 module installed on V4 AE Single Server. The automatic migration utilities migrate the application and resources to a stand-alone application server.

2. Automatic migration: V5 Application Server to V6 Application Server Express

   This example starts with the J2EE 1.3 application consisting of a Web module and an EJB 2.0 module installed on V5.1 application server. The automatic migration utilities migrate the application and resources to a stand-alone application server. An IBM HTTP Server V6 is installed and configured.

3. Automatic migration: V4 AE Clustered Server Group to V6 Network Deployment Cell

   This example starts with a J2EE 1.2 application installed on a server group in V4 AE. The automatic migration utilities migrate the application and resources to a group of V6 stand-alone application servers. The servers are added to a V6 Network Deployment cell and the servers are configured in a cluster.

4. Automatic migration: V5 Network Deployment Cell to V6 Network Deployment Cell

   This example starts with a J2EE 1.3 application installed on a V5 Network Deployment cell containing clustered servers. The automatic migration utilities migrate the application and resources to a V6 cell. The deployment manager and server nodes are migrated in a staged sequence illustrating the concept of mixed version cells.

5. Manual Migration: Installing J2EE 1.3 Enterprise Application on V6 Application Server

   This example starts with a V6 stand-alone application server. The J2EE 1.3 application is installed and the data source and JMS resources are configured manually using the administrative console.

The examples in this chapter are based on the Windows operating system., Other operating systems have small variations in some terminology. If you are using an operating system other than Windows, you may need to make adjustments related to the following issues:

- File system path names have a different syntax
- Script names use a different suffix
- The profile creation wizard has a different name for each platform

12.2 Automatic migration: V4 AE Single Server to V6

This example shows how to use the migration utilities to migrate WebSphere Application Server Advanced Edition Single Server to a WebSphere Application Server V6 environment. We tested V4.0.1, but any V4 fix level should work. The application we used to illustrate the migration techniques is a J2EE 1.2 application we call MyBank. The application is a very simple simulation of a banking application. The EJB module contains a CMP bean and a session bean. The CMP bean uses a data source that points to a database of a single table. The database is implemented on IBM DB2 Version 8.1.

Preparing your existing environment for migration

You should perform any application backups before performing a migration of your existing environment. The migration utilities simply make a copy of the key files on the existing system. There are no changes made on the source configuration, and therefore, there is no explicit reason that the source configuration would be modified in any way. However, it is always good practice to back up a working configuration before undertaking any major event.

Reviewing application and resources on your existing environment

The particular operation of the application is not important to this exercise. However, it is useful to show the application and resources that we intend to test.

Figure 12-1 shows the administrative console and the MyBank application.

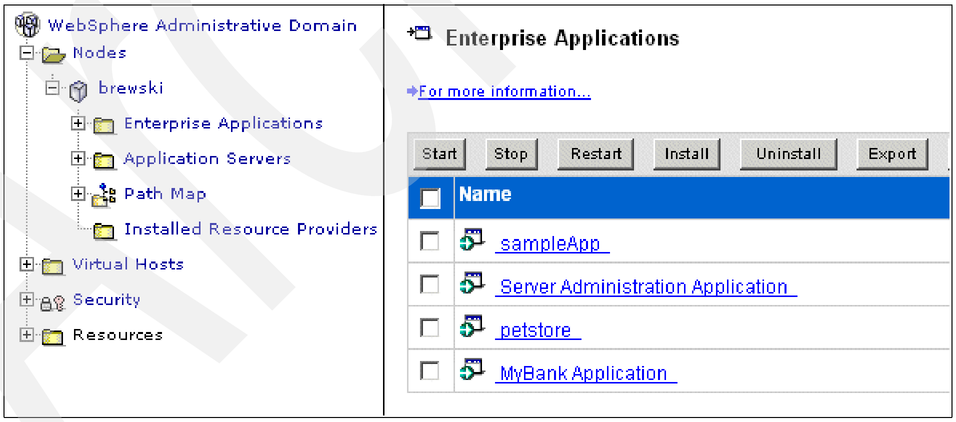
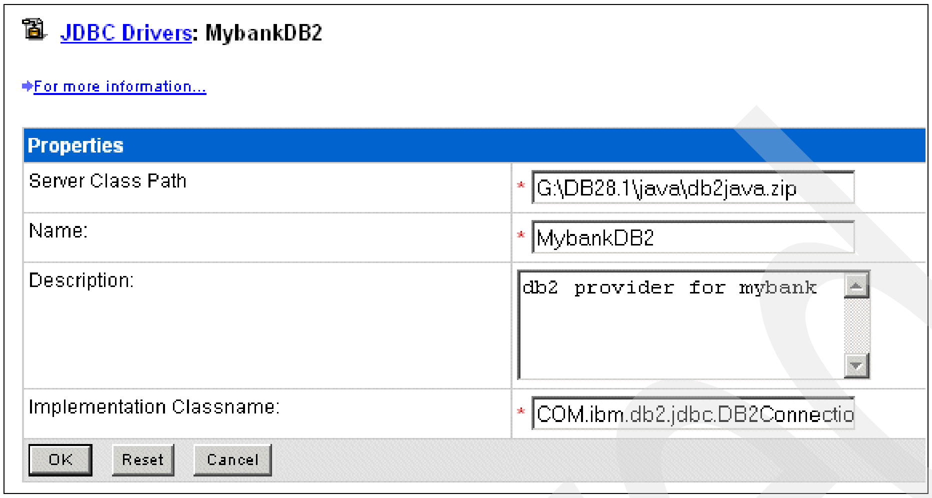


Figure 12-1 Version 4 applications

Figure 12-2 shows the JDBC Provider that MyBank uses, and Figure 12-3 shows the data source.

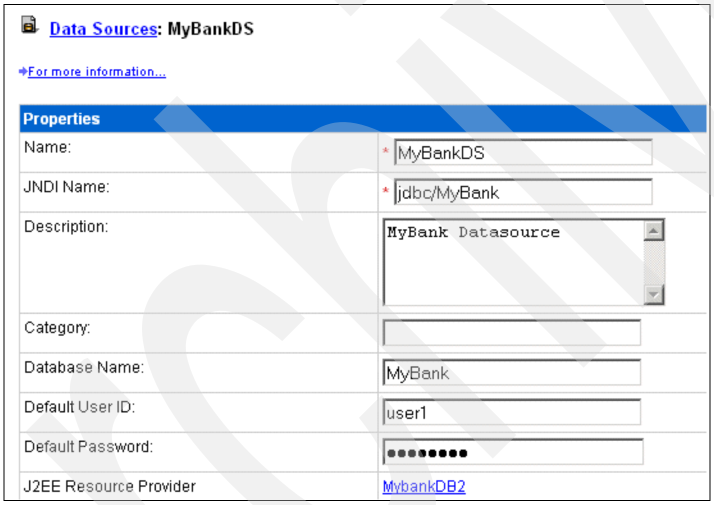


JDBC Drivers: MybankDB2

For more information...

Properties	
Server Class Path	* G:\DB28.1\java\db2java.zip
Name:	* MybankDB2
Description:	db2 provider for mybank
Implementation Classname:	* COM.ibm.db2.jdbc.DB2Connectio

Figure 12-2 Version 4 JDBC drivers



Data Sources: MyBankDS

For more information...

Properties	
Name:	* MyBankDS
JNDI Name:	* jdbc/MyBank
Description:	MyBank Datasource
Category:	
Database Name:	MyBank
Default User ID:	user1
Default Password:	●●●●●●●●
J2EE Resource Provider	MybankDB2

Figure 12-3 Version 4 data source

Preparing the V6 installation

You will need to install and configure WebSphere V6 according to the instructions found on line in this article:

ftp://ftp.software.ibm.com/software/eod/WAS_6-0/Install_Migration/index.html

When you create the V6 profile, you have the flexibility to specify any node name or cell name that you desire. Neither the cell name nor the node name is crucial in making this exercise work; however, the node name and cell names are important in the other exercises that we discuss in this chapter. The important name to record is the profile name because this profile name will be specified in the steps involving the auto-migration utilities.

Running automatic migration utilities

For this migration exercise, we are illustrating migrating such that Version 4 and Version 6 do not reside on the same system. This is possible because the installation media contains a **`WASPreUpgrade`** command that can be run directly from the installation media without first installing WebSphere V6. In the following example, we have the installation media mounted as drive H: and execute the command from this drive.

```
set WAS_BIN=h:\migration\bin
%WAS_BIN%\WASPreUpgrade.bat G:\BACKUPS\brewski_AESV4 G:\WebSphereV4\AppServer
```

After running the **WASPreUpgrade** command, check its status by consulting the output log, which will be located in the backup directory. The following example shows the last few lines of the log, which indicate successful completion.

```
File - G:\BACKUPS\brewski_AESV4\ WASPreUpgrade.Mon-Oct-11-10.01.35-2004.log
           deleted lines for brevity
MIGR0303I: The existing WebSphere Application Server environment is saved.
MIGR0420I: The first step of migration completed successfully.
```

The **WASPreUpgrade** command created the G:\BACKUPS\brewski_AESV4 directory as its output. Transfer the contents of this directory to the second system where WebSphere V6 is installed. The means of transfer is not important. We happened to use the technique of zipping the directory, then transferred the zip file via FTP, and finally unzipped to a directory called E:\BACKUPS\brewski_AESV4.

Once this backup directory has been transferred, run the **WASPreUpgrade** command using the directory as input, and the V6 profile as a container for the output. The following example shows the commands. The **WASPostUpgrade** command is run from the V6 installation directory.

```
call e:\WASv6\bin\setupCmdline.bat
set WAS_BIN=%WAS_HOME%\bin
set BACKUPDIR=E:\BACKUPS\brewski_AESV4
set PROFILE=olepaint_v4_mig
%WAS_BIN%\WASPostUpgrade %BACKUPDIR% -profileName %PROFILE%
```

After the **WASPostUpgrade** command completes, review the output log for errors or warnings. The following example shows a sample warning message. Since this warning pertains to the JDBC driver for the application data source we are migrating, we show later that we do indeed heed the warning and change the classpath as suggested.

```
File E:\WASv6\profiles\olepaint_v4_mig\logs\WASPostUpgrade.Mon-Oct-11-10.39.42-2004.log

MIGR0260W: JDBCProvider MybankDB2 has been added to the configuration, but the class path
parameter might need to be updated if its prerequisite has changed.
```

Reviewing application and resources on V6

After running the migration utilities and verifying that the utilities ran correctly, the next step is to start the Version 6 application server and verify that the application and its resources have been migrated and are intact.

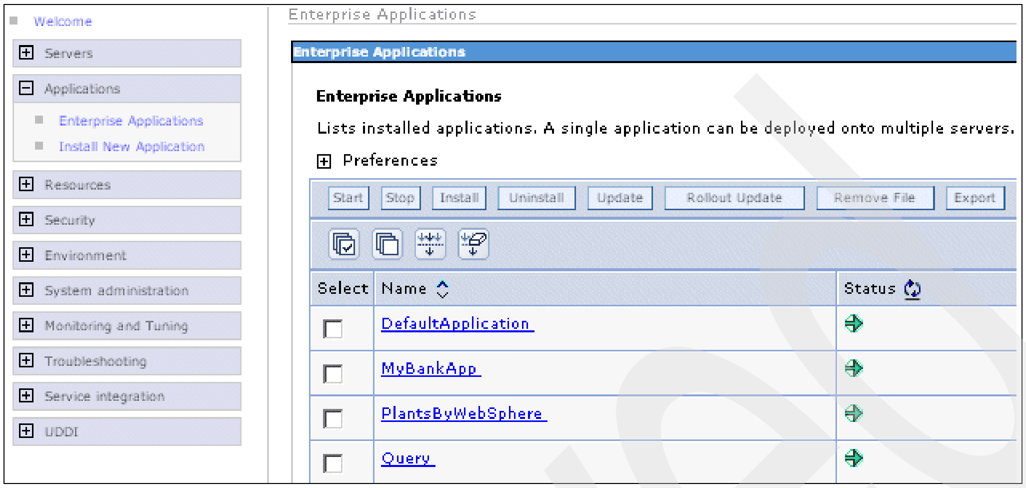
Change the directory to the profile location containing the bin directory and start the server from that directory. Our location is shown in this example:

```
cd E:\WASv6\profiles\olepaint_v4_mig\bin
startServer server1
```

Your directory would be different. Note that WebSphere Application server is installed at E:\WASv6 and we are using the profile named olepaint_v4_mig.

Once the server has started, observe the application and resources in the administrative console. Start the console; the default URL is `http://localhost:9060/ibm/console/`. Your console URL may be different if you configured a non-default port.

Figure 12-4 shows the MyBankApp application has been migrated to the new profile. Figure 12-5 shows the JDBC driver resource. Note that the JDBC DB2 driver resided in a different directory, and we changed the classpath to update this new path.

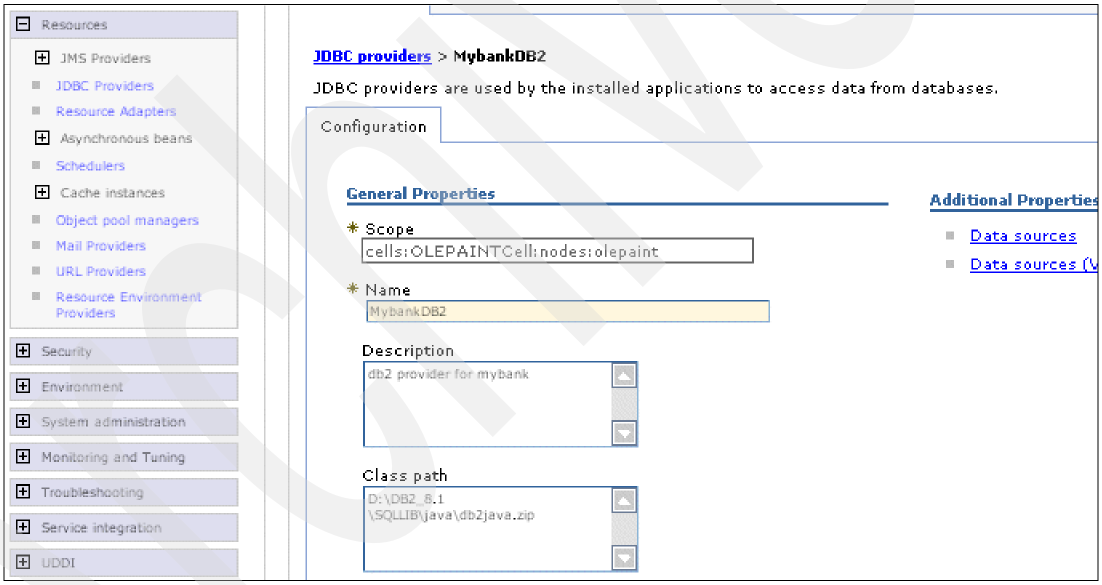


Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
□	DefaultApplication	➔
□	MyBankApp	➔
□	PlantsByWebSphere	➔
□	Query	➔

Figure 12-4 Version 6 application view



JDBC providers > MybankDB2

JDBC providers are used by the installed applications to access data from databases.

* Scope
cells:OLEPAINTCell:nodes:olepaint

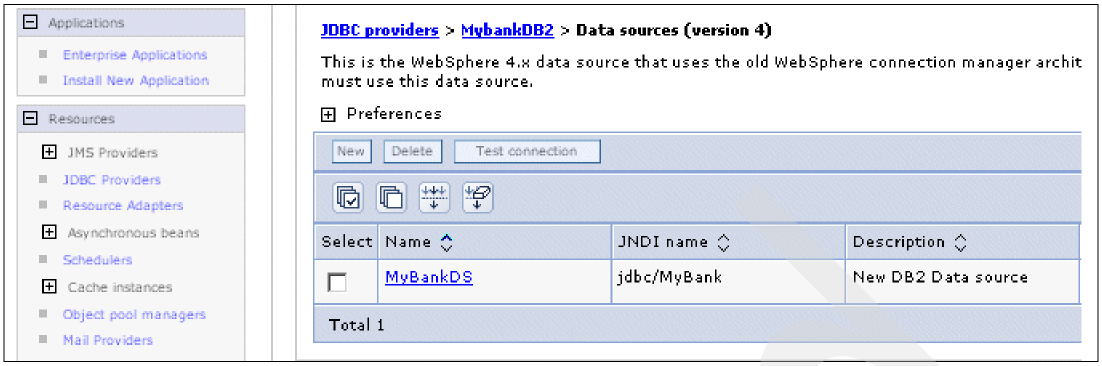
* Name
MybankDB2

Description
db2 provider for mybank

Class path
D:\DB2_8.1
\SQLLIB\java\db2java.zip

Figure 12-5 Version 6 JDBC provider

Figure 12-6 on page 191 shows the V4 data source. Note that since the data source was migrated from V4, its has V4 semantics and is found under the V4 data source link.



JDBC providers > MybankDB2 > Data sources (version 4)

This is the WebSphere 4.x data source that uses the old WebSphere connection manager archit must use this data source.

⊞ Preferences

Select	Name	JNDI name	Description
□	MyBankDS	jdbc/MyBank	New DB2 Data source
Total 1			

Figure 12-6 Migrated Version 4 data source

Confirm the application is functioning correctly with the appropriate application test scenarios.

12.3 Automatic migration: Application Server V5 to Express V6

This example shows how to use the migration utilities to migrate WebSphere Application Server V5.1 to a WebSphere Application Server Express V6 environment. We tested V5.1, but any V5 fix level should work. The application we used to illustrate the migration techniques is a more complex variation of the MyBank application used in 12.3, “Automatic migration: Application Server V5 to Express V6” on page 191. This variation we have named MyBankMDB. This application has added an MDB interface and therefore must use JMS resources. One of the interesting aspects of this example is to show how the JMS resources are migrated.

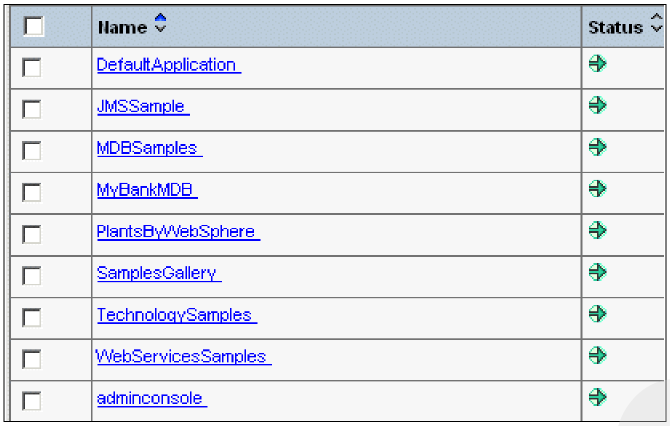
12.3.1 Preparing your existing environment for migration

Perform any application backups before performing a migration of your existing environment. The migration utilities simply make a copy of the key files on the existing system. There are no changes made on the source configuration, and therefore, there is no explicit reason that the source configuration would be modified in any way. However, it is always good practice to back up a working configuration before undertaking any major event.

12.3.2 Reviewing application and resources on your existing environment

The particular operation of the application is not important to this exercise. However, it is useful to show the application and resources that we intend to test.

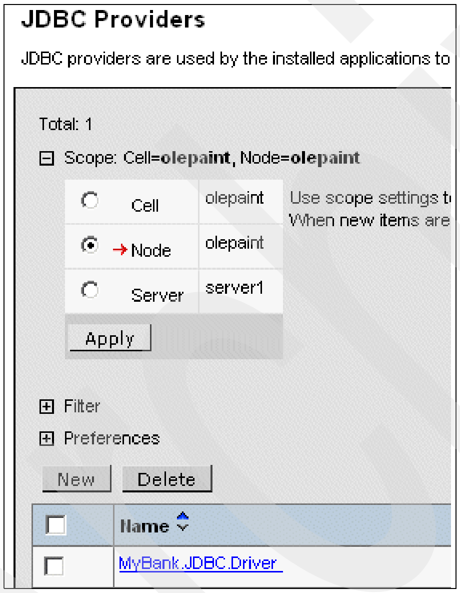
Figure 12-7 on page 192 shows the applications we are using. The application that we are testing is MyBankMDB.



□	Name	Status
□	DefaultApplication	➜
□	JMSSample	➜
□	MDBSamples	➜
□	MyBankMDB	➜
□	PlantsByWebSphere	➜
□	SamplesGallery	➜
□	TechnologySamples	➜
□	WebServicesSamples	➜
□	adminconsole	➜

Figure 12-7 Version 5 applications

Figure 12-8 shows the JDBC resources that the application uses.



JDBC Providers

JDBC providers are used by the installed applications to

Total: 1

⊟ Scope: Cell=**olepaint**, Node=**olepaint**

○ Cell	olepaint	Use scope settings t When new items are
● →Node	olepaint	
○ Server	server1	

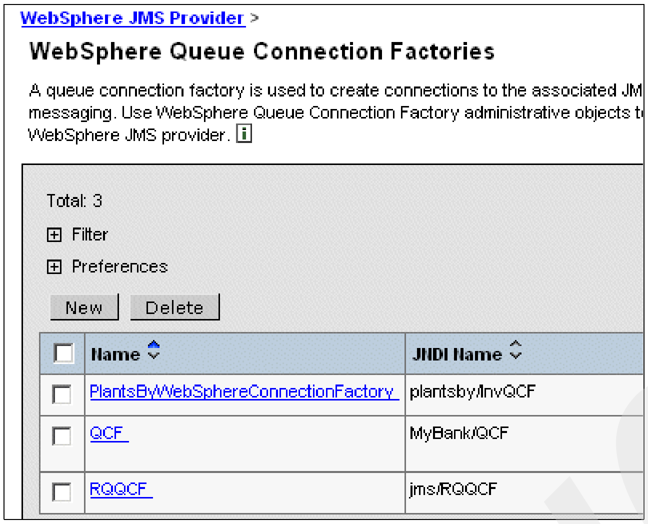
⊞ Filter

⊞ Preferences

□	Name
□	MyBank JDBC Driver

Figure 12-8 Version 5 JDBC Provider

Figure 12-9 shows the WebSphere JMS Provider queue connection factory, which is QCF for our application.



WebSphere JMS Provider >

WebSphere Queue Connection Factories

A queue connection factory is used to create connections to the associated JM messaging. Use WebSphere Queue Connection Factory administrative objects t WebSphere JMS provider.

Total: 3

⊞ Filter

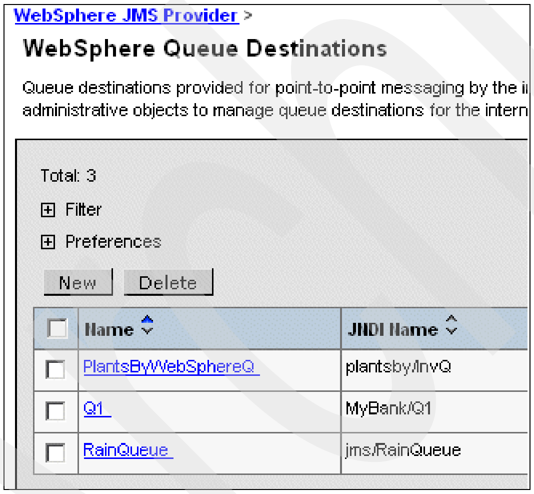
⊞ Preferences

New Delete

□	Name	JNDI Name
□	PlantsByWebSphereConnectionFactory	plantsby/InvQCF
□	QCF	MyBank/QCF
□	RQQCF	jms/RQQCF

Figure 12-9 Version 5 Queue Connection Factory

Figure 12-10 shows the WebSphere JMS Provider queue destination, which is Q1 for our application.



WebSphere JMS Provider >

WebSphere Queue Destinations

Queue destinations provided for point-to-point messaging by the i administrative objects to manage queue destinations for the intern

Total: 3

⊞ Filter

⊞ Preferences

New Delete

□	Name	JNDI Name
□	PlantsByWebSphereQ	plantsby/InvQ
□	Q1	MyBank/Q1
□	RainQueue	jms/RainQueue

Figure 12-10 V5 Queue Destination

12.3.3 Preparing the V6 installation

WebSphere Application Server Express V6 is installed by starting the **Launchpad** command as shown in 8.2.1, “Launchpad” on page 101. The defaults are taken for all the server port assignments, as shown in Figure 8-8 on page 103. The target directory location of the installation is overridden in order to make typing the directory name straightforward, but this directory location is not crucial.

12.3.4 Running automigration utilities

For this migration exercise, we are illustrating migrating such that Version 5.1 and Version 6 reside on the same system. This is in contrast to the scenario described in 12.2, “Automatic migration: V4 AE Single Server to V6” on page 187, where the two installations are on different systems. The case where both installations are on the same system simplifies the required steps.

In the following example, we have installed WebSphere Application Server Express V6 under the directory E:\WebSphere_V6.

```
call E:\WebSphere_V6\bin\setupCmdline.bat
set WAS_BIN=%WAS_HOME%\bin
set OLDWASHOME=d:\WebSphere\AppServer
set BACKUPDIR=E:\BACKUPS\20041029
%WAS_BIN%\WASPreUpgrade %BACKUPDIR% %OLDWASHOME%
```

After running the **WASPreUpgrade** command, check its status by consulting the output log, which will be located in the backup directory. The following example shows the last few lines of the log, which indicate successful completion.

```
File - E:\BACKUPS\20041029\WASPreUpgrade.Fri-Oct-29-08.29.07-2004.log
            First several lines deleted
MIGR0303I: The existing WebSphere Application Server environment is saved.
MIGR0420I: The first step of migration completed successfully.
```

The **WASPreUpgrade** command creates the E:\BACKUPS\20041029 directory as its output.

You run **WASPreUpgrade** using the directory as input, and the V6 profile as a container for the output. The following example shows the commands. The **WASPostUpgrade** command is run from the V6 installation directory.

```
call E:\WebSphere_V6\bin\setupCmdline.bat
set WAS_BIN=%WAS_HOME%\bin
set BACKUPDIR=E:\BACKUPS\20041029
%WAS_BIN%\WASPostUpgrade %BACKUPDIR%
```

Please note that the **WASPostUpgrade** command in this example only uses the single required argument of the backup directory. The profile name is defaulted because we are operating on the default profile.

After the **WASPostUpgrade** command completes, review the output log for errors or warnings. The following example shows a sample warning message. This warning message is quite common and can often be ignored. Examine any warnings about port assignments to determine whether the assignment will conflict with other servers.

```
file E:\WebSphere_V6\profiles\default\logs\WASPostUpgrade.Fri-Oct-29-08.35.11-2004.log

MIGR0331W: Port 9080 in file
E:\WebSphere_V6\profiles\default\config\cells\OLEPAINTNode01Cell\nodes\olepaint\servers\ser
ver1\server.xml is migrated, but is already assigned in file
com.ibm.websphere.migration.postupgrade.ApplicationServerHelper$PortMapping@6ca5260a; this
situation might result in port conflicts.
```

12.3.5 Reviewing application and resources on V6

After running the migration utilities and verifying that the utilities ran correctly, the next step is to start the V6 application server and verify that the application and its resources have been migrated and are intact.

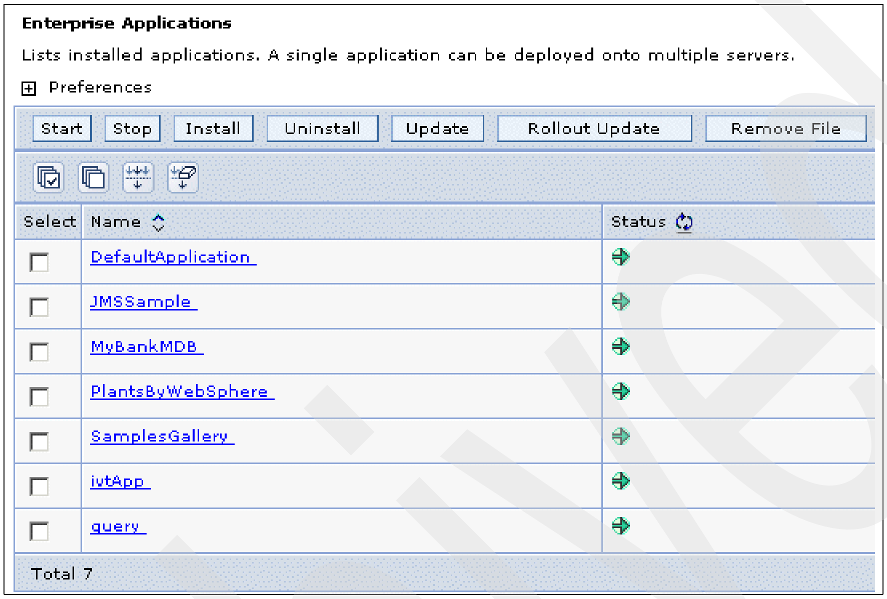
Change the directory to the profile location containing the bin directory and start the server from that directory. Our location is shown here:

```
cd /d E:\WebSphere_V6\bin\profiles\default\bin
startServer server1
```

Your directory would be different.

Once the server has started, observe the application and resources in the administrative console. Start the console according to default URL `http://localhost:9060/ibm/console/`. Your console URL may be different if you configured a non-default port.

Figure 12-11 shows the MyBankMDB application has been migrated to the new profile.



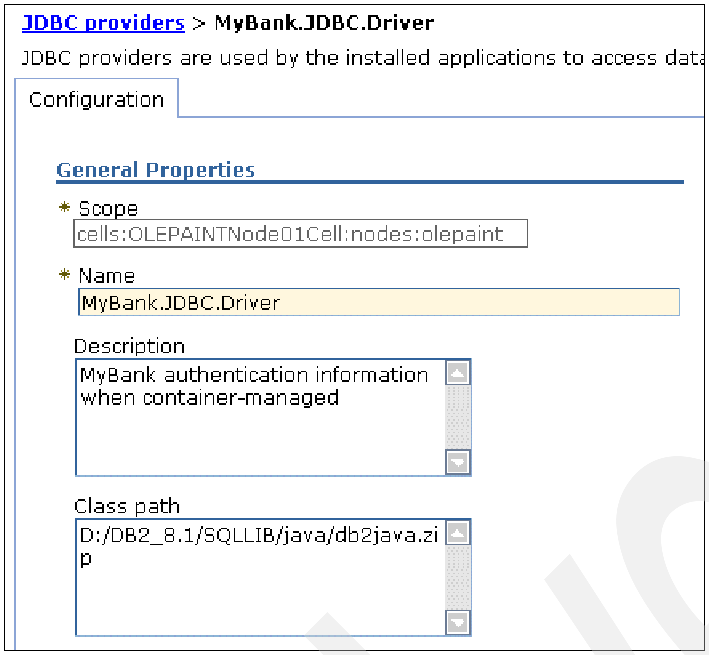
Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
□	DefaultApplication	➔
□	JMSSample	➔
□	MyBankMDB	➔
□	PlantsByWebSphere	➔
□	SamplesGallery	➔
□	ivtApp	➔
□	query	➔
Total 7		

Figure 12-11 Applications list

The JDBC driver and JDBC data source are shown in Figure 12-12 and Figure 12-13 on page 196. Note that the classpath for the JDBC provider was not changed for this example because we are using the same system to do the migration. Also note that this is a default V5 type data source that is migrated. Contrast this normal data source with the V4 data source that is migrated in the example in Figure 12-6 on page 191.



JDBC providers > MyBank.JDBC.Driver

JDBC providers are used by the installed applications to access data

Configuration

General Properties

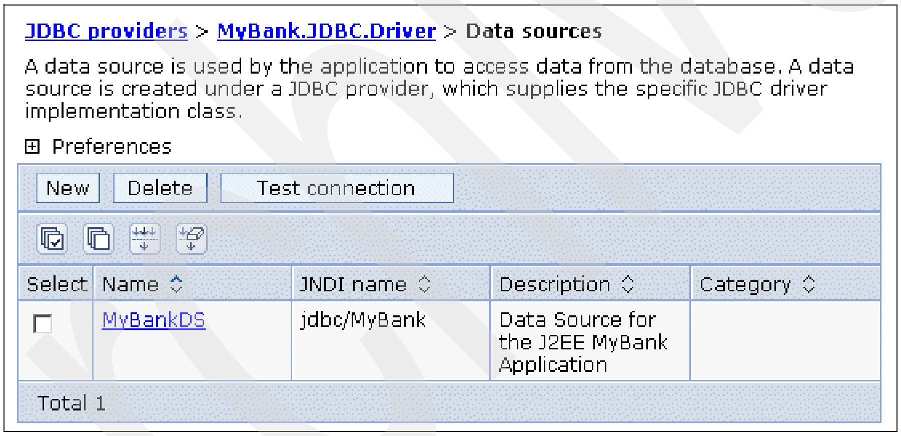
* Scope
cells:OLEPAINTNode01Cell:nodes:olepaint

* Name
MyBank.JDBC.Driver

Description
MyBank authentication information when container-managed

Class path
D:/DB2_8.1/SQLLIB/java/db2java.zip

Figure 12-12 Example V5 JDBC driver



JDBC providers > MyBank.JDBC.Driver > Data sources

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

⊞ Preferences

Select	Name ◇	JNDI name ◇	Description ◇	Category ◇
□	MyBankDS	jdbc/MyBank	Data Source for the J2EE MyBank Application	

Total 1

Figure 12-13 Example V5 data source

System Integration Bus (SIBus) is the first JMS resource to examine. A default SIBus is created by the migration utilities in which the name matches the node name. Figure 12-14, Figure 12-15 and Figure 12-16 on page 197 show the SIBus that is created.

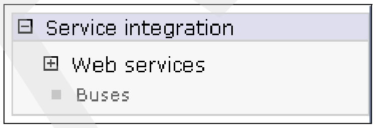
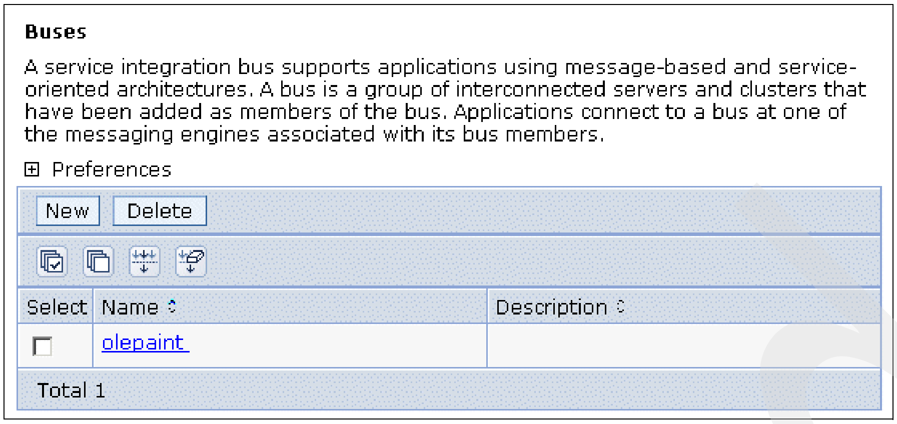


Figure 12-14 Service integration resources



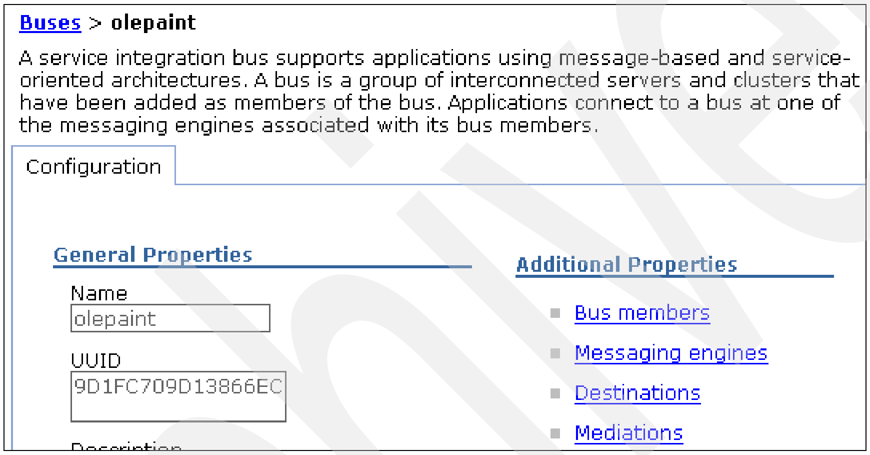
Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

⊞ Preferences

Select	Name	Description
☐	olepaint	
Total 1		

Figure 12-15 Service integration buses



Buses > **olepaint**

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

Name: olepaint

UUID: 9D1FC709D13866EC

Description

Figure 12-16 Bus general properties

The JMS resources that are found in V5 that are required by the example applications are queue connection factories and queue destinations. These resources are located under JMS providers identified as V5 default messaging, as shown in Figure 12-17, Figure 12-18, and Figure 12-19 on page 198.

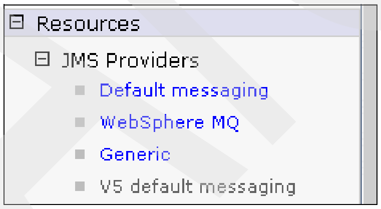
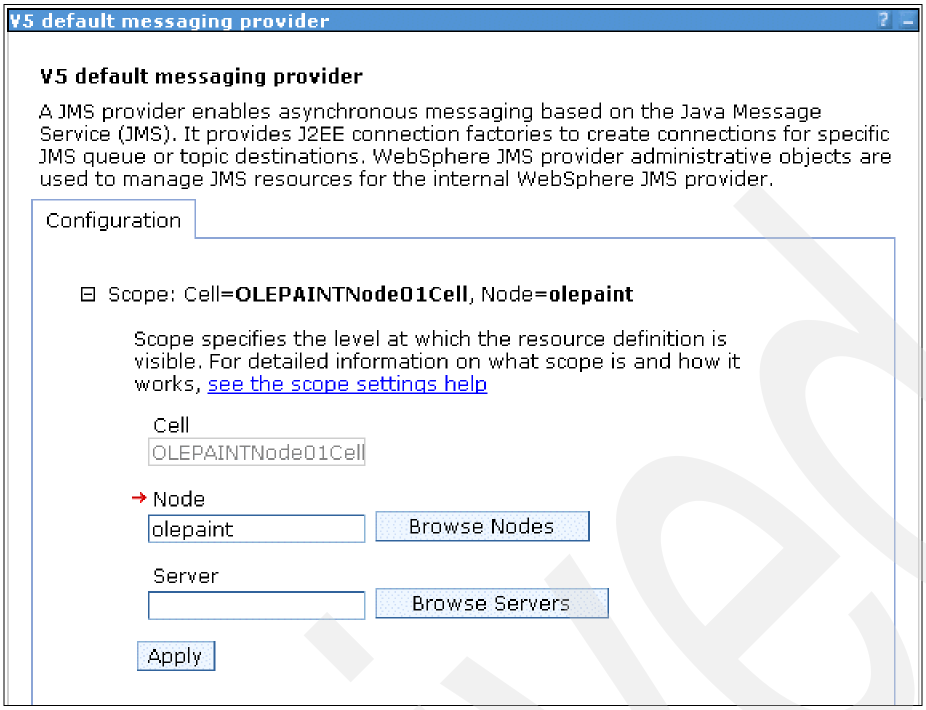


Figure 12-17 JMS provider resources



V5 default messaging provider

V5 default messaging provider

A JMS provider enables asynchronous messaging based on the Java Message Service (JMS). It provides J2EE connection factories to create connections for specific JMS queue or topic destinations. WebSphere JMS provider administrative objects are used to manage JMS resources for the internal WebSphere JMS provider.

Configuration

⊟ Scope: Cell=**OLEPAINTNode01Cell**, Node=**olepaint**

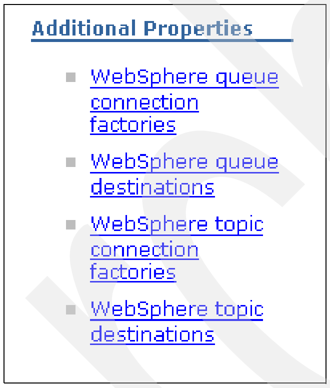
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

Cell
OLEPAINTNode01Cell

→ Node
olepaint

Server

Figure 12-18 v5 default messaging provider

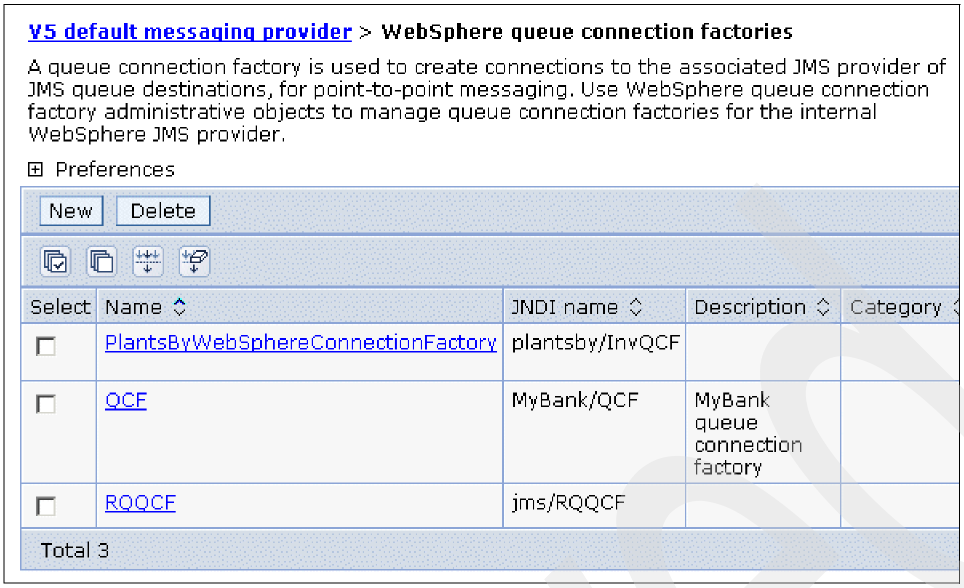


Additional Properties

- WebSphere queue connection factories
- WebSphere queue destinations
- WebSphere topic connection factories
- WebSphere topic destinations

Figure 12-19 JMS provider additional properties

The applications in the example require queue connection factories and queue destinations. Figure 12-20 and Figure 12-21 on page 199 show these resources.



V5 default messaging provider > **WebSphere queue connection factories**

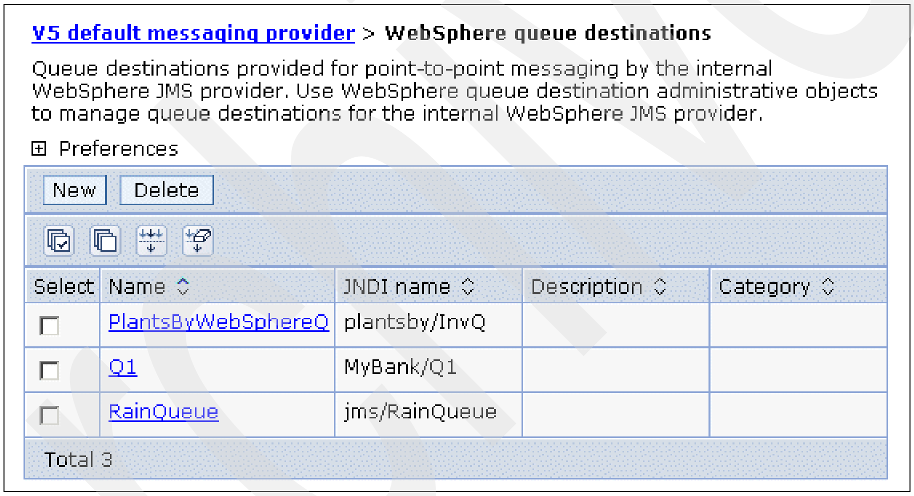
A queue connection factory is used to create connections to the associated JMS provider of JMS queue destinations, for point-to-point messaging. Use WebSphere queue connection factory administrative objects to manage queue connection factories for the internal WebSphere JMS provider.

⊞ Preferences

Select	Name	JNDI name	Description	Category
□	PlantsByWebSphereConnectionFactory	plantsby/InvQCF		
□	QCF	MyBank/QCF	MyBank queue connection factory	
□	RQQCF	jms/RQQCF		

Total 3

Figure 12-20 Queue connection factory



V5 default messaging provider > **WebSphere queue destinations**

Queue destinations provided for point-to-point messaging by the internal WebSphere JMS provider. Use WebSphere queue destination administrative objects to manage queue destinations for the internal WebSphere JMS provider.

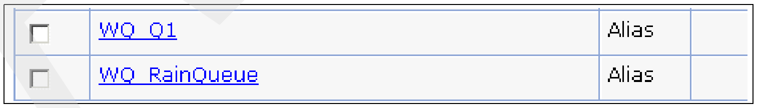
⊞ Preferences

Select	Name	JNDI name	Description	Category
□	PlantsByWebSphereQ	plantsby/InvQ		
□	Q1	MyBank/Q1		
□	RainQueue	jms/RainQueue		

Total 3

Figure 12-21 Queue destination

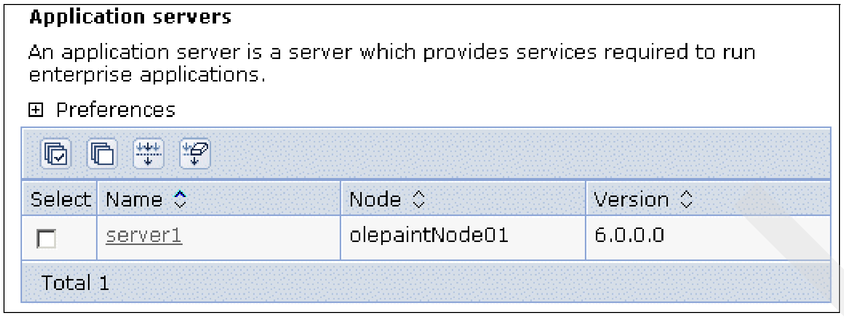
Queue destinations also have references under the SIBus destinations. Figure 12-22 shows the queue destinations. Note that the SIBus destination prefixes the V5 default messaging queue destination with the characters WQ_.



□	WQ_Q1	Alias	
□	WQ_RainQueue	Alias	

Figure 12-22 Queue destinations under SI Bus

The MDB listener ports are defined under the application server. The migration utilities migrate both the MDB listener port and the application listener port binding. Figure 12-23, Figure 12-24, Figure 12-25, Figure 12-26, and Figure 12-27 on page 200 illustrate navigating under the server definition to find the listener port definition.



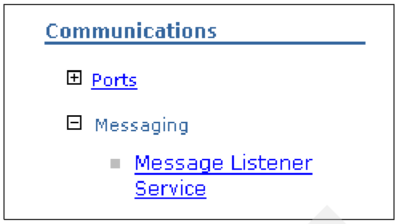
Application servers

An application server is a server which provides services required to run enterprise applications.

⊞ Preferences

Select	Name	Node	Version
□	server1	olepaintNode01	6.0.0.0
Total 1			

Figure 12-23   Application servers



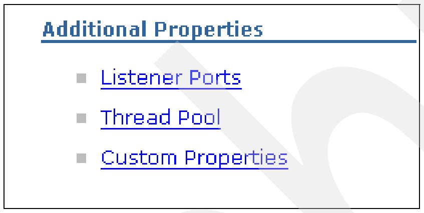
Communications

⊞ Ports

⊟ Messaging

- Message Listener Service

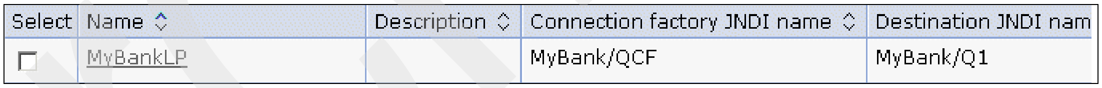
Figure 12-24   Message listener service under application servers



Additional Properties

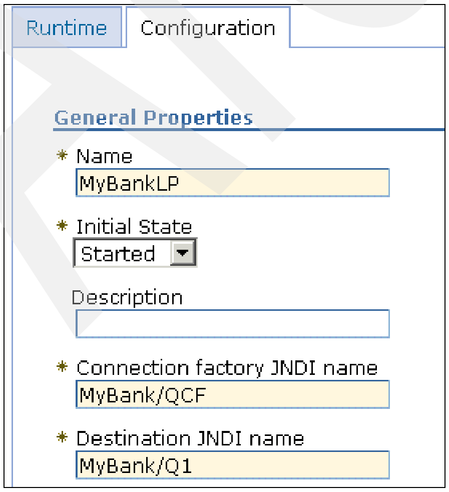
- Listener Ports
- Thread Pool
- Custom Properties

Figure 12-25   Message listener service properties



Select	Name	Description	Connection factory JNDI name	Destination JNDI nam
□	MyBankLP		MyBank/QCF	MyBank/Q1

Figure 12-26   Listener port for MyBank application



General Properties

* Name
MyBankLP

* Initial State
Started

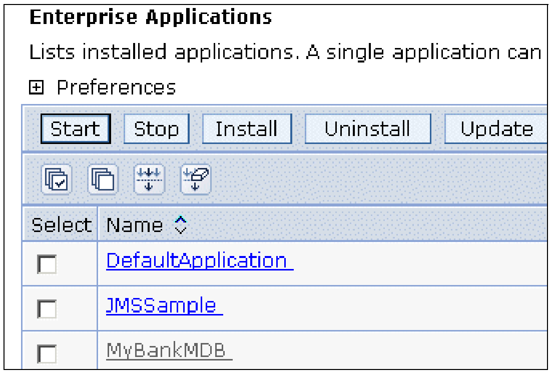
Description

* Connection factory JNDI name
MyBank/QCF

* Destination JNDI name
MyBank/Q1

Figure 12-27   General properties for listener port

The application bindings are also bound to the same MDB listener port. Figure 12-28, Figure 12-29, and Figure 12-30 show the listener port attributes.



Enterprise Applications

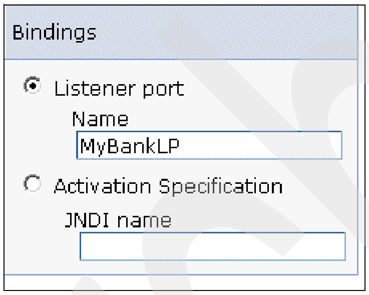
Lists installed applications. A single application can

⊞ Preferences

Select	Name
□	DefaultApplication
□	JMSSample
□	MyBankMDB

Figure 12-28 Migrated example applications

Figure 12-29 Listener port bindings under each application



Bindings

● Listener port

Name: MyBankLP

○ Activation Specification

JNDI name:

Figure 12-30 Listener port bindings values

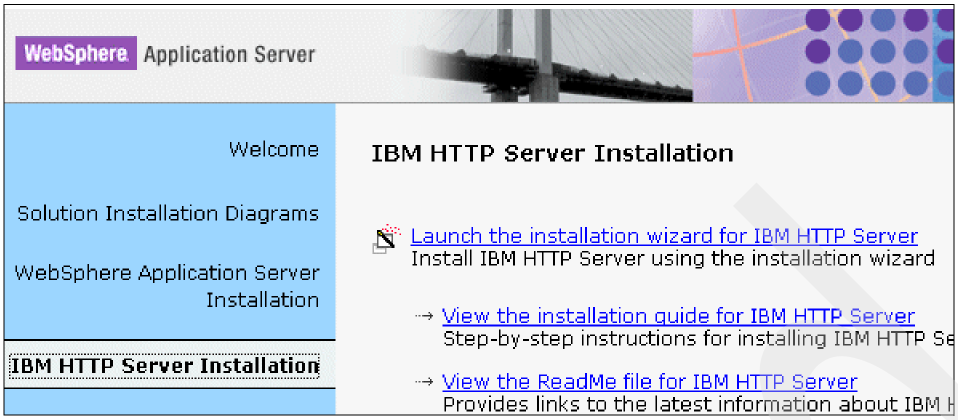
12.3.6 Migrating Web server and WebSphere plug-ins

A single server scenario may optionally use a Web server in the configuration. WebSphere plug-in management has been enhanced for WebSphere Application Server Version 6 and consequently the installation and configuration of the Web server and plug-ins is a bit more complex than on previous versions. See “Web Server plug-ins” on page 105 for an overview of WebSphere plug-ins.

The example shows the process of installing IBM HTTP Server V6 along with the required plug-in on the same system as the application server. This example, where everything is installed on the same system, represents a typical configuration in a developer test environment.

Before beginning the installation of the Web server, ensure that the application server is stopped.

Install the IBM HTTP Server V6 by launching the launchpad and selecting **IBM HTTP Server Installation**. Then select the **Launch the installation wizard** link. See 8.2.1, “Launchpad” on page 101 for an overview of the launchpad. Figure 12-31 on page 202 shows the launchpad screen for this step.


WebSphere Application Server
Welcome
Solution Installation Diagrams
WebSphere Application Server Installation
IBM HTTP Server Installation

IBM HTTP Server Installation

Launch the installation wizard for IBM HTTP Server
Install IBM HTTP Server using the installation wizard

→ View the installation guide for IBM HTTP Server
Step-by-step instructions for installing IBM HTTP Se

→ View the ReadMe file for IBM HTTP Server
Provides links to the latest information about IBM H

Figure 12-31 Launchpad selecting installation of IBM HTTP Server

Follow the prompts in the Web server installation wizard. Specify the Web server installation directory, typical default installation, and decide if you want the Web server to be installed as a Windows service.

At the conclusion of the installation of the Web server, you are given the option to run the WebSphere plug-ins installation wizard. Figure 12-32 shows the introductory screen where you can choose to see more detailed installation instructions.

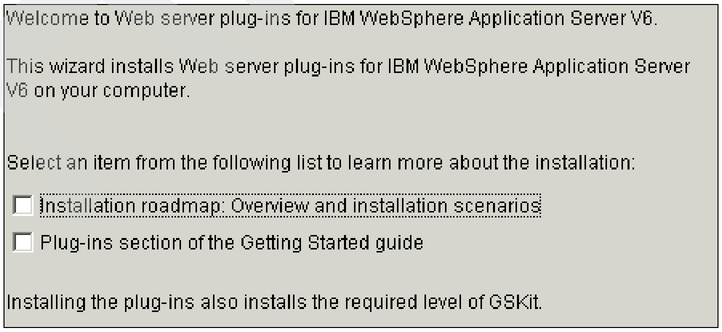
After accepting the license agreement, Figure 12-33 on page 203 shows the type of Web server, which is the IBM HTTP Server V6 for this example.

Figure 12-34 on page 203 shows the remote configuration selection screen where we select **Remote**. A Remote selection seems to contradict the actual example configuration, so an explanation is in order.

The meaning of *Remote* in the selection screen is that the Web server and the application server are remote from each other on separate systems. The meaning of *Local* in the selection screen is that the Web server is on the same system as the application server. If Local is chosen, the WebSphere plug-in installer creates a managed Web server instance in the application server to manage the plug-in configuration files, which only works with the IBM HTTP Server V6 Web server. We select the **Remote** case so that this configuration does not occur.

The plug-ins are installed on the same system as the Web server by necessity because the plug-in is executable code, which must be loaded by the Web server.

.



Welcome to Web server plug-ins for IBM WebSphere Application Server V6.

This wizard installs Web server plug-ins for IBM WebSphere Application Server V6 on your computer.

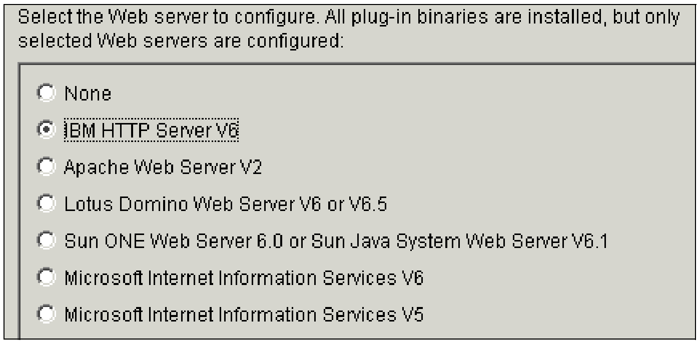
Select an item from the following list to learn more about the installation:

☐ Installation roadmap: Overview and installation scenarios

☐ Plug-ins section of the Getting Started guide

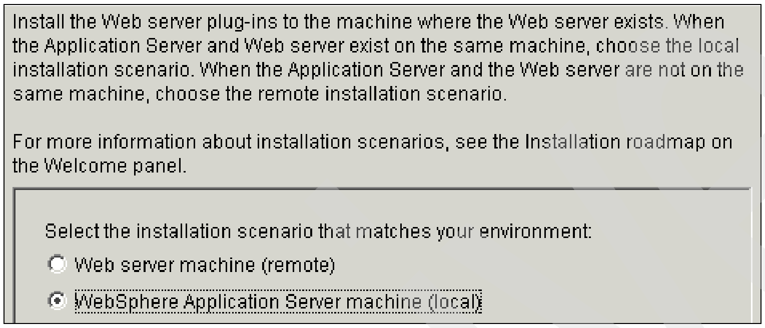
Installing the plug-ins also installs the required level of GSKit.

Figure 12-32 Plug-in installation initial window


Select the Web server to configure. All plug-in binaries are installed, but only selected Web servers are configured:

○ None
● IBM HTTP Server V6
○ Apache Web Server V2
○ Lotus Domino Web Server V6 or V6.5
○ Sun ONE Web Server 6.0 or Sun Java System Web Server V6.1
○ Microsoft Internet Information Services V6
○ Microsoft Internet Information Services V5

Figure 12-33 Plug-in selection

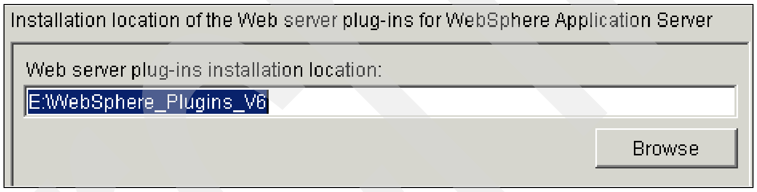

Install the Web server plug-ins to the machine where the Web server exists. When the Application Server and Web server exist on the same machine, choose the local installation scenario. When the Application Server and the Web server are not on the same machine, choose the remote installation scenario.

For more information about installation scenarios, see the Installation roadmap on the Welcome panel.

Select the installation scenario that matches your environment:
○ Web server machine (remote)
● WebSphere Application Server machine (local)

Figure 12-34 Plug-in selection remote and local

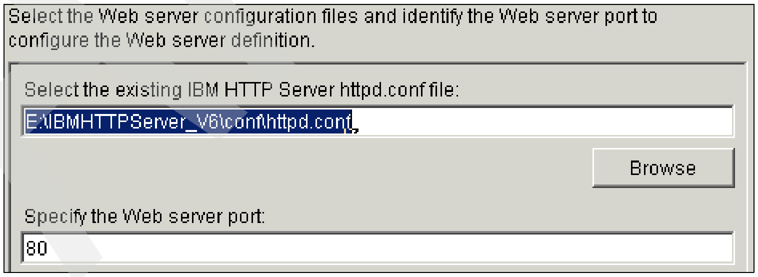
Figure 12-35 specifies a directory location for the plug-in binary files.


Installation location of the Web server plug-ins for WebSphere Application Server

Web server plug-ins installation location:
E:\WebSphere_Plugins_V6

Figure 12-35 Plug-in target directory

Figure 12-36 shows the location of the Web server configuration file, which is edited by the installer to point to the plug-in binary and the plug-in configuration file.

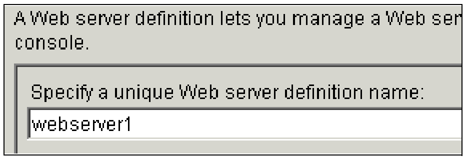

Select the Web server configuration files and identify the Web server port to configure the Web server definition.

Select the existing IBM HTTP Server httpd.conf file:
E:\IBMHTTPServer_V6\conf\httpd.conf

Specify the Web server port:
80

Figure 12-36 Plug-in Web server configuration file location

Figure 12-37 on page 204 shows the default name of the Web server instance under the application server. Choose a name that represents something meaningful to you.

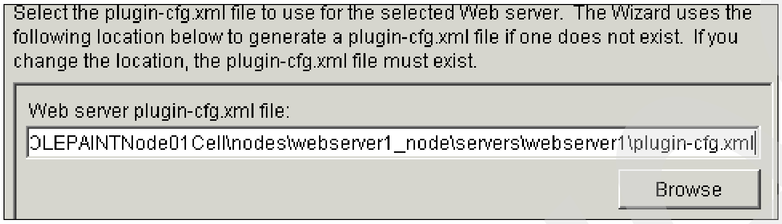


A Web server definition lets you manage a Web ser
console.

Specify a unique Web server definition name:
webserver1

Figure 12-37 Plug-in Web server definition

Figure 12-38 shows the default location of the plug-in configuration file that is contained in the application server configuration. This file location is placed into a line in the Web server configuration file.

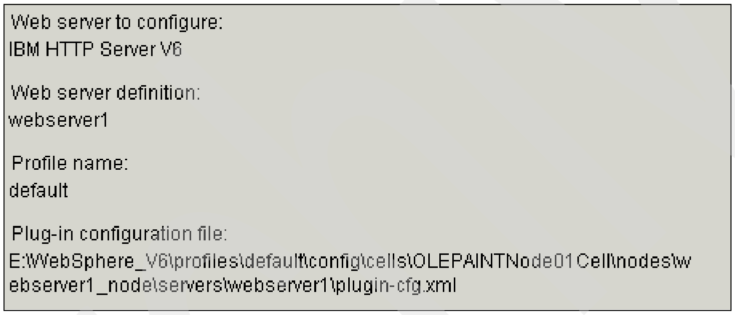


Select the plugin-cfg.xml file to use for the selected Web server. The Wizard uses the following location below to generate a plugin-cfg.xml file if one does not exist. If you change the location, the plugin-cfg.xml file must exist.

Web server plugin-cfg.xml file:
OLEPAINTNode01Cell\nodes\webserver1_node\servers\webserver1\plugin-cfg.xml

Figure 12-38 Plug-in configuration file

Figure 12-39 shows the final screen of the plug-in installation wizard.



Web server to configure:
IBM HTTP Server V6

Web server definition:
webserver1

Profile name:
default

Plug-in configuration file:
E:\WebSphere_V6\profiles\default\config\cells\OLEPAINTNode01Cell\nodes\w
ebserver1_node\servers\webserver1\plugin-cfg.xml

Figure 12-39 Plug-in confirmation

When the plug-in installation is complete, verify that the configurations in various files are correct. The last few lines of the Web server configuration file look like this:

```
file E:\IBMHTTPServer_V6\conf\httpd.conf
                lines deleted for brevity
LoadModule was_ap20_module "E:\WebSphere_Plugins_V6\bin\mod_was_ap20_http.dll"
WebSpherePluginConfig
"E:\WebSphere_V6\profiles\default\config\cells\OLEPAINTNode01Cell\nodes\webserver1_node\ser
vers\webserver1\plugin-cfg.xml"
```

This excerpt from the plugin-cfg.xml shows that the applications are mapped properly:

```
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/JMSSampleWeb/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/JMSSampleWebservlet/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/MyBankWeb/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/MyBankWebservlet/*"/>
```

Note that the plug-in configuration file was automatically generated by the plug-in installer. Also note that all the applications that are migrated by the migration utilities are mapped into the plug-in configuration file because the plug-in installer runs *after* the migration utilities. The order of these two steps is important. If the plug-ins are installed before running the migration utilities, you must map each application to webserver1 and generate the plug-in configuration file. See 8.3.4, “Plugin-Cfg generation” on page 115 for the steps to map applications to Web servers.

12.3.7 Confirming that the application is functioning correctly

Before testing the applications, start the Web server and application server. If you configured the servers under Windows services, start them with the Windows Services Control Panel. Otherwise, start the servers via the command line as shown in this example:

```
cd /d E:\IBMHTTPServer_V6\bin
E:\IBMHTTPServer_V6\bin>apache

cd /d E:\WebSphere_V6\profiles\default\bin
startServer server1
```

Verify that the applications are accessible by accessing the proper URL via both ports 9080 and 80.

12.4 Automatic migration: V4 AE Server Group to a V6 Network Deployment Cluster

This example shows how to use the command line migration utilities to migrate a WebSphere Application Server Version 4 Advanced Edition Server Group to a WebSphere Application Server Network Deployment Version 6 environment. The migration wizard can also be used if preferred. An example using the migration wizard is provided for migration of the WebSphere Application Server Network Deployment Server Cluster from V5 to V6. We used fix level 4.0.6 for this example, but any V4 fix level should work. The application used to illustrate the migration techniques is one of the sample applications that ships with WebSphere Application Server Version 4 Advanced Edition. The sample application includes two Web modules as well as two EJB modules. The EJB modules include a stateless session EJB for the “BeenThere” application and a CMP entity EJB for the “Hitcount” application. The CMP EJB bean uses a data source that points to a single table database, which is implemented in IBM DB2 Version 8.1.

Note: This example is based on the Windows operating system. Other operating systems will vary slightly in the script names (for example, backupConfig.sh instead of backupConfig.bat) as well as the OS commands for directory creation, and so forth. Details about the differences for each OS are documented in the WebSphere Application Server V6 Info Center.

12.4.1 Preparing your existing environment for migration

Even though the WebSphere Application Server Version 6 migration utilities make copies of the key files on the existing WebSphere system, it is recommended that you perform a file system backup on your system using a mechanism appropriate for your operating system. A backup provides you with a recovery point should problems occur during the migration process. In addition, since the configuration repository for WebSphere Application Server

Version 4 Advanced Edition is stored in a relational database, you should also perform a database backup using the tools provided by your database vendor.

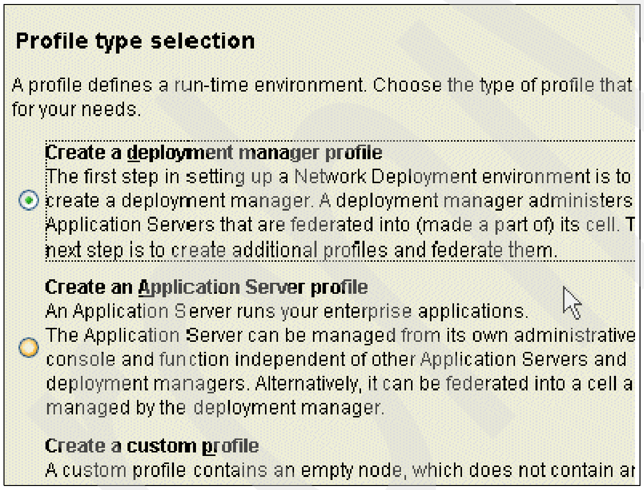
12.4.2 Preparing for a V6 installation

After installation of WebSphere Application Server Version 6, you need to create two types of profiles to which to migrate your WebSphere Application Server Version 4 Advanced Edition environment:

- A custom profile for each Version 6 node to which the V4 Server Group is migrated
- A deployment manager profile on the Version 6 node where the deployment manager is running

> **Note:** Each of the Version 6 nodes that you create custom profiles for must be a stand-alone node during the migration process. *Do not* federate the custom profiles into the Version 6 cell with the `addNode` command until after the migration process is complete.

Figure 12-40 shows that the deployment manager profile type is selected. At the bottom of the window, you also see the custom profile type. A more complete discussion of profile creation is in “Creating profiles” on page 106.



Profile type selection

A profile defines a run-time environment. Choose the type of profile that for your needs.

● **Create a deployment manager profile**
The first step in setting up a Network Deployment environment is to create a deployment manager. A deployment manager administers Application Servers that are federated into (made a part of) its cell. T next step is to create additional profiles and federate them.

○ **Create an Application Server profile**
An Application Server runs your enterprise applications. The Application Server can be managed from its own administrative console and function independent of other Application Servers and deployment managers. Alternatively, it can be federated into a cell a managed by the deployment manager.

Create a custom profile
A custom profile contains an empty node, which does not contain ar

Figure 12-40 Version 6 profile type selection

12.4.3 Reviewing the V4 configuration

The current WebSphere Application Server Version 4 Advanced Edition system configuration is shown in Figure 12-41 on page 207. There is a single node in the cell, “sonoma” and a “SonomaServerGroup” which is comprised of two servers, “SonomaServer1” and SonomaServer2.” In most cases you will have more than one node, but you can simply repeat the steps indicated here for each node in order to perform your migration.

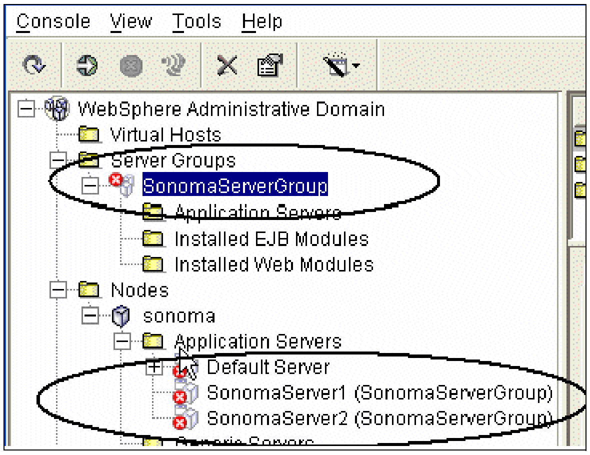
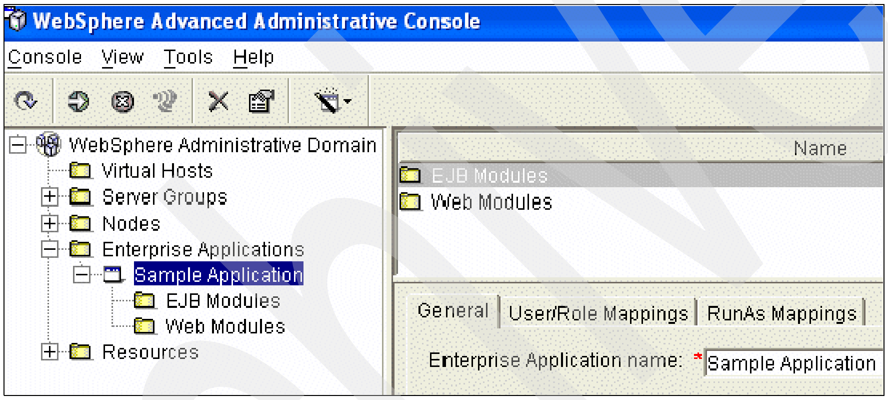


Figure 12-41 Version 4 console showing server groups

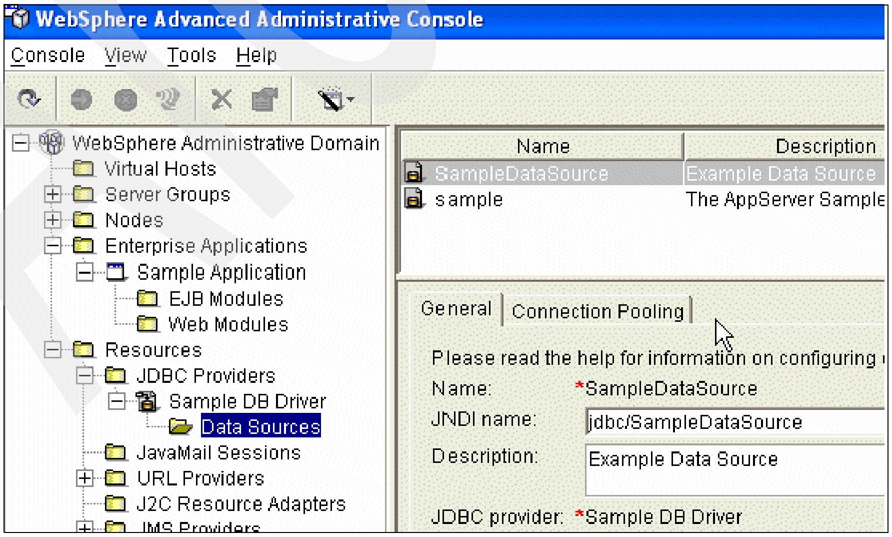
Figure 12-42 shows the application that is currently installed.



Enterprise Application name: *Sample Application

Figure 12-42 Version 4 console showing applications

Figure 12-43 shows the JDBC driver and associated data sources.



Name	Description
SampleDataSource	Example Data Source
sample	The AppServer Sample

Please read the help for information on configuring

Name: *SampleDataSource

JNDI name: jdbc/SampleDataSource

Description: Example Data Source

JDBC provider: *Sample DB Driver

Figure 12-43 Version 4 console showing JDBC provider

12.4.4 Migrating nodes using automatic migration utilities

The first step in performing the migration is to execute the **WASPreUpgrade** command. This command and the results are shown in Example 12-1. The required command line arguments for this command are:

```
WASPreUpgrade <backup directory name> <existing WAS V4 install directory>
```

In this example, we have chosen to save the results in the directory C:\WAS40\backups and the existing V4 install directory is C:\WAS40\AppServer.

Example 12-1 Executing WASPreUpgrade

```
C:\WAS60\AppServer\profiles\SonomaCustom01\bin>WASPreUpgrade c:\WAS40\backups
c:\WAS40\AppServer sonoma
IBM WebSphere Application Server, Release 6.0
Product Upgrade PreUpgrade tool, Version 1.0
Copyright IBM Corp., 1997-2004

MIGR0300I: The migration function is starting to save the existing WebSphere Application
Server environment.
MIGR0241I: Output of Get character encoding settings.
MIGR0301I: The existing configuration is being exported.
MIGR0241I: Output of XMLConfig Export.
[11/30/04 21:00:58:769 PST] 5c405690 VirtualHostCo A XMLC0052I: Exporting VirtualHost :
default_host
[11/30/04 21:00:58:849 PST] 5c405690 JMSProviderCo A XMLC0052I: Exporting JMSProvider : IBM
MQSeries
[11/30/04 21:00:59:000 PST] 5c405690 URLProviderCo A XMLC0052I: Exporting URLProvider :
Default URL Provider
[11/30/04 21:00:59:100 PST] 5c405690 JDBCDriverCon A XMLC0052I: Exporting JDBCDriver :
Sample DB Driver
[11/30/04 21:00:59:210 PST] 5c405690 DataSourceCon A XMLC0052I: Exporting DataSource :
SampleDataSource
[11/30/04 21:00:59:230 PST] 5c405690 DataSourceCon A XMLC0052I: Exporting DataSource :
sample
[11/30/04 21:00:59:280 PST] 5c405690 NodeConfig    A XMLC0052I: Exporting Node : sonoma
[11/30/04 21:00:59:661 PST] 5c405690 ApplicationSe A XMLC0052I: Exporting ApplicationServer
: Default Server
[11/30/04 21:01:00:262 PST] 5c405690 SecurityConfi A XMLC0052I: Exporting SecurityConfig :
{1}
[11/30/04 21:01:01:033 PST] 5c405690 ServerGroupCo A XMLC0052I: Exporting ServerGroup :
SonomaServerGroup
[11/30/04 21:01:01:984 PST] 5c405690 EnterpriseApp A XMLC0052I: Exporting EnterpriseApp :
Sample Application
MIGR0302I: The existing files are being saved.
MIGR0303I: The existing WebSphere Application Server environment is saved.
MIGR0420I: The first step of migration completed successfully.

C:\WAS60\AppServer\profiles\SonomaCustom01\bin>
```

After successful completion of **WASPreUpgade**, we are ready to run **WASPostUpgrade**, which imports the configuration for our WebSphere Application Server Version 4 Advanced Edition node to the WebSphere Application Server Version 6 node custom profile that we have created. Example 12-2 on page 209 shows the execution of the **WASPostUpgrade** command. Note that we have run this command from the bin directory for the custom profile that corresponds to the custom profile that we created earlier. Also note that if we had multiple nodes, we would perform the **WASPostUpgrade** command once on each of the Version 6 nodes.

Example 12-2 Executing WASPostUpgrade

```
C:\WAS60\AppServer\profiles\SonomaCustom01\bin>WASPostUpgrade C:\WAS40\backups
IBM WebSphere Application Server, Release 6.0
Product Upgrade PostUpgrade tool, Version 1.0
Copyright IBM Corp., 1997-2004

MIGR0304I: The previous WebSphere environment is being restored.
MIGR0367I: Backing up the current WebSphere Application Server environment.
MIGR0241I: Output of backupConfig.
ADMU0116I: Tool information is being logged in file
           C:\WAS60\AppServer\profiles\SonomaCustom01\logs\backupConfig.log
ADMU0128I: Starting tool with the SonomaCustom01 profile
ADMU5001I: Backing up config directory
           C:\WAS60\AppServer/profiles/SonomaCustom01\config to file
           C:\WAS60\AppServer\profiles\SonomaCustom01\temp\MigrationBackup.Thu-D
ec-02-18.47.42-2004.zip
ADMU5002I: 174 files successfully backed up
MIGR0237I: The migration function read configuration file C:\WAS60\AppServer\pro
files\SonomaCustom01\config\cells\localhostNode01Cell\security.xml successfully.

                   output edited for brevity

MIGR0408I: The configuration is saved successfully.
MIGR0341W: The application Sample_Application was not installed. The application is part of
a model or server group configuration.
MIGR0241I: Output of GenPluginCfg.

IBM WebSphere Application Server, Release 6.0
WebSphere Plugin Configuration Generator
Copyright IBM Corp., 1997-2004

PLGC0013I: The plug-in is generating a server plug-in configuration file for all of servers
in the cell localhostNode01Cell.

PLGC0005I: Plug-in configuration file =
C:\WAS60\AppServer/profiles/SonomaCustom01\config\cells\plugin-cfg.xml
MIGR0307I: The restoration of the previous WebSphere Application Server environment is
complete.
MIGR0271W: The migration completed with warnings.
```

The output runs for many pages, so the example in this book has been edited to exclude many of the repetitive lines. As you can see, the migration has completed with warnings. It is important that you examine the warnings to determine if you need to take any corrective action before proceeding. In this example, we see the following warnings:

- The application server initial and maximum heap sizes were not imported and may need to be adjusted.
- Port conflicts might occur if there are pre-existing servers on our V6 node. In this case, there aren't any we can safely ignore.
- It may be necessary to modify the classpath entries for the URL Provider, JMS Provider, and JDBC Driver if the locations differ for these entries between the V4.x node and the V6.0 node.
- Certain security settings and properties are not migrated. You should configure the LDAP settings before enabling security. Also, various KeyRing files are not migrated. You should either manually migrate the KeyRing settings, or configure SSL to use new KeyRings. Since periodic KeyRing maintenance is a good idea, you might want to take this opportunity to do this, but only if it does not impact your current production machines.

In our example, we are going to wait before making any changes since we will need to run **WASPostUpgrade** on the deployment manager profile. Some of the warnings refer to cell-wide configuration parameters, such as LDAP server settings for Global Security. It is best to note the warnings here and make the changes all at once.

Once you have examined the output from the **WASPostUpgrade** command, you are ready to run **addNode**. This is depicted in Example 12-3, where the deployment manager is running on the same node, sonoma, that the application servers are on.

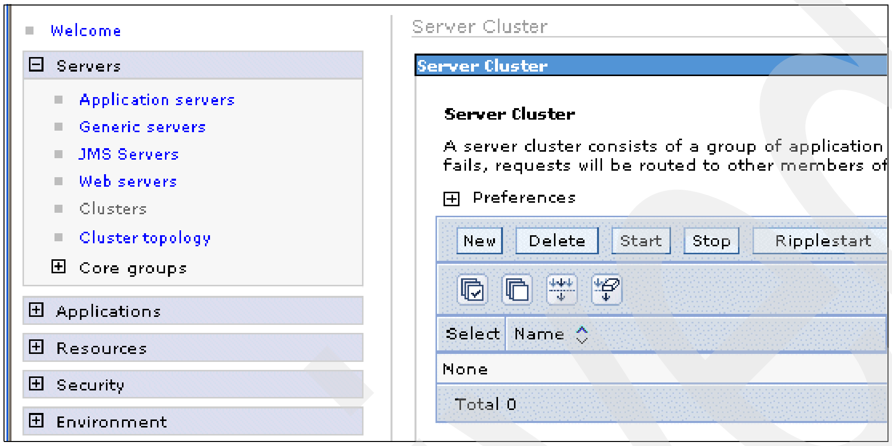
> **Note:** Recall that you need to run **WASPostUpgrade** on *each* of the nodes that you are migrating, and you should do so *before* running **addNode**.

Example 12-3 Executing addNode

```
C:\WAS60\AppServer\profiles\SonomaCustom01\bin>addnode sonoma
ADMU0116I: Tool information is being logged in file
           C:\WAS60\AppServer\profiles\SonomaCustom01\logs\addNode.log
ADMU0128I: Starting tool with the SonomaCustom01 profile
ADMU0001I: Begin federation of node sonoma with Deployment Manager at
           sonoma:8879.
ADMU0009I: Successfully connected to Deployment Manager Server: sonoma:8879
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU0506I: Server name: SonomaServer1
ADMU0506I: Server name: SonomaServer2
ADMU2010I: Stopping all server processes for node sonoma
ADMU0512I: Server server1 cannot be reached. It appears to be stopped.
ADMU0512I: Server SonomaServer1 cannot be reached. It appears to be stopped.
ADMU0512I: Server SonomaServer2 cannot be reached. It appears to be stopped.
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating Node Agent configuration for node: sonoma
ADMU0014I: Adding node sonoma configuration to cell: sonomaNetwork
ADMU0016I: Synchronizing configuration between node and cell.
oldVarTable: {}
newVarTable: {}
diffTable: {}
ADMU0018I: Launching Node Agent process for node: sonoma
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0030I: Node Agent initialization completed successfully. Process id is:
           3004
ADMU0300I: Congratulations! Your node sonoma has been successfully incorporated
           into the sonomaNetwork cell.
ADMU0306I: Be aware:
ADMU0302I: Any cell-level documents from the standalone localhostNode01Cell
           configuration have not been migrated to the new cell.
ADMU0307I: You might want to:
ADMU0303I: Update the configuration on the sonomaNetwork Deployment Manager
           with values from the old cell-level documents.
ADMU0306I: Be aware:
ADMU0304I: Because -includeapps was not specified, applications installed on
           the standalone node were not installed on the new cell.
ADMU0307I: You might want to:
ADMU0305I: Install applications onto the sonomaNetwork cell using wsadmin
           $AdminApp or the Administrative Console.
ADMU0003I: Node sonoma has been successfully federated.
```

12.4.5 Migrating the deployment manager using automatic migration utilities

While the node has been successfully migrated and federated into the cell, we are still not finished. We still need to run **WASPostUpgrade** on the deployment manager profile so that various cell-wide definitions such as the Server Cluster and Resources can be migrated. If you were to go to the administrative console at this point, without running **WASPostUpgrade** on the deployment manager, these resources would not be configured. An example of this is shown by the missing Server Cluster in Figure 12-44.



A server cluster consists of a group of application
fails, requests will be routed to other members of

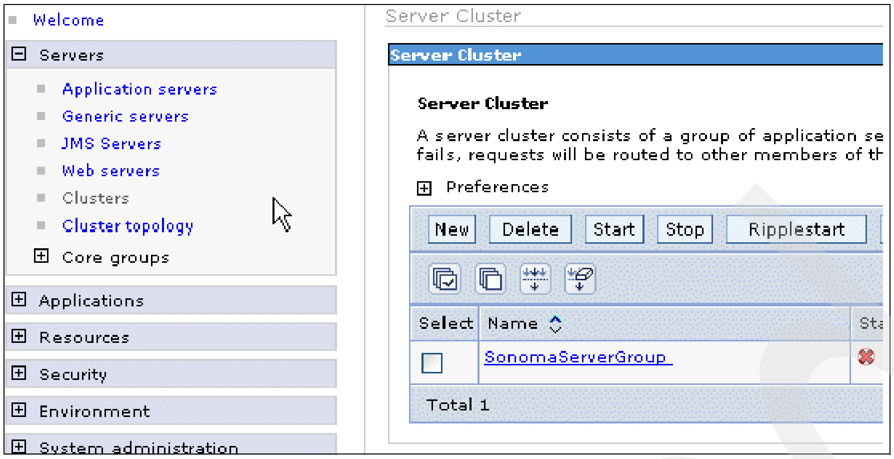
Figure 12-44 Version 6 console showing clusters

After stopping the deployment manager with the **stopManager** command, we are ready to run **WASPostUpgrade** on the deployment manager profile that we created. We execute this command from the bin directory under the deployment manager profile as shown in Example 12-4.

Example 12-4 Executing WASPostUpgrade on deployment manager

```
C:\WAS60\AppServer\profiles\SonomaDmgr01\bin>WASPostUpgrade C:\WAS40\backups
IBM WebSphere Application Server, Release 6.0
Product Upgrade PostUpgrade tool, Version 1.0
Copyright IBM Corp., 1997-2004

MIGR0304I: The previous WebSphere environment is being restored.
MIGR0367I: Backing up the current WebSphere Application Server environment.
MIGR0241I: Output of backupConfig.
ADMU0116I: Tool information is being logged in file
           C:\WAS60\AppServer\profiles\SonomaDmgr01\logs\backupConfig.log

                   edited for brevity

MIGR0408I: The configuration is saved successfully.
MIGR0307I: The restoration of the previous WebSphere Application Server environment is
complete.
MIGR0271W: The migration completed with warnings.
```

At this point, make a note of the various warnings so that you can make any changes or updates required for the application server and deployment manager all at once, as mentioned previously, after starting the deployment manager using the **startManager** command.



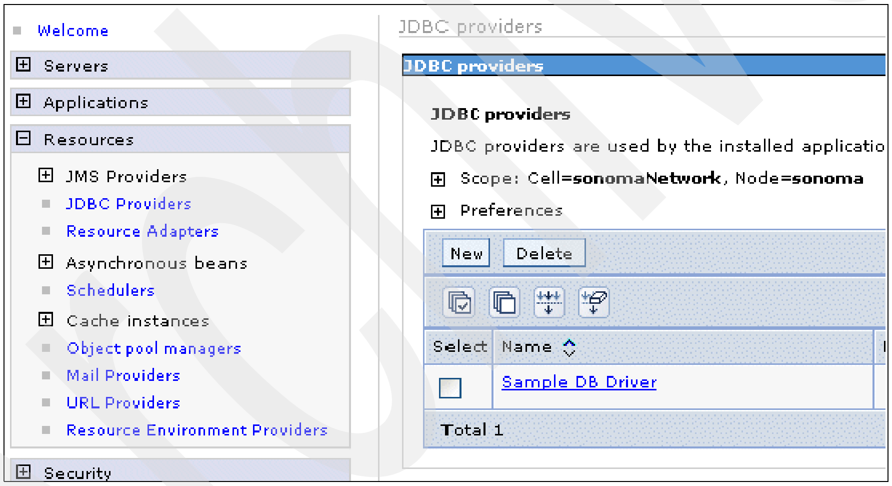
Server Cluster

A server cluster consists of a group of application se
fails, requests will be routed to other members of th

Select	Name	Sta
□	SonomaServerGroup	✖
Total 1		

Figure 12-45 Server clusters after deployment manager migration

After starting the deployment manager, you should navigate to various artifacts mentioned previously to make sure that they are now present. These are depicted in Figure 12-46, Figure 12-47 on page 213, and Figure 12-48 on page 214 for the ServerGroup that is now a Server Cluster, the JDBC Driver and its associated V4 Data Sources. Verify that the classpath settings are correct for the JDBC Driver, and update the LDAP server settings for Global Security as well as the KeyRing settings.



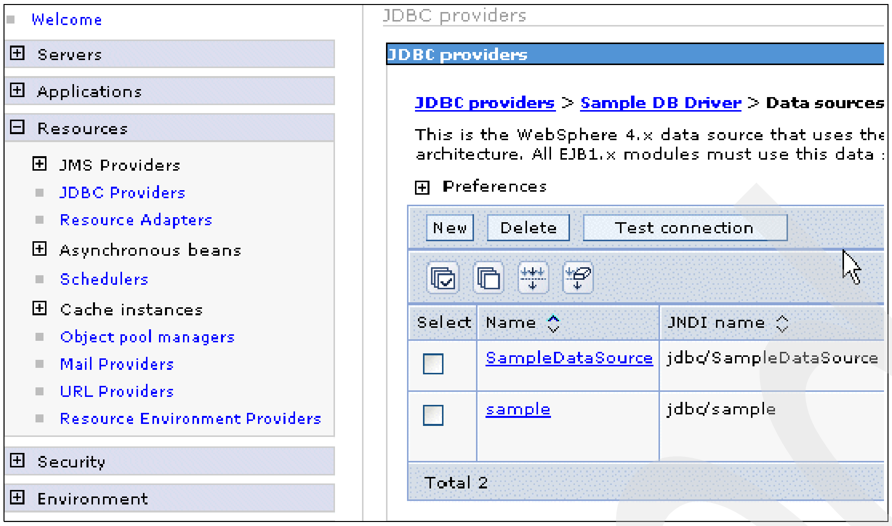
JDBC providers

JDBC providers are used by the installed applicatio

Scope: Cell=**sonomaNetwork**, Node=**sonoma**

Select	Name
□	Sample DB Driver
Total 1	

Figure 12-46 Version 6 console showing JDBC provider



This is the WebSphere 4.x data source that uses the
architecture. All EJB1.x modules must use this data

Select	Name	JNDI name
□	SampleDataSource	jdbc/SampleDataSource
□	sample	jdbc/sample
Total 2		

Figure 12-47 Version 6 console showing data source

Once you have made updates to the resources or application server definitions such as heap size, you are ready to install the Enterprise Application to the Server Cluster. Note that migration from V4 to V6 does not migrate the application installation, but it does place the EAR file for any applications in this directory:

```
profiles/nodeprofile/installableApps
```

In our example, this is:

```
C:\WAS60\AppServer\profiles\SonomaCustom01\installableApps
```

You then install any applications in this directory via the administrative console. This process is very similar to installation on V4 or V5. Once you have finished installing the application, be sure to save your changes and synchronize your changes with all the nodes in the cluster.

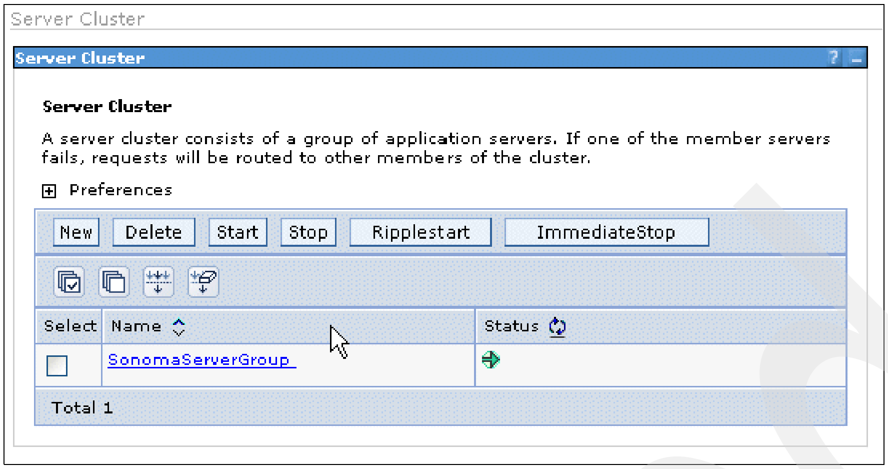
12.4.6 Migrating Web Server and plug-ins

You are now ready to regenerate the HTTP server plug-in file and distribute it to the HTTP servers. Depending on whether or not you have chosen to create managed nodes for your HTTP servers, the mechanism for distributing the updated plug-in configuration file varies. Your options are:

- Create a custom profile so that you have a node agent to manage your HTTP server, plug-in file configuration, and plug-in file distribution.
- Install IBM HTTP Server V6.0, which provides HTTP server and plug-in maintenance without a node agent.
- Manually manage HTTP server and plug-in maintenance.

See 12.3.6, “Migrating Web server and WebSphere plug-ins” on page 201 for more details on migrating the Web server.

Once you have taken care of plug-in maintenance, you are ready to start your Server Cluster. Once the Server Cluster is started, it looks as shown in Figure 12-48 on page 214.



Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Select	Name	Status
□	SonomaServerGroup	➜
Total 1		

Figure 12-48 Version 6 console showing starting the cluster

Assuming your application test is successful, you have completed your V4.x to V6.0 migration.

12.5 Automatic migration: V5 Network Deployment Cluster to V6 Network Deployment Cluster

This example shows how to use the migration wizard to migrate a WebSphere Application Server Network Deployment Version 5 Server Cluster to a V6 environment. The command line utilities can also be used if preferred. We used fix level 5.1.1 for this example, but any Version 5 level can be migrated in the same manner. The application we used is one of the sample applications that ships with WebSphere Application Server Version 5; the default application includes a Web module as well as an EJB module. The EJB module includes a CMP entity EJB for the hitcount application. The CMP EJB bean uses a data source that points to a single table database, which is implemented in IBM DB2 Version 8.1.

While we are going to migrate both the application server (Server Cluster) nodes and the deployment manager node in this example, a Version 6 deployment manager can manage a cell comprised of both Version 5 and Version 6 nodes. We demonstrate the V6 deployment manager managing a Version 5 node. This capability allows you to incrementally upgrade your runtime and applications, minimizing or possibly eliminating any application service interruption.

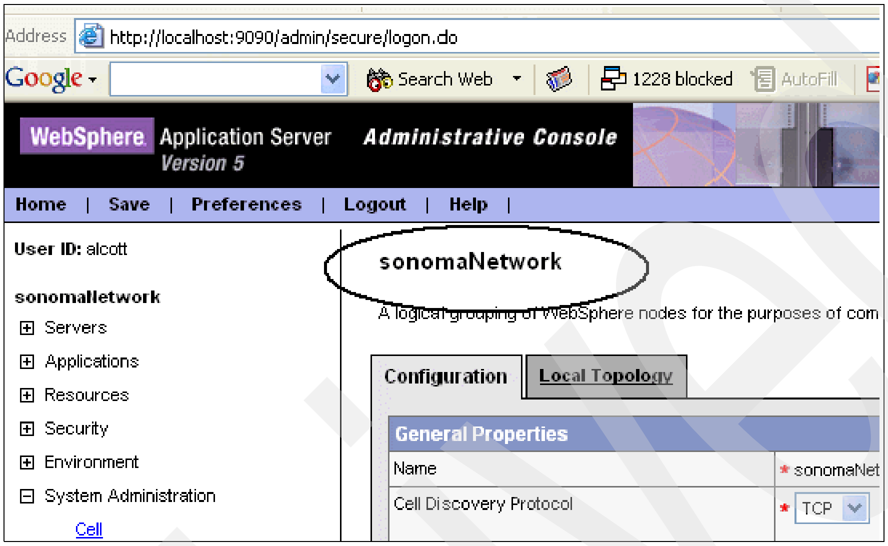
Note: Within a given cell, the deployment manager must be at the latest release and highest version of any nodes in the cell in order for it to manage the nodes in the cell.

12.5.1 Preparing your existing environment for migration

Even though the automatic migration utilities make copies of the key files on the existing WebSphere system, it is recommended that you perform a file system backup on your system, using a mechanism appropriate for your operating system, before starting any migration activities; this provides you with a recovery point should problems occur during the migration process.

12.5.2 Starting the configuration

You will need to take note of the current Version 5 node and cell names before you proceed with Version 6 profile creation. The names for the cell and the nodes should match when moving from Version 5 to Version 6. This information is available from a variety of sources; perhaps the easiest way is to look at the cell name in the administration console shown in Figure 12-49.



User ID: alcott

sonomaNetwork

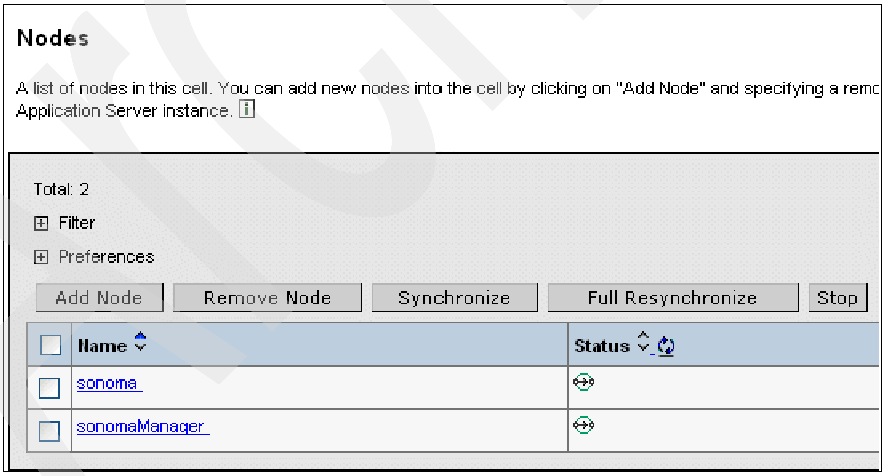
A logical grouping of WebSphere nodes for the purposes of com

Name * sonomaNet

Cell Discovery Protocol * TCP

Figure 12-49 Version 5 console cell name

The node names are shown in Figure 12-50, where "sonoma" is the application server node name and "sonomaManager" is the deployment manager node name. In cases where multiple nodes exist, record the names for the V5 application server nodes and then create a V6 custom or stand-alone application server profile for each of those nodes.



A list of nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remc Application Server instance.

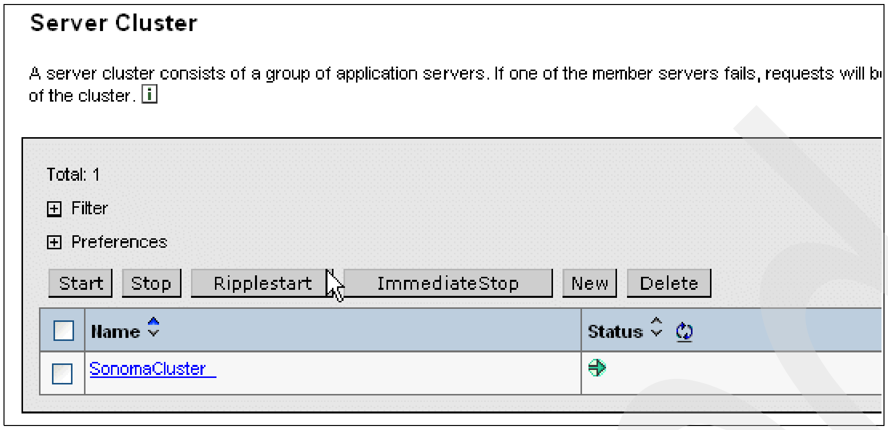
Total: 2

☐	Name	Status
☐	sonoma	⇔
☐	sonomaManager	⇔

Figure 12-50 Version 5 console nodes in cell

Note: You *must* use the same node and cell names when migrating from Version 5 to Version 6.

The current V5.x cell has a server cluster named “SonomaCluster” as depicted in Figure 12-51.



Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will b of the cluster. ⓘ

Total: 1

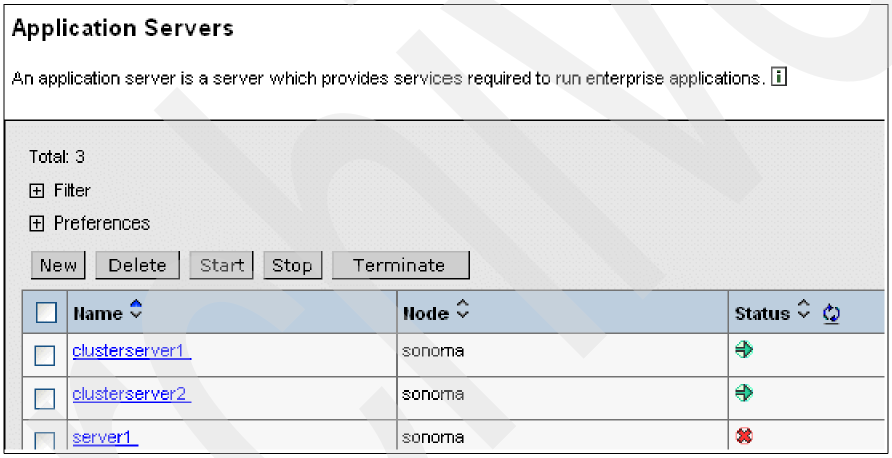
⊞ Filter

⊞ Preferences

□	Name	Status
□	SonomaCluster	➔

Figure 12-51 Version 5 console clusters in cell

The Server Cluster is comprised of two application servers, clusterserver1 and clusterserver2, as shown in Figure 12-52.



Application Servers

An application server is a server which provides services required to run enterprise applications. ⓘ

Total: 3

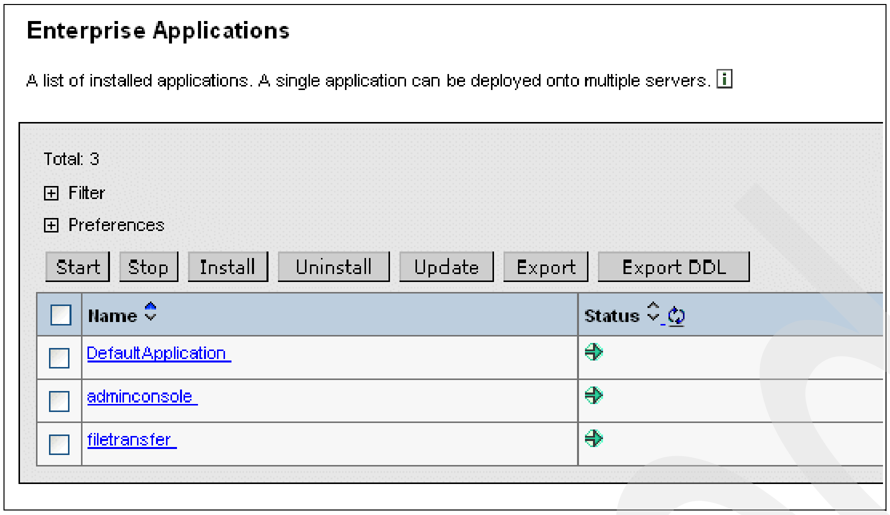
⊞ Filter

⊞ Preferences

□	Name	Node	Status
□	clusterserver1	sonoma	➔
□	clusterserver2	sonoma	➔
□	server1	sonoma	✖

Figure 12-52 Version 5 console servers in cluster

The Enterprise Applications installed are shown in Figure 12-53 on page 217. The application running in the Server Cluster is the DefaultApplication. You will note that the Server Cluster and application are running. You can migrate a V5.x node without stopping it. The migration tools can collect all the configuration data while the node is either running or stopped, but you must stop the V5.x node before you can start the V6.0 node that you are migrating to.



Enterprise Applications

A list of installed applications. A single application can be deployed onto multiple servers.

Total: 3

⊞ Filter

⊞ Preferences

□	Name	Status
□	DefaultApplication	➜
□	adminconsole	➜
□	filetransfer	➜

Figure 12-53 Version 5 console applications

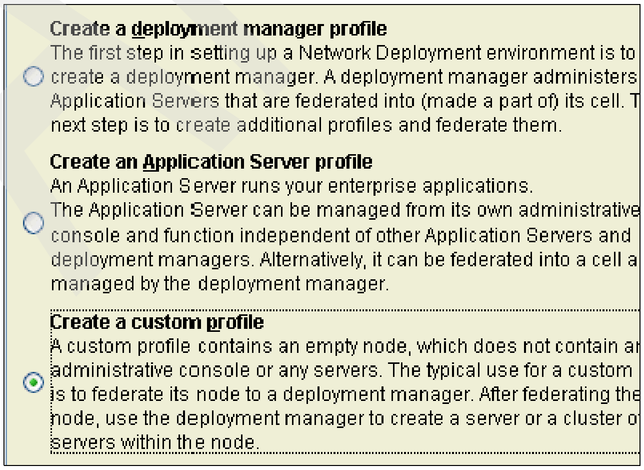
12.5.3 V6 installation preparation

After the installation of WebSphere Application Server Network Deployment, you must create two types of profiles:

- Create either a custom profile or a stand-alone server profile for each node where the Server Cluster is installed. We use a custom profile in this example.
- Create a deployment manager profile on the node where the deployment manager is running.

> **Note:** Each of the target Version 6 nodes that you create custom profiles for must be stand-alone nodes during the migration process. Do not federate the custom profiles into the Version 6 cell with the **addNode** command prior to starting the migration process. The migration utilities perform this task during the migration process.

The specification of a custom profile is shown in Figure 12-54. Node and host name properties are depicted in Figure 12-55 on page 218. Recall that the node name information must match the current V5 node name information, as discussed previously.

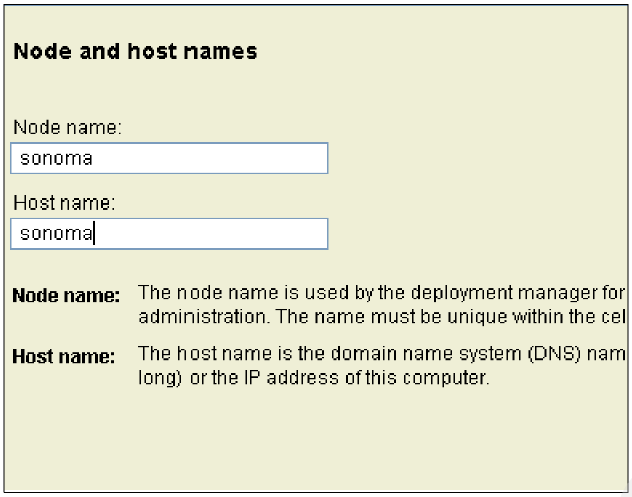


○ **Create a deployment manager profile**
The first step in setting up a Network Deployment environment is to create a deployment manager. A deployment manager administers Application Servers that are federated into (made a part of) its cell. T next step is to create additional profiles and federate them.

○ **Create an Application Server profile**
An Application Server runs your enterprise applications. The Application Server can be managed from its own administrative console and function independent of other Application Servers and deployment managers. Alternatively, it can be federated into a cell a managed by the deployment manager.

● **Create a custom profile**
A custom profile contains an empty node, which does not contain ar administrative console or any servers. The typical use for a custom is to federate its node to a deployment manager. After federating the node, use the deployment manager to create a server or a cluster o servers within the node.

Figure 12-54 Selecting custom profile



Node and host names

Node name:

sonoma

Host name:

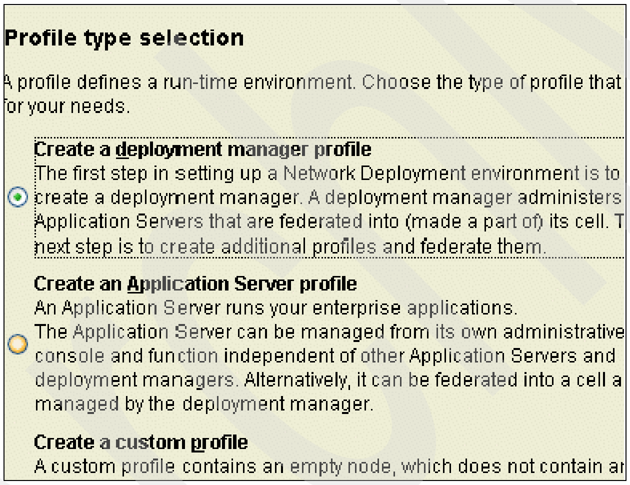
sonoma

Node name: The node name is used by the deployment manager for administration. The name must be unique within the cel

Host name: The host name is the domain name system (DNS) nam long) or the IP address of this computer.

Figure 12-55 Profile node and host names

The selection of a deployment manager profile in the profile creation wizard is shown in Figure 12-56. Specification of the node, host, and cell names is shown in Figure 12-57 on page 219. Again as noted previously, the node and cell names in V6.0 must match the information from the V5.x configuration.



Profile type selection

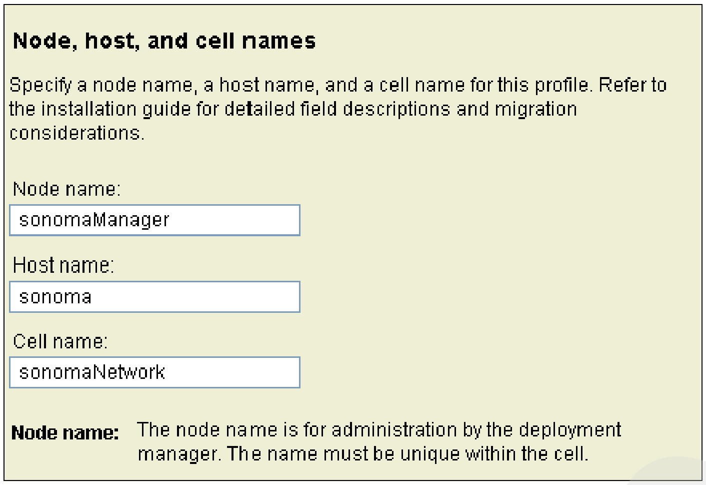
A profile defines a run-time environment. Choose the type of profile that for your needs.

● **Create a deployment manager profile**
The first step in setting up a Network Deployment environment is to create a deployment manager. A deployment manager administers Application Servers that are federated into (made a part of) its cell. T next step is to create additional profiles and federate them.

○ **Create an Application Server profile**
An Application Server runs your enterprise applications. The Application Server can be managed from its own administrative console and function independent of other Application Servers and deployment managers. Alternatively, it can be federated into a cell a managed by the deployment manager.

Create a custom profile
A custom profile contains an empty node, which does not contain ar

Figure 12-56 Selecting deployment manager profile



Node, host, and cell names

Specify a node name, a host name, and a cell name for this profile. Refer to the installation guide for detailed field descriptions and migration considerations.

Node name: sonomaManager

Host name: sonoma

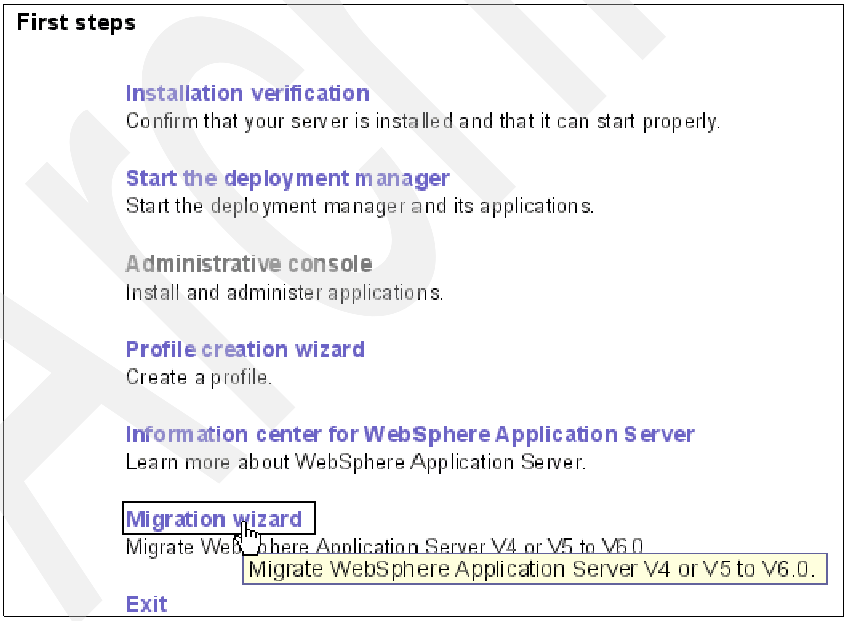
Cell name: sonomaNetwork

Node name: The node name is for administration by the deployment manager. The name must be unique within the cell.

Figure 12-57 Profile node and host names for deployment manager

12.5.4 Deployment manager migration using the migration wizard

The V6 migration wizard is a graphical alternative to the command line **`WASPreUpgrade`** and **`WASPostUpgrade`** commands, and is launched from the firststeps directory under the profile that you are migrating to. Since the first step in migrating the cell is to migrate the V5.x deployment manager to a V6.0 deployment manager, locate the firststeps directory under your deployment manager profile. In our case, this is C:\WAS60\AppServer\profiles\SonomaDmgr01\firststeps. From there, open a command window and type in **`firststeps`**. This brings up the First steps client, as shown in Figure 12-58.



First steps

Installation verification
Confirm that your server is installed and that it can start properly.

Start the deployment manager
Start the deployment manager and its applications.

Administrative console
Install and administer applications.

Profile creation wizard
Create a profile.

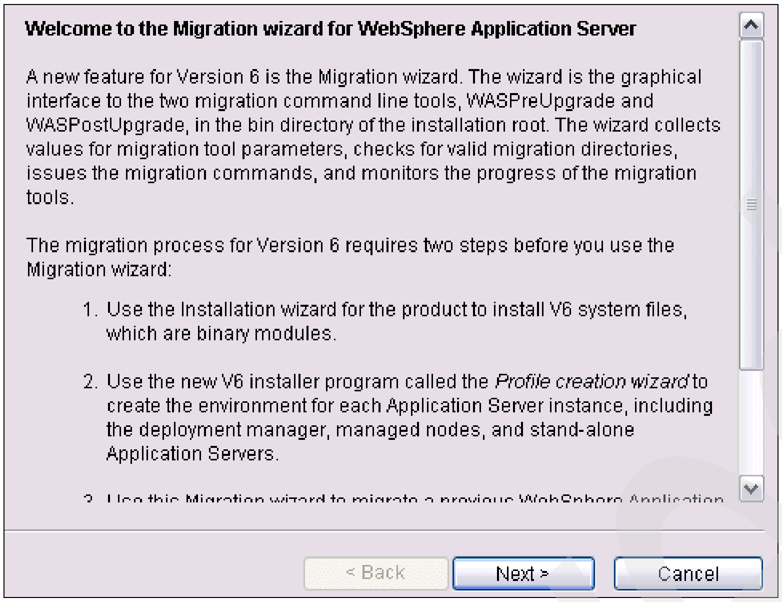
Information center for WebSphere Application Server
Learn more about WebSphere Application Server.

Migration wizard
Migrate WebSphere Application Server V4 or V5 to V6.0
Migrate WebSphere Application Server V4 or V5 to V6.0.

Exit

Figure 12-58 First steps

Select the migration wizard; this launches the migration wizard client as shown in Figure 12-59 on page 220. Select **Next** to continue with the migration wizard dialog.



Welcome to the Migration wizard for WebSphere Application Server

A new feature for Version 6 is the Migration wizard. The wizard is the graphical interface to the two migration command line tools, WASPreUpgrade and WASPostUpgrade, in the bin directory of the installation root. The wizard collects values for migration tool parameters, checks for valid migration directories, issues the migration commands, and monitors the progress of the migration tools.

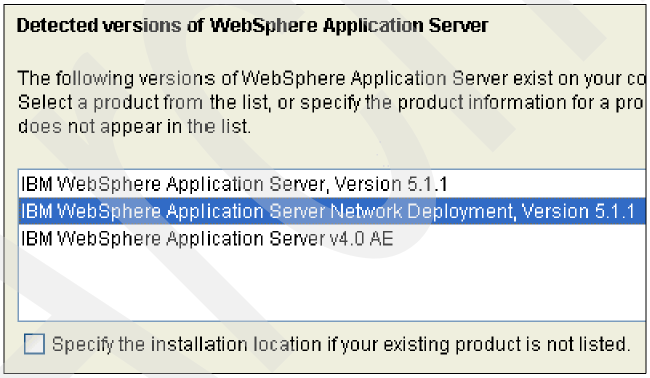
The migration process for Version 6 requires two steps before you use the Migration wizard:

1. Use the Installation wizard for the product to install V6 system files, which are binary modules.
2. Use the new V6 installer program called the *Profile creation wizard* to create the environment for each Application Server instance, including the deployment manager, managed nodes, and stand-alone Application Servers.
3. Use this Migration wizard to migrate a previous WebSphere Application

Figure 12-59 Starting migration wizard

This brings up the version selection dialog as shown in Figure 12-60. In this case, since you are migrating the deployment manager, you select the **Network Deployment** installation on the machine.

> **Note:** Before proceeding any further, be sure to stop the Version 5 deployment manager with the **`stopManager`** command if you have not already done so. Failure to do so results in a migration failure.



Detected versions of WebSphere Application Server

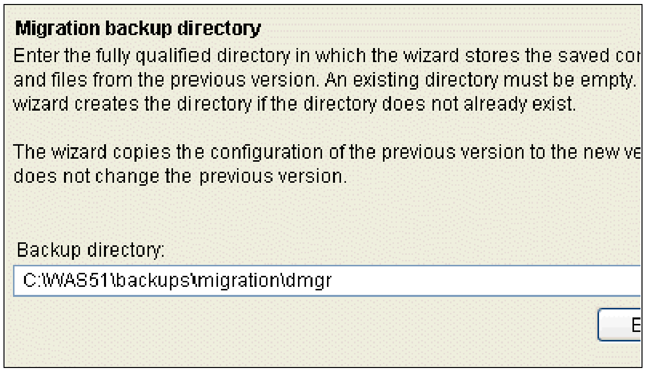
The following versions of WebSphere Application Server exist on your co
Select a product from the list, or specify the product information for a pro
does not appear in the list.

IBM WebSphere Application Server, Version 5.1.1
IBM WebSphere Application Server Network Deployment, Version 5.1.1
IBM WebSphere Application Server v4.0 AE

☐ Specify the installation location if your existing product is not listed.

Figure 12-60 Migration wizard existing versions

Specify the directory where the backup of the current V5.x configuration is saved, as shown in Figure 12-61 on page 221.



Migration backup directory

Enter the fully qualified directory in which the wizard stores the saved co
and files from the previous version. An existing directory must be empty.
wizard creates the directory if the directory does not already exist.

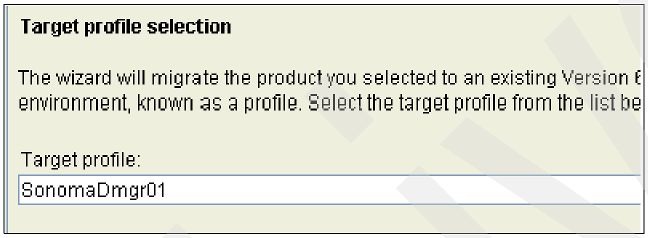
The wizard copies the configuration of the previous version to the new ve
does not change the previous version.

Backup directory:

C:\WAS51\backups\migration\dmgr

Figure 12-61 Migration wizard backup directory

The last input before the migration occurs is the specification of the target profile, as shown in Figure 12-62.



Target profile selection

The wizard will migrate the product you selected to an existing Version 6
environment, known as a profile. Select the target profile from the list be

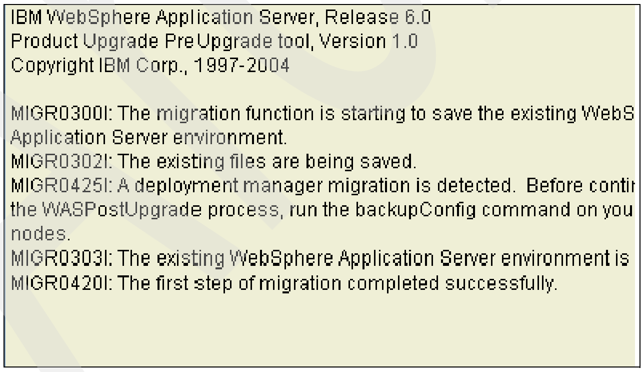
Target profile:

SonomaDmgr01

Figure 12-62 Migration wizard target profile

Note: Be sure to start the migration wizard from the startup directory under the profile you are migrating to; this simplifies input of the information related to the configuration you are migrating to.

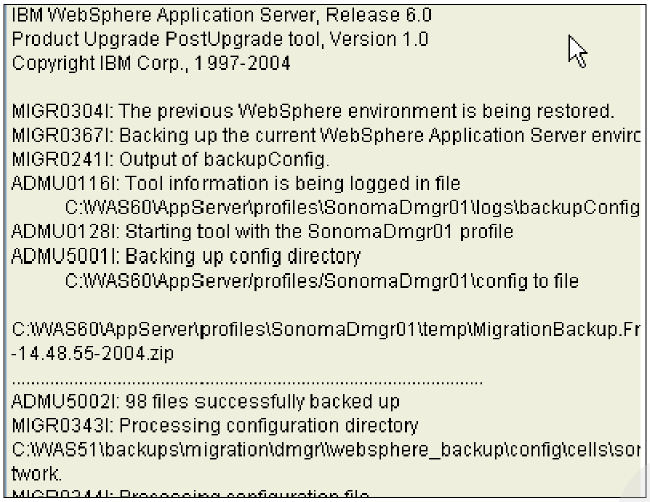
Start the migration wizard, which invokes the **WASPreUpgrade** command as shown in Figure 12-63.



```
IBM WebSphere Application Server, Release 6.0
Product Upgrade PreUpgrade tool, Version 1.0
Copyright IBM Corp., 1997-2004

MIGR0300I: The migration function is starting to save the existing WebS
Application Server environment.
MIGR0302I: The existing files are being saved.
MIGR0425I: A deployment manager migration is detected.  Before contir
the WASPostUpgrade process, run the backupConfig command on you
nodes.
MIGR0303I: The existing WebSphere Application Server environment is
MIGR0420I: The first step of migration completed successfully.
```

Figure 12-63 Migration wizard completion of WASPreUpgrade

Selecting **Next** invokes the **WASPostUpgrade** command, as shown in Figure 12-64 on page 222.



```
IBM WebSphere Application Server, Release 6.0
Product Upgrade PostUpgrade tool, Version 1.0
Copyright IBM Corp., 1997-2004

MIGR0304I: The previous WebSphere environment is being restored.
MIGR0367I: Backing up the current WebSphere Application Server enviro
MIGR0241I: Output of backupConfig.
ADMU0116I: Tool information is being logged in file
        C:\WAS60\AppServer\profiles\SonomaDmgr01\logs\backupConfig
ADMU0128I: Starting tool with the SonomaDmgr01 profile
ADMU5001I: Backing up config directory
        C:\WAS60\AppServer/profiles/SonomaDmgr01\config to file

C:\WAS60\AppServer\profiles\SonomaDmgr01\temp\MigrationBackup.Fr
-14.48.55-2004.zip
...............................................................................
ADMU5002I: 98 files successfully backed up
MIGR0343I: Processing configuration directory
C:\WAS51\backups\migration\dmgr\\websphere_backup\config\cells\sor
twork.
MIGR0344I: Processing configuration file
```

Figure 12-64 Migration wizard WASPostUpgrade starting

As with the command line version of **WASPostUpgrade**, you should review all the messages generated to make sure that your deployment manager migration has completed successfully. In our example, the final message was:

```
MIGR0271W: The migration completed with warnings.
```

We look at the migration log at C:\was51\backups\migration\dmgr, specifically looking for warning message codes that ended in "W." In our example, the following warnings were generated:

- `MIGR0401W: The Transports setting in file server.xml is deprecated.`
- `MIGR0331W: Port 9043 in file C:\WAS60\AppServer\profiles\SonomaDmgr01\config\cells\sonomaNetwork\nodes\sonomaManager\servers\dmgr\server.xml is migrated, but is already assigned in file com.ibm.websphere.migration.postupgrade.ApplicationServerHelper$PortMapping@21d5bca; this situation might result in port conflicts.`
- `MIGR0401W: The PMIService:initialSpecLevel setting in file server.xml is deprecated.`
- `MIGR0379W: Do not use the existing deployment manager in the old configuration. It has been disabled.`

> **Tip:** The last character in WebSphere log messages indicates the type of message. The possible values are:
>
> **A** Audit
> **I** Informational
> **E** Error
> **W** Warning
> **F** Fatal
> **O** System.out by user application or internal components
> **R** System.err by user application or internal components
> **U** A special type used by the message logging component of the WebSphere Application Server runtime
> **Z** A placeholder to indicate that the type was not recognized

After review of the messages and investigation of the possible port conflict, we determine that we are ready to start the V6 deployment manager to start managing our V5.1 cell. All the other messages in the log end in “I,” indicating that they are informational messages.

After starting the deployment manager with the **`StartManager`** command, we invoke the adminconsole application from a Web browser with the URL:

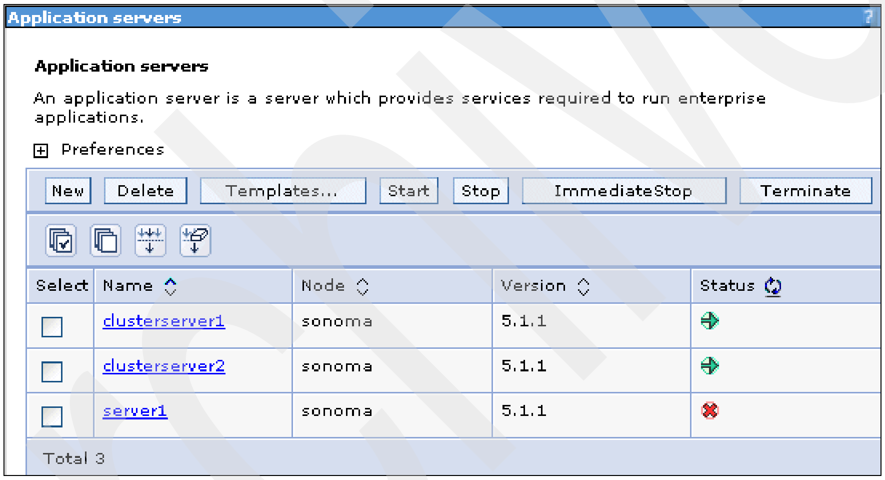
`http://sonoma:9090/admin`

which redirects to:

`http://sonoma:9090/ibm/console`

This is the new URL for Version 6. Note that even though the default Version 6 administration port is 9060, we choose to retain 9090 while selecting options in the migration wizard.

As shown in Figure 12-65, we navigate to the Application servers dialog (**Servers** → **Application servers**). We can see the two application servers in our cluster running and we can also see that they are V5.1.1 application servers. It is worth noting that, during the conversion of the deployment manager, we did not stop the node agent or the application servers running on our node. As a result, application availability was not impacted during this portion of the migration.



Application servers

An application server is a server which provides services required to run enterprise applications.

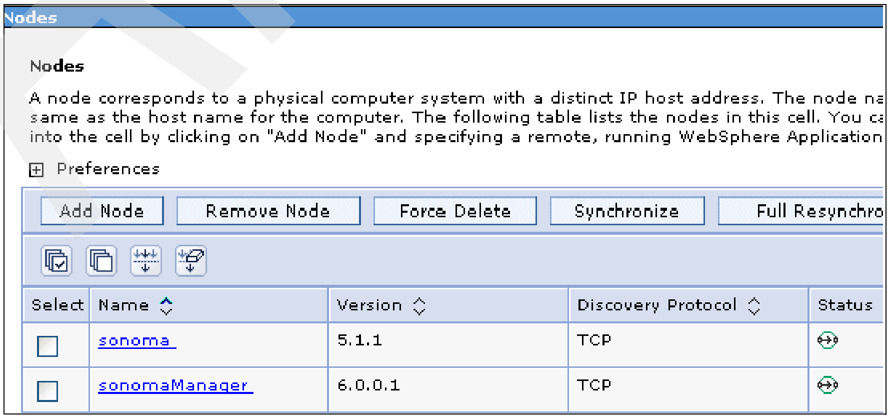
⊞ Preferences

Select	Name	Node	Version	Status
□	clusterserver1	sonoma	5.1.1	➔
□	clusterserver2	sonoma	5.1.1	➔
□	server1	sonoma	5.1.1	✖

Total 3

Figure 12-65 Version 6 console showing servers

When we look at the nodes in our cell (Figure 12-66) we see the managed node, sonoma, is a V5.1.1 node, but the deployment manager node, sonomaManager, is a V6.0 node.



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node na
same as the host name for the computer. The following table lists the nodes in this cell. You ca
into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application

⊞ Preferences

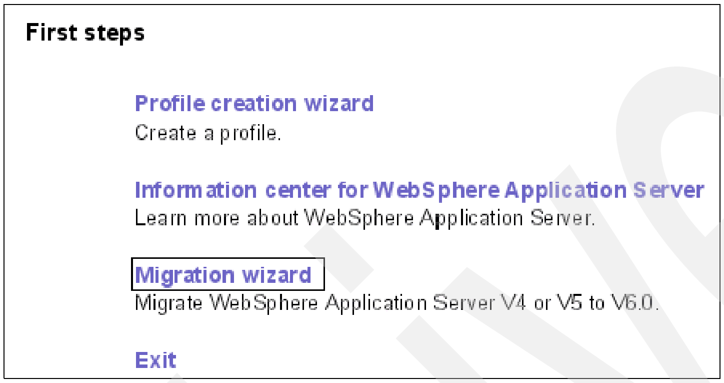
Select	Name	Version	Discovery Protocol	Status
□	sonoma	5.1.1	TCP	↔
□	sonomaManager	6.0.0.1	TCP	↔

Figure 12-66 Version 6 console showing nodes in cell

12.5.5 Managed node migration using the migration wizard

Now that you have a V6.0 deployment manager managing your cell, you can start to migrate your managed nodes. In our example, we are only dealing with a single managed node, while in a production environment multiple nodes would exist, which would allow us to stop and migrate one node (and the application servers on it) at a time, while the remaining nodes (and application servers) continue to serve applications requests.

Start the migration wizard by navigating to the firststeps directory under the custom profile for the managed node that you created earlier, and select the **Migration wizard (**Figure 12-67). Note that the First Steps Java client for a managed node profile differs from the First Steps Java client for a deployment manager profile.



First steps

Profile creation wizard
Create a profile.

Information center for WebSphere Application Server
Learn more about WebSphere Application Server.

Migration wizard
Migrate WebSphere Application Server V4 or V5 to V6.0.

Exit

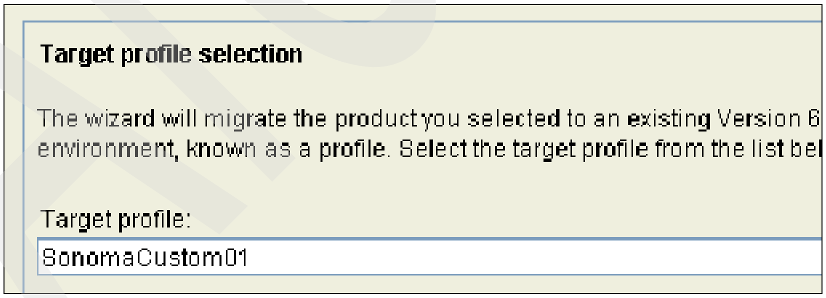
Figure 12-67 First steps for node migration

Proceed through the migration wizard as you did for the deployment manager migration. First, select the V5.1 version that you are going to migrate, then specify a backup directory.

```
C:\WAS51\backups\migration\sonomanode
```

If you were migrating multiple nodes, you would specify a different backup directory for each of your nodes in this fashion.

Select the target profile for your migration, as shown in Figure 12-68. By starting First Steps and the migration wizard from the profile, the profile name is populated automatically.



Target profile selection

The wizard will migrate the product you selected to an existing Version 6
environment, known as a profile. Select the target profile from the list bel

Target profile:

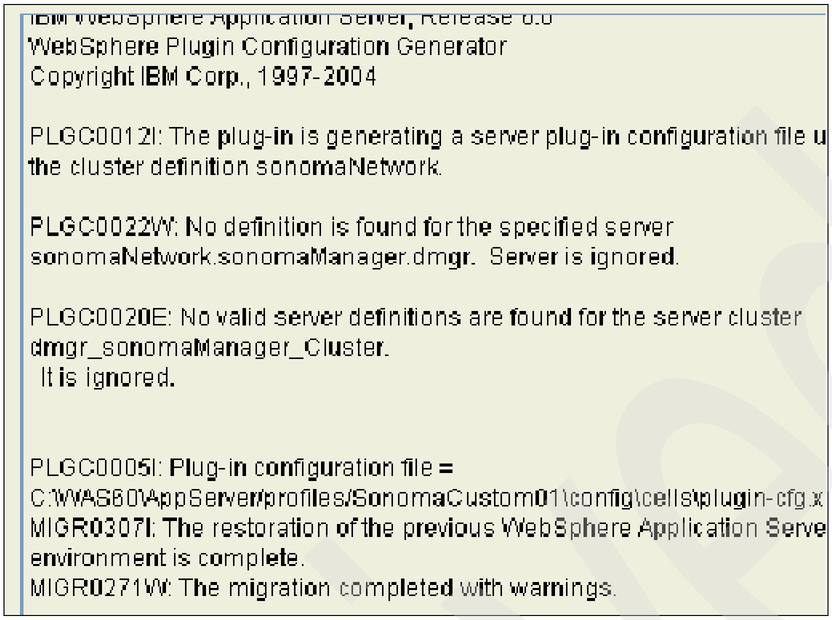
SonomaCustom01

Figure 12-68 Migration wizard target profile for node

The migration wizard then runs the **`WASPreUpgrade`** script.

Until now, the application servers on the node have continued to run in order to maximize availability. Before starting the next step in the migration wizard, which runs the **`WASPostUpgrade`** script, you must stop the application servers and the node agent on the node being migrated. If you forget to do this, the migration wizard stops the node agent and application servers if they are not already stopped. Once the migration wizard completes

WASPostUpgrade, as shown in Figure 12-69, you should look at the migration logs to see what warnings or errors were generated during the migration.



```
IBM WebSphere Application Server, Release 6.0
WebSphere Plugin Configuration Generator
Copyright IBM Corp., 1997-2004

PLGC0012I: The plug-in is generating a server plug-in configuration file u
the cluster definition sonomaNetwork.

PLGC0022W: No definition is found for the specified server
sonomaNetwork.sonomaManager.dmgr.  Server is ignored.

PLGC0020E: No valid server definitions are found for the server cluster
dmgr_sonomaManager_Cluster.
 It is ignored.


PLGC0005I: Plug-in configuration file =
C:\WAS60\AppServer/profiles/SonomaCustom01\config\cells\plugin-cfg.x
MIGR0307I: The restoration of the previous WebSphere Application Serve
environment is complete.
MIGR0271W: The migration completed with warnings.
```

Figure 12-69 Migration wizard completion for node

As you can see, the migration has completed with warnings. It is important that you examine the warnings to determine if you need to take any corrective action before proceeding. In this example, we see the following messages:

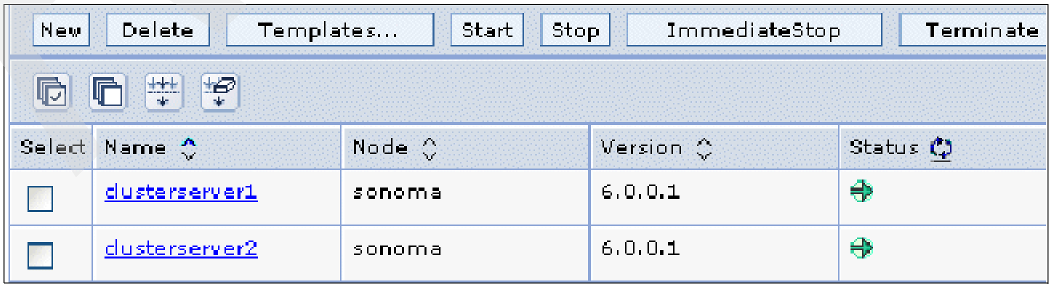
- `MIGR0404W: Do not use the node agent in the old configuration. It has been disabled.`
- `PLGC0020E: No valid server definitions are found for the server cluster dmgr_sonomaManager_Cluster. It is ignored.`

Neither of these messages requires any corrective action.

At this point, you are ready to start the node agent with the **startNode** command:

```
cd /d c:\WAS60\AppServer\profiles\SonomaCustom01\bin
startNode
```

Once the node agent is started, you can return to the administrative console and start the application servers on the node that are part of the server cluster. This is shown in Figure 12-70.



Select	Name	Node	Version	Status
☐	clusterserver1	sonoma	6.0.0.1	➡
☐	clusterserver2	sonoma	6.0.0.1	➡

Figure 12-70 Version 6 console showing servers migrated to Version 6

At this point, you are ready to test the applications running on the application servers to make sure that they are functioning correctly. Do this by invoking the Web application directly on the Web container HTTP port and validating that the application is functioning correctly. Once this is done, you are ready to place this node into service. Since the **WasPostUpgrade** script already regenerated the plug-in for you, all you need to do is to distribute the plugin-cfg-xml file to the HTTP servers in order to allow client requests to flow to your newly upgraded V6 node, application servers, and application.

Depending on whether or not you have chosen to create managed nodes for your HTTP servers, the mechanism for distributing the updated plug-in configuration file varies. Your three options are:

- Install WebSphere Application Server Version 6 and create a custom profile so that you have a node agent to manage your HTTP server, plug-in configuration file, and plug-in file distribution.
- Install IIBM HTTP Server Version 6, which provides the noted HTTP server and plug-in maintenance without a node agent.
- Continue to manually manage your HTTP server and plug-in maintenance.

At this point, you have successfully completed migration of your deployment manager and one of your managed nodes. In a multi-node environment you would repeat the managed node migration steps for each node in your deployment.

12.6 Manual migration: Installation of J2EE 1.3 Enterprise Application on V6 Application Server

This example uses the same application discussed in 12.3, “Automatic migration: Application Server V5 to Express V6” on page 191. That example shows a migration using the automatic migration utilities; this example shows how to install and configure that same application manually. In other words, we do not start with the application already installed on a system. We instead take the application EAR file and install it using the administrative console. We then configure the required resources with the administrative console.

12.6.1 Preparing the V6 installation

A V6 application server is created using the profile creation wizard. This example uses an installation of WebSphere Application Server Network Deployment Version 6 and creates an application server profile using the default ports. This example should work equally well on any of the WebSphere server types, including a deployment manager. We do not show the details of installing and configuring the profile. See 8.2.3, “Profiles” on page 105 for an overview of these steps.

Before getting to the basics of configuration in the administrative console, you first need to start the server and administrative console if you have not already done so.

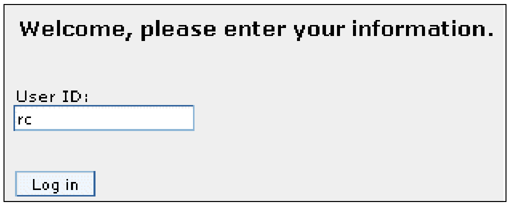
Start the application server:

```
cd /d e:\WebSphere_V6\profiles\olepaint\bin
startServer server1
```

Start the administrative console in your Web browser:

```
http://localhost:9060/ibm/console
```

Log in using any name you wish. Figure 12-71 shows the login to the administrative console. The name is an arbitrary string that tracks the changes you make before you commit and save.



Welcome, please enter your information.

User ID: rc

Figure 12-71 Administrative console login

Configuring JMS resources

1. Create an SI Bus.

   a. Navigate to the Service Integration section and open **Buses** (Figure 12-72).

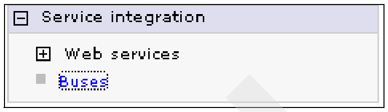
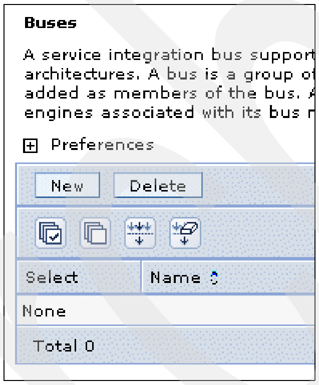


Figure 12-72 Creating SI bus - Service integration - Buses

   b. Create a new bus with the **New** button (Figure 12-73).



Buses

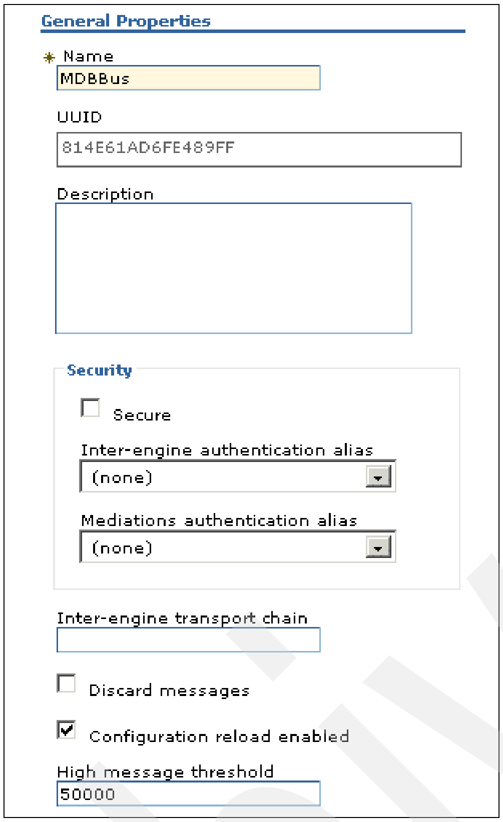
A service integration bus support
architectures. A bus is a group of
added as members of the bus. A
engines associated with its bus r

Select	Name
None	
Total 0	

Figure 12-73 Creating SI bus - New bus

   c. Specify the following fields in the new bus window (Figure 12-74 on page 228) and accept defaults for the remainder of the fields. Click **OK** when done.

   - Name: `MDBbus`
   - Secure: Deselect



General Properties

* Name
MDBBus

UUID
814E61AD6FE489FF

Description

Security

□ Secure

Inter-engine authentication alias
(none)

Mediations authentication alias
(none)

Inter-engine transport chain

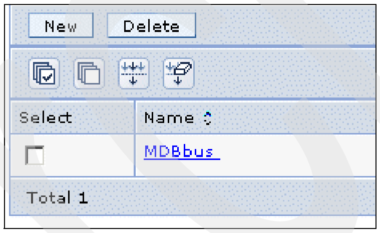
□ Discard messages

☑ Configuration reload enabled

High message threshold
50000

Figure 12-74 Creating SI bus - General properties

d. Reselect **MDBbus** to specify further attributes (Figure 12-75).

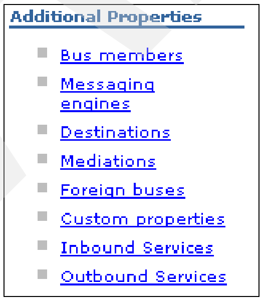


Select	Name
□	MDBbus

Total 1

Figure 12-75 Creating SI bus - New bus MDBbus

e. Select **Bus Members** under Additional Properties as shown in Figure 12-76.

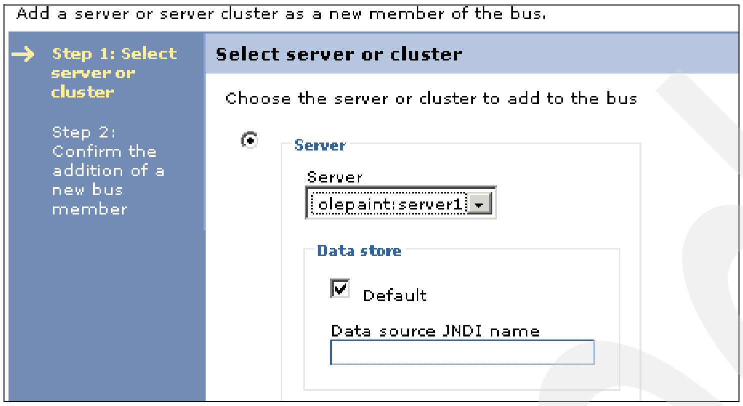


Additional Properties

- Bus members
- Messaging engines
- Destinations
- Mediations
- Foreign buses
- Custom properties
- Inbound Services
- Outbound Services

Figure 12-76 Creating SI bus - Bus members

f. Add a new member with the **Add** button. Specify `Server` as the member type and select the server name (Figure 12-77). Our example only has server1 available to select. Accept defaults for the remaining fields on that screen. Select **Next** when done.



Add a server or server cluster as a new member of the bus.

→ Step 1: Select server or cluster

Step 2: Confirm the addition of a new bus member

Select server or cluster

Choose the server or cluster to add to the bus

● Server

Server: olepaint:server1

Data store

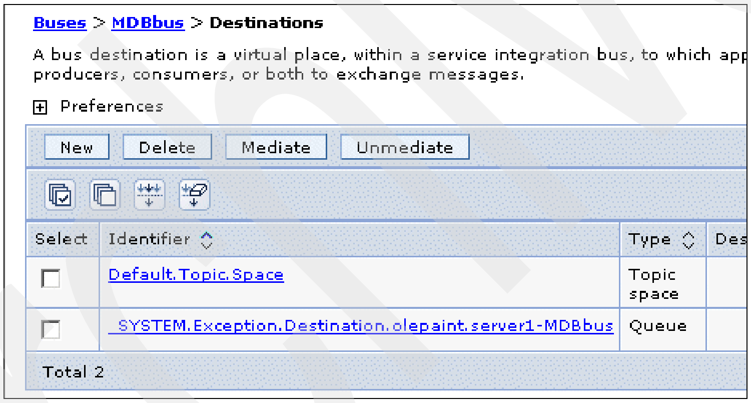
☑ Default

Data source JNDI name

Figure 12-77   Create SI bus - Select server

g. Select **Finish** on the next screen.

h. Reselect **MDBbus** and then **Destinations** under Additional Properties. Create a new destination with the **New** button (Figure 12-78).



Buses > MDBbus > Destinations

A bus destination is a virtual place, within a service integration bus, to which app producers, consumers, or both to exchange messages.

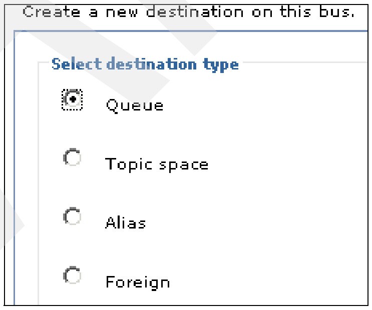
⊞ Preferences

New | Delete | Mediate | Unmediate

Select	Identifier	Type	Des
□	Default.Topic.Space	Topic space	
□	_SYSTEM.Exception.Destination.olepaint.server1-MDBbus	Queue	
Total 2			

Figure 12-78   Create SI bus - New destination

i. Select **Queue** as the destination type (Figure 12-79).



Create a new destination on this bus.

Select destination type

● Queue

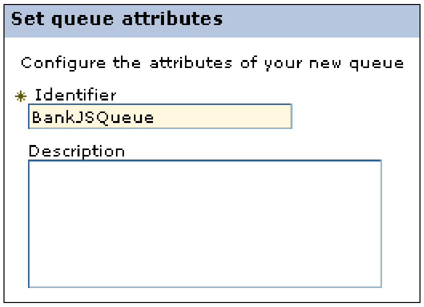
○ Topic space

○ Alias

○ Foreign

Figure 12-79   Creating SI bus - Selecting destination type

j. In Set queue attributes, specify `BankJSQueue` as the Identifier (Figure 12-80). Note that this name must match the Queue name shown in Figure 12-86 on page 232.



Set queue attributes

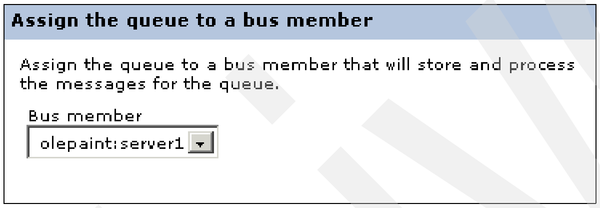
Configure the attributes of your new queue

* Identifier
BankJSQueue

Description

Figure 12-80 Creating SI bus - Adding queue destination ID

k. Assign the queue to the existing bus members (Figure 12-81). This example only has server1 available as a bus member.



Assign the queue to a bus member

Assign the queue to a bus member that will store and process the messages for the queue.

Bus member
olepaint:server1

Figure 12-81 Creating SI bus - Assign queue to bus member

l. Select **Next** when done. On the next screen, select **Finish**.

That concludes creating the SI bus resources. Once you have an SI bus created, you can attach multiple destinations and associate multiple JMS resources to those destinations. We show you how to create JMS resources in the next step.

2. Create a JMS Queue Connection Factory.

a. Navigate to **Resources** → **JMS Providers** → **Default Messaging** (Figure 12-82).

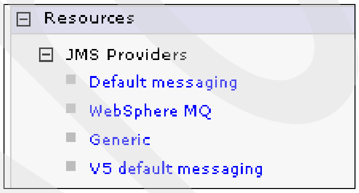


Figure 12-82 Creating JMS connection factory - Default messaging

b. Under Connection Factories, select **JMS Queue Connection Factory** (Figure 12-83).

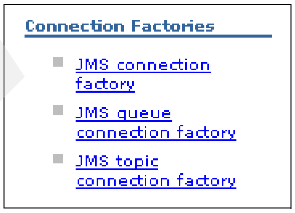
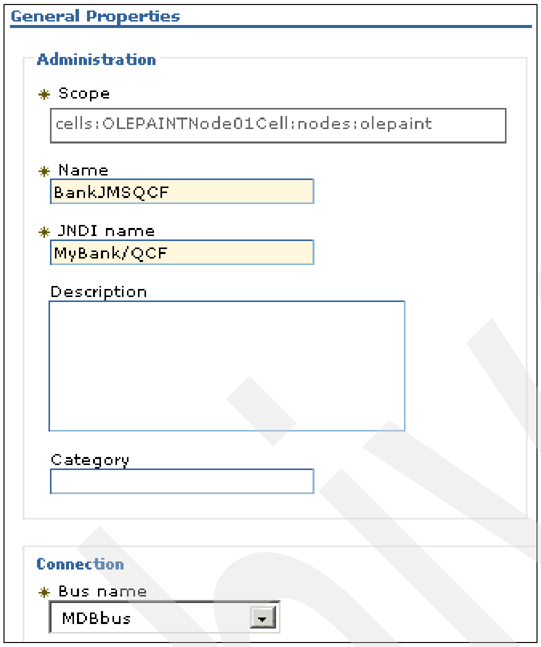


Figure 12-83 Creating JMS connection factory - Connection factories

c. In the next screen, click **New**. Specify the following fields and accept the defaults for the remaining fields. Click **OK** when done.

Name	`BankJMSQCF`
JNDI Name	`MyBank/QCF`
Bus Name	`MDBbus`

Note that the JNDI name must match the name used by the application.



General Properties

Administration

* Scope: cells:OLEPAINTNode01Cell:nodes:olepaint
* Name: BankJMSQCF
* JNDI name: MyBank/QCF

Description

Category

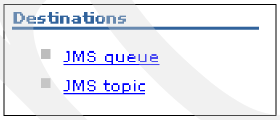
Connection

* Bus name: MDBbus

Figure 12-84 Creating JMS queue connection factory - General properties

3. Create a JMS Queue.

a. Navigate to **Default Messaging Providers** → **Destinations** → **JMS Queue**.



Destinations

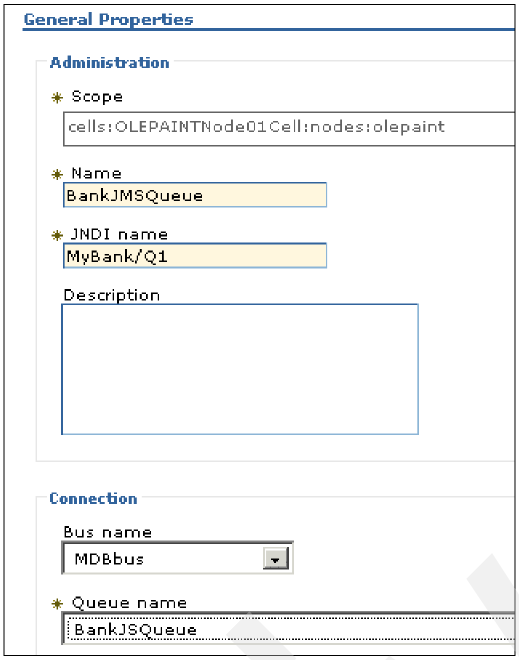
- JMS queue
- JMS topic

Figure 12-85 Creating JMS queue - Destinations

b. In the next screen, click **New**. Specify the following fields (Figure 12-86 on page 232), and accept the defaults for the remaining fields. Click **OK** when done.

Name	`BankJMSQueue`
JNDI Name	`MyBank/Q1`
Bus Name	`MDBbus`
Queue Name	`BankJSQueue`

Note that the JNDI Name must match the name in Figure 12-88 on page 233. Also, the Queue Name must match the name in Figure 12-80 on page 230.



General Properties

Administration

* Scope
cells:OLEPAINTNode01Cell:nodes:olepaint

* Name
BankJMSQueue

* JNDI name
MyBank/Q1

Description

Connection

Bus name
MDBbus

* Queue name
BankJSQueue

Figure 12-86 Create JMS queue - General properties

4. Create a JMS activation specification
   a. Navigate to **Default Messaging Providers** → **Activation Specifications** → **JMS Activation Specification** (Figure 12-87).

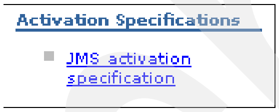
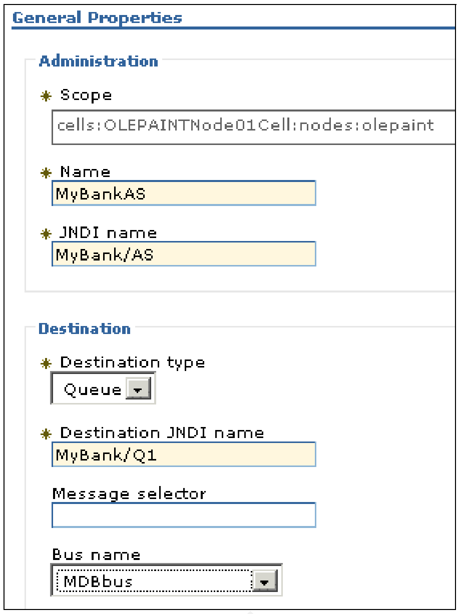


Figure 12-87 Create JMS activation specification - Navigation

   b. Create a new specification by clicking the **New** button. In the new activation specification screen (Figure 12-88 on page 233), specify these fields and accept the defaults for the remaining fields. Select **OK** when done.

Name	`MyBankAS`
JNDI Name	`MyBank/AS`
Destination type	`Queue`
Destination JNDI Name	`MyBank/Q1`
Bus Name	`MDBbus`

Note that the Destination JNDI name must match the JNDI Name of the JMS queue destination that you configured in Figure 12-86.



General Properties

Administration

* Scope
cells:OLEPAINTNode01Cell:nodes:olepaint

* Name
MyBankAS

* JNDI name
MyBank/AS

Destination

* Destination type
Queue

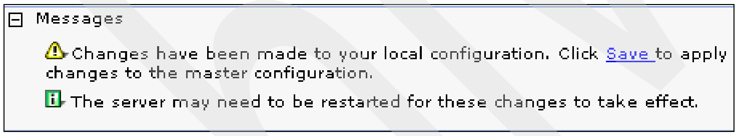
* Destination JNDI name
MyBank/Q1

Message selector

Bus name
MDBbus

Figure 12-88 Create JMS activation specification - General properties

c. Save the configuration by clicking the **Save** link at the top of the screen. Click **Save** again to confirm the save, as shown in Figure 12-89.



Messages

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

ℹ The server may need to be restarted for these changes to take effect.

Figure 12-89 Saving configuration

Configuring JDBC resources

1. Create an authentication alias.

   a. Navigate to **Global Security** under **Security** in the left navigation bar. On the Global Security screen, select **J2C Authentication Alias** under Authentication, as shown in Figure 12-90.

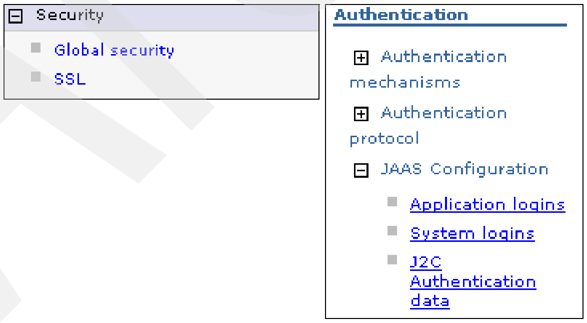
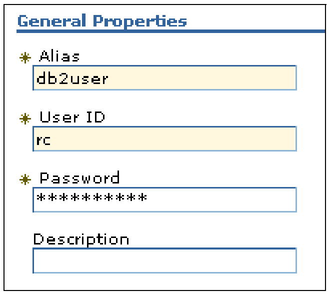


Figure 12-90 Creating authentication alias - Navigation

b. Create a new alias with the **New** button and specify the following fields (Figure 12-91 on page 234). Click **OK** when done.

Alias	`db2user`
User ID	Specify your own DB2 user
Password	Specify the matching password for the DB2 user



General Properties

* Alias
db2user

* User ID
rc

* Password
****

Description

Figure 12-91 Creating Authentication Alias - General Properties

2. Create a JDBC Provider.

a. Navigate to **Resources** → **JDBC Providers** in the left navigation bar (Figure 12-92).

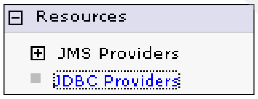
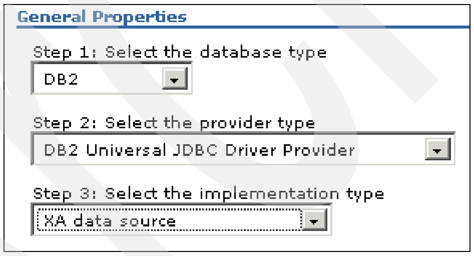


Figure 12-92 Creating JDBC provider - Navigation

b. Create a new JDBC provider with the **New** button. Select these settings from the selector fields, as shown in Figure 12-93. Click **OK** when done.

Database type	`DB2`
Provider type	`DB2 Universal JDBC Driver Provider`
Implementation type	`XA data source`



General Properties

Step 1: Select the database type
DB2

Step 2: Select the provider type
DB2 Universal JDBC Driver Provider

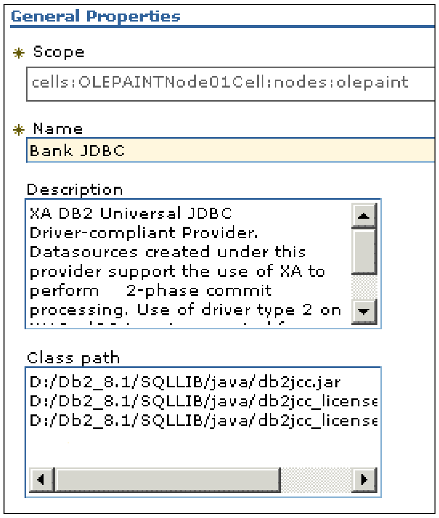
Step 3: Select the implementation type
XA data source

Figure 12-93 Creating JDBC provider - Type selection

c. On the next screen, as shown in Figure 12-94 on page 235, specify a unique provider name and the correct classpath for your DB2 JDBC drivers. Do not rely on the settings you see defaulted in the classpath. The variables identifying the DB2 driver location may or may not be correct, depending on what order you installed DB2 in relation to the application server. Specify a full path name for all the concerned files, as shown here. You do not have to worry about which direction the slash file delimiters are pointing. You can use either forward or reverse slashes, regardless of the operating system you are using.

```
D:/Db2_8.1/SQLLIB/java/db2jcc.jar
D:/Db2_8.1/SQLLIB/java/db2jcc_license_cu.jar
D:/Db2_8.1/SQLLIB/java/db2jcc_license_cisuz.jar
```

For this example, we specify a provider name of `Bank JDBC` and the correct JDBC driver path.



General Properties

* Scope
cells:OLEPAINTNode01Cell:nodes:olepaint

* Name
Bank JDBC

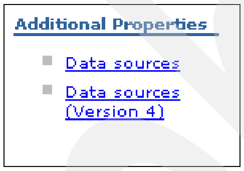
Description
XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on

Class path
D:/Db2_8.1/SQLLIB/java/db2jcc.jar
D:/Db2_8.1/SQLLIB/java/db2jcc_license
D:/Db2_8.1/SQLLIB/java/db2jcc_license

Figure 12-94 Creating JDBC provider - General properties

3. Create a data source.

   a. Select the **Bank JDBC** provider you just created. In the next screen, shown in Figure 12-95, select **Data sources**.



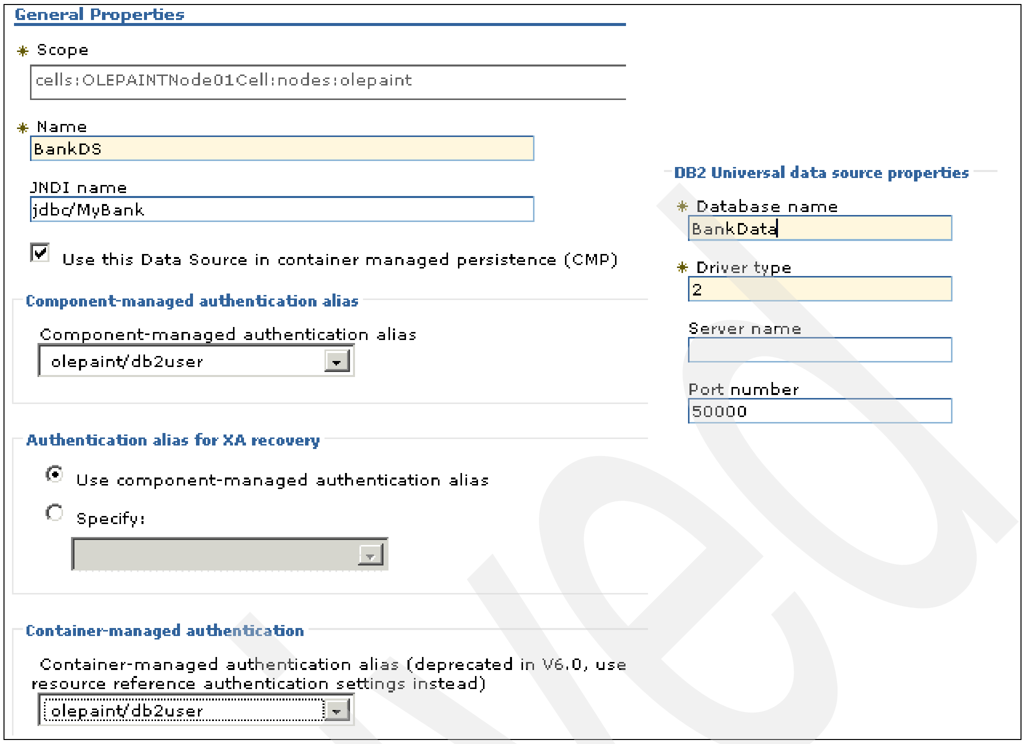
Additional Properties

- Data sources
- Data sources (Version 4)

Figure 12-95 Creating data source - Navigation

   b. On the next screen, shown in Figure 12-96 on page 236, specify the following fields and accept the defaults for the remaining fields.

Name	`BankDS`
JNDI Name	`jdbc/MyBank`
Use this Data Source in container managed persistence	Checked
Container managed authentication alias	`db2user`
Database Name	`BankData`



General Properties

* Scope
cells:OLEPAINTNode01Cell:nodes:olepaint

* Name
BankDS

JNDI name
jdbc/MyBank

☑ Use this Data Source in container managed persistence (CMP)

Component-managed authentication alias

Component-managed authentication alias
olepaint/db2user

Authentication alias for XA recovery

● Use component-managed authentication alias

○ Specify:

Container-managed authentication

Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)
olepaint/db2user

DB2 Universal data source properties

* Database name
BankData

* Driver type
2

Server name

Port number
50000

Figure 12-96 Creating data source - Properties

c. Save the configuration again, as we did in a previous step in Figure 12-89 on page 233.

You are not strictly required to perform intermediate saves of the configuration, but we recommend saving at periodic intervals. Once you get in the habit of saving your configuration work, you will be less likely to forget to save at the end of the task.

12.6.2 Installing the application

The application installation wizard presents you with several steps in which you can override default information. The number of steps varies with the exact configuration of your application. We show you the screens where this particular example needs customization. We also show screens that you may need to change for your own particular application.

For a more complete tutorial on application installation, consult the IBM Education Assistant for the application installation topic. This topic has both charts and browser animation for application installation. You can find this information at:

`ftp://ftp.software.ibm.com/software/eod/WAS_6-0/SystemManagement/index.html`

For all these steps, you can navigate backward and forward by either selecting the **Previous** or **Next** buttons, or selecting the explicit step number on the left side of the screen. Figure 12-97 on page 237 shows a list of all the steps we encounter for this application. You may encounter a different number of steps with a different application. By selecting the step number, you can select steps in any order. The summary step is the last step, where you finish and commit all your answers.

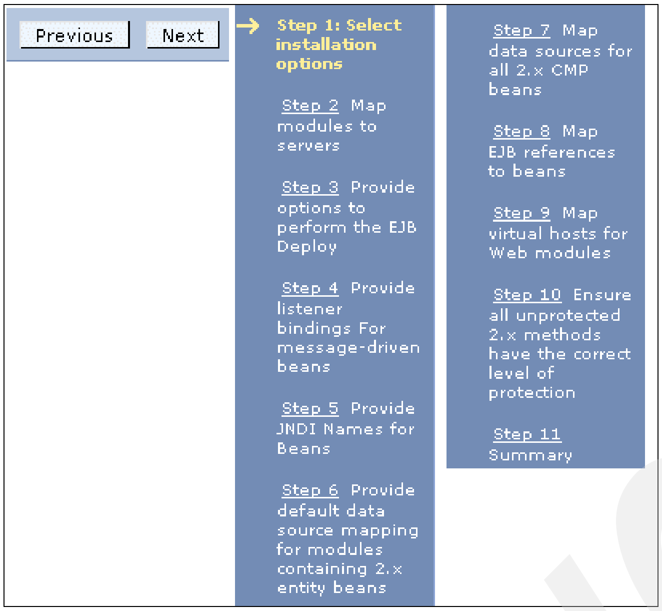

Previous
Next
Step 1: Select installation options
Step 2 Map modules to servers
Step 3 Provide options to perform the EJB Deploy
Step 4 Provide listener bindings For message-driven beans
Step 5 Provide JNDI Names for Beans
Step 6 Provide default data source mapping for modules containing 2.x entity beans
Step 7 Map data sources for all 2.x CMP beans
Step 8 Map EJB references to beans
Step 9 Map virtual hosts for Web modules
Step 10 Ensure all unprotected 2.x methods have the correct level of protection
Step 11 Summary

Figure 12-97 Application install - Navigation though steps

1. Start the application installation wizard by selecting **Install New Applications** under the **Applications** section of the left navigation bar (Figure 12-98).

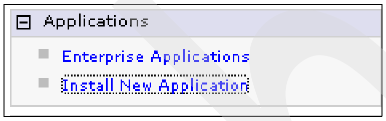
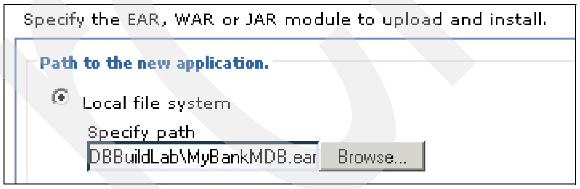


Figure 12-98 Application installation - Starting wizard

2. Specify the EAR file for the application (Figure 12-99).



Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

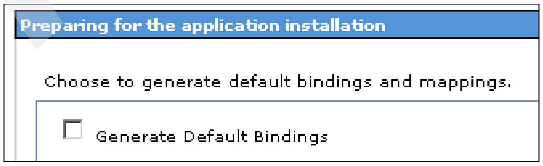
● Local file system

Specify path

DBBuildLab\MyBankMDB.ear

Figure 12-99 Application installation - Specify file

3. On the preparations page, do not check Generate Default Bindings, as shown in Figure 12-100. This effectively ignores the remaining fields on that page. Generating default bindings for this application is not necessary because the EAR file we use already has explicit bindings assigned. If your application was not previously bound, you could generate default bindings automatically.



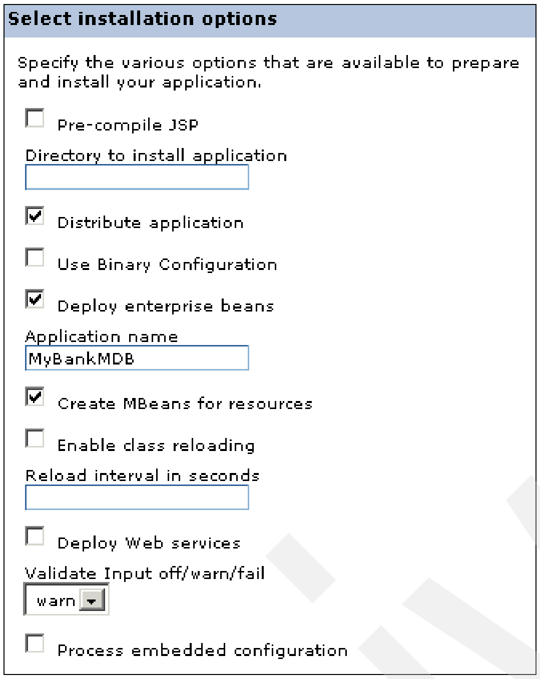
Preparing for the application installation

Choose to generate default bindings and mappings.

☐ Generate Default Bindings

Figure 12-100 Application installation - Default bindings

4. On the Select installation options page (Figure 12-101), accept the defaults, which are usually correct.



Select installation options

Specify the various options that are available to prepare and install your application.

□ Pre-compile JSP

Directory to install application

☑ Distribute application

□ Use Binary Configuration

☑ Deploy enterprise beans

Application name: MyBankMDB

☑ Create MBeans for resources

□ Enable class reloading

Reload interval in seconds

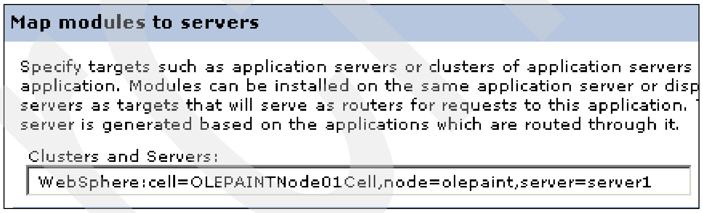
□ Deploy Web services

Validate Input off/warn/fail: warn

□ Process embedded configuration

Figure 12-101 Install application - Options

5. On the Map modules to servers page (Figure 12-102), accept the defaults. In the topology we use, there is only one server and therefore only one choice can be made. If you were installing on a deployment manager, you would have multiple servers from which to choose.



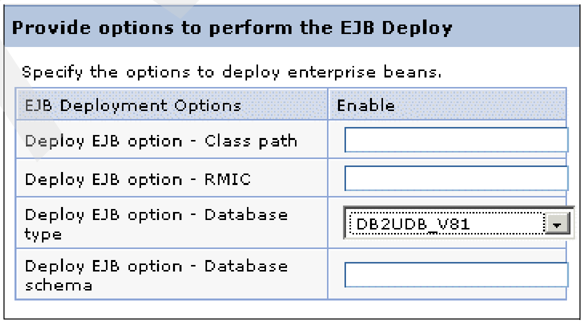
Map modules to servers

Specify targets such as application servers or clusters of application servers
application. Modules can be installed on the same application server or disp
servers as targets that will serve as routers for requests to this application.
server is generated based on the applications which are routed through it.

Clusters and Servers:
WebSphere:cell=OLEPAINTNode01Cell,node=olepaint,server=server1

Figure 12-102 Application install - Map servers

6. On the EJBDeploy options page (Figure 12-103), specify the data base type, which in this case is `IBM DB2 8.1`.



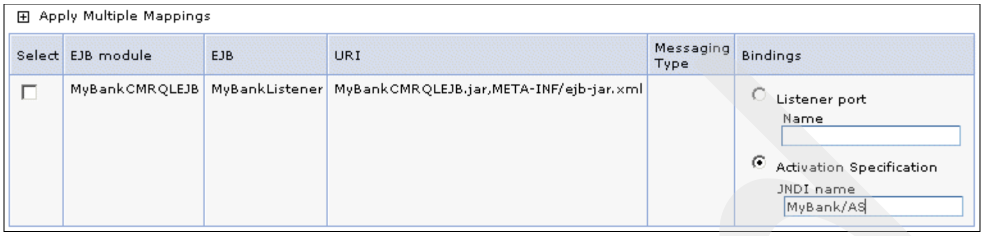
Provide options to perform the EJB Deploy

Specify the options to deploy enterprise beans.

EJB Deployment Options	Enable
Deploy EJB option - Class path	
Deploy EJB option - RMIC	
Deploy EJB option - Database type	DB2UDB_V81
Deploy EJB option - Database schema	

Figure 12-103 Application installation - EJB deploy options

7. On the MDB listener bindings page (Figure 12-104), specify either a Listener port or an Activation specification. This example specifies the activation specification with a JNDI name `MyBank/AS`. Note that this name must match the name specified in Figure 12-88 on page 233.

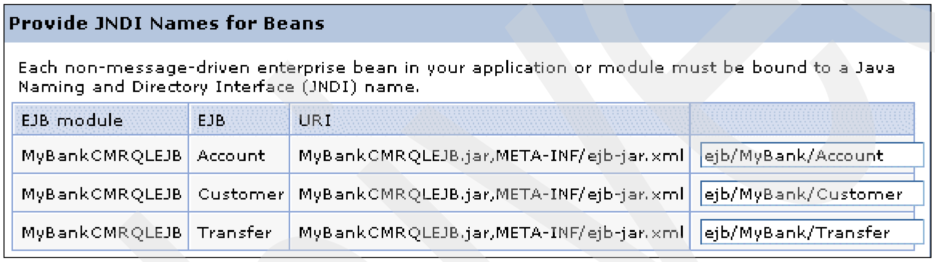


⊞ Apply Multiple Mappings

Select	EJB module	EJB	URI	Messaging Type	Bindings
□	MyBankCMRQLEJB	MyBankListener	MyBankCMRQLEJB.jar,META-INF/ejb-jar.xml		○ Listener port Name ● Activation Specification JNDI name MyBank/AS

Figure 12-104 Application installation - Specify Activation specification

8. On the JNDI Names for Beans page, shown in Figure 12-105, accept the default assignments.



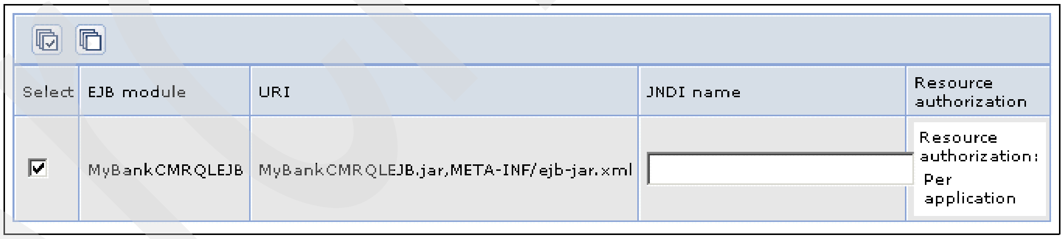
Provide JNDI Names for Beans

Each non-message-driven enterprise bean in your application or module must be bound to a Java Naming and Directory Interface (JNDI) name.

EJB module	EJB	URI	
MyBankCMRQLEJB	Account	MyBankCMRQLEJB.jar,META-INF/ejb-jar.xml	ejb/MyBank/Account
MyBankCMRQLEJB	Customer	MyBankCMRQLEJB.jar,META-INF/ejb-jar.xml	ejb/MyBank/Customer
MyBankCMRQLEJB	Transfer	MyBankCMRQLEJB.jar,META-INF/ejb-jar.xml	ejb/MyBank/Transfer

Figure 12-105 Application installation - Specify JNDI names

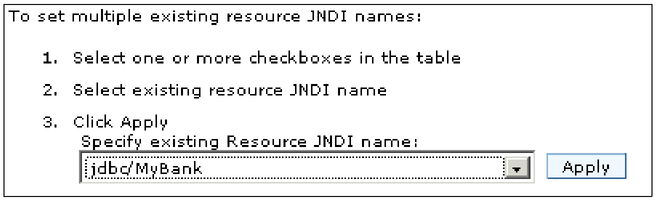
9. On the Default data sources page, shown in Figure 12-106, select a data source to apply to all the EJBs in the application. Scroll to the bottom of the page and select the EJB modules you want to apply a data source for. This example only has one EJB to choose.



Select	EJB module	URI	JNDI name	Resource authorization
☑	MyBankCMRQLEJB	MyBankCMRQLEJB.jar,META-INF/ejb-jar.xml		Resource authorization: Per application

Figure 12-106 Application install - Default data source - Step a

10. On the same page, scroll up to the top and select the data source which you want to apply as the default data source. This is shown in Figure 12-107 on page 240. Click the **Apply** button. For this example, we had already configured jdbc/MyBank; therefore, it can be selected from the pull-down menu. It is also possible to key the JNDI name value in case the data source has not yet been defined. You can scroll back down to the EJB modules and see that the JNDI name has been applied.


To set multiple existing resource JNDI names:

1. Select one or more checkboxes in the table
2. Select existing resource JNDI name
3. Click Apply

Specify existing Resource JNDI name: jdbc/MyBank

Figure 12-107 Application install - Default data source - Step b

11. The remainder of the steps are omitted because the default values are acceptable for this exercise.
12. Select the last summary step and click **Finish**. The installation begins. Observe the output and confirm that the installation wizard was successful.
13. Save the configuration as shown in Figure 12-89 on page 233.
14. Log out from the administrative console.
15. Restart the application server:

```
cd /d e:\WebSphere_V6\profiles\olepaint\bin
stopServer server1
startServer server1
```

12.6.3 Verifying the application operation

The example application MyBankMDB is tested by exercising the account create and transfer beans from the Web interface, thus proving the data source is configured correctly. The transfer bean is also accessed through the MDB interface, thus proving the JMS resources are configured correctly.

13

z/OS runtime migration examples

This chapter shows examples of migrating server configurations for the z/OS platform.

13.1 Migrating Network Deployment cell from V 5.1 to V6.0

Our example Network Deployment cell has the configuration shown in Figure 13-1.

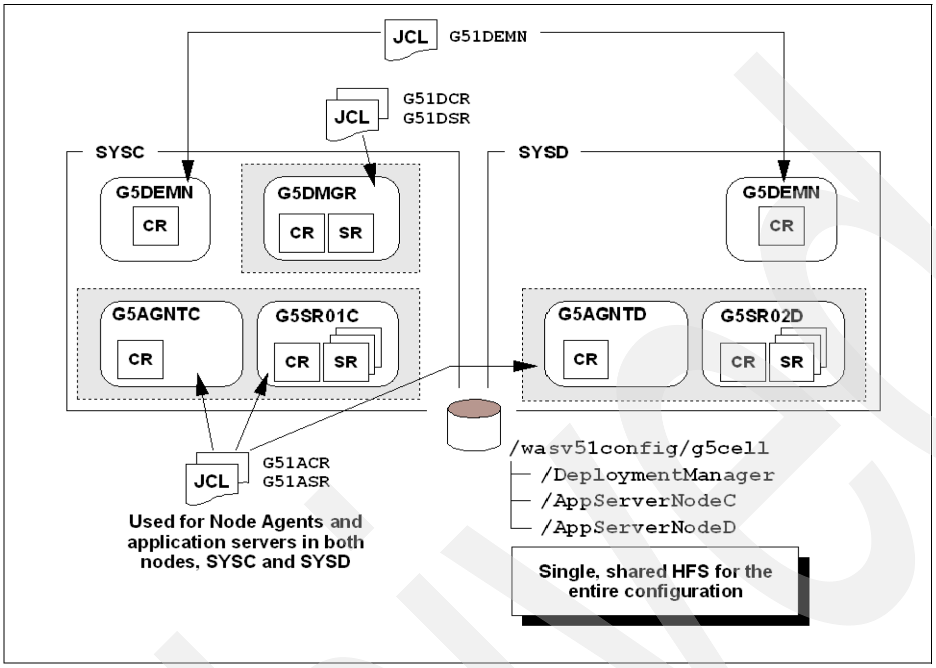


Figure 13-1 Schematic diagram of G5CELL

Description of configuration

The G5CELL is a Network Deployment configuration consisting of a deployment manager node on SYSC and two federated application server nodes, one on SYSC and one on SYSD. The entire configuration is kept in a single, shared HFS.

Common JCL is used. Both daemon servers use the G51DEMN start procedure. The deployment manager has its own procedures, G51DCR and G51DSR. All servers in the application server nodes use the same procedures: G51ACR and G51ASR. The shared HFS permits you to use the same JCL procedures. However, using common JCL has the potential to restrict the flexibility of a cell migration.

> **Important:** To state this issue more strongly, we suggest you try to avoid sharing the same JCL procedures. We had made this sharing decision when we first installed our V 5.1 configuration. We discovered that this was not a good idea once we were in the middle of migration. If you are stuck with the same situation, there are several ways to accommodate the problem. See 13.1.4, “Daemon migration problem” on page 273 for more details.

The following five tables (Table 13-1 through Table 13-5) summarize the important configuration data for our example cell. We refer to these tables throughout our discussion of this migration scenario.

Table 13-1 Cell parameters

Parameter	Value
Cell short name	G5CELL
Version of code	V5.1, maintenance W510207

Parameter	Value
HFS mount point	/wasv51config/g5cell
HFS data set name	OMVS.WAS51.G5CELL.CONFIG.HFS
Security Enabled	Yes
WebSphere Admin ID	G5ADMIN
WebSphere Admin ID password	G5ADMIN
WebSphere Configuration Group ID	G5CFG
Default RACF keyring name	WASKeyring

Table 13-2 Deployment Manager parameters

Parameter	Value
Configuration Root	/wasv51config/g5cell/DeploymentManager
Node short name	G5NODE
DMGR short name	G5DMGR
DMGR JOBNAME	G5DMGR
DMGR controller ID	G5DCRU
DMGR servant ID	G5DSRU
DMGR controller JCL procedure	G51DCR
DMGR servant JCL procedure	G51DSR
DMGR ENV= string	ENV=G5CELL.G5NODE.G5DMGR
Daemon controller ID	G5DEMN
Daemon controller JCL procedure	G5DEMN

Table 13-3 Application server on SYSC

Parameter	Value
Configuration Root	/wasv51config/g5cell/AppServerNodeC
Node short name	G5NODEC
Node Agent short name	G5AGNTC
Node Agent JOBNAME	G5AGNTC
Application server short name	G5SR01C
Application server JOBNAME	G5SR01C
Common controller ID	G5ACRU
Common servant ID	G5ASRU
Common controller JCL procedure	G51ACR
Common servant JCL procedure	G51ASR
Node Agent ENV= string	ENV=G5CELL.G5NODEC.G5AGNTC

Parameter	Value
G5SR01C ENV= string	ENV=G5CELL.G5NODEC.G5SR01C
Daemon controller ID	(uses same Daemon as DMGR)

Table 13-4 Application server on SYSD

Parameter	Value
Configuration Root	/wasv51config/g5cell/AppServerNodeD
Node short name	G5NODED
Node Agent short name	G5AGNTD
Node Agent JOBNAME	G5AGNTD
Application server short name	G5SR02D
Application server JOBNAME	G5SR02D
Common controller ID	G5ACRU (same as node on SYSC)
Common servant ID	G5ASRU (same as node on SYSC)
Common controller JCL procedure	G51ACR (same as node on SYSC)
Common servant JCL procedure	G51ASR (same as node on SYSC)
Node Agent ENV= string	ENV=G5CELL.G5NODED.G5AGNTD
G5SR01C ENV= string	ENV=G5CELL.G5NODED.G5SR01D
Daemon controller ID	G5DEMN
Daemon controller JCL procedure	G5DEMN

Table 13-5 Migration target information

Parameter	Value
HFS mount point	/wasv6config/g5cell
HFS data set name	OMVS.WAS6.G5CELL.MIGRATED.HFS

Planned sequence of node migration

1. The deployment manager node on SYSC is migrated first. When a cell has mixed version nodes, the deployment manager must always be at a higher level compared to the other nodes. The daemon server on SYSC is migrated at the same time as is the deployment manager node.
2. The application server node on SYSD was planned for migration next. Actually, either SYSC or SYSD could have been migrated next. We chose to migration SYSD next so we could illustrate how a V6.0 node (the deployment manager) and a V5.1 node (the application server node on SYSC) could coexist in the same cell on the same MVS image.
3. The application server node on SYSC was planned for migration next.

13.1.1 Migrating the deployment manager node on SYSC

What follows is a step-by-step account of how to migrate the deployment manager node.

Preliminary work

1. Take inventory of key information by recording the values shown in Table 13-1 through Table 13-5.
2. Back up the source configuration HFS. Example 13-1 is a sample backup job.

Example 13-1 Job to back up deployment manager configuration

```
//COPYHFS   JOB (ACCTNO,ROOM),'BAGWELL',CLASS=A,REGION=0M
//*
//* DFDSS TO COPY HFS DATASETS
//*
//COPY  EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//HFSOUT   DD UNIT=3390,DISP=OLD,VOL=SER=SMSA09
//SYSIN    DD *
 COPY DATASET(INCLUDE( -
     OMVS.WAS51.G5CELL.CONFIG.HFS -
            )) -
      RENAMEU( -
        OMVS.WAS51.G5CELL.CONFIG.HFS,OMVS.WAS51.G5CELL.MIGCOPY.HFS -
             ) -
      OUTDD(HFSOUT) -
      REPLACEU -
      CATALOG
/*
//
```

3. Back up the JCL start procedures used by the deployment manager node.

   The deployment manager node has six JCL members to back up:

   – G51DCR
   – G51DCRZ
   – G51DSR
   – G51DSRZ
   – G51DEMN
   – G51DEMNZ

   We back up the JCL because we intend to use the same names for the newly created V6 JCL start procedures. This reduces the amount of post-migration work required.

4. Ensure that G5CELL's administrator ID can work in the /tmp directory.

   In our case there is no /tmp/migrate directory, so that means the path is clear to invoke our migration job. The BBOWMG3* job attempts to copy the file bbomigrt2.sh into the /tmp/migrate directory and create another directory there. It needs write access to the /tmp/migrate directory and write access to the bbomigrt2.sh file. However, if it exists and is owned by some other ID, you have two options:

   – Delete the /tmp/migrate directory.
   – Set the permissions of /tmp/migrate to 777 and delete the bbomigrt2.sh file in that directory.

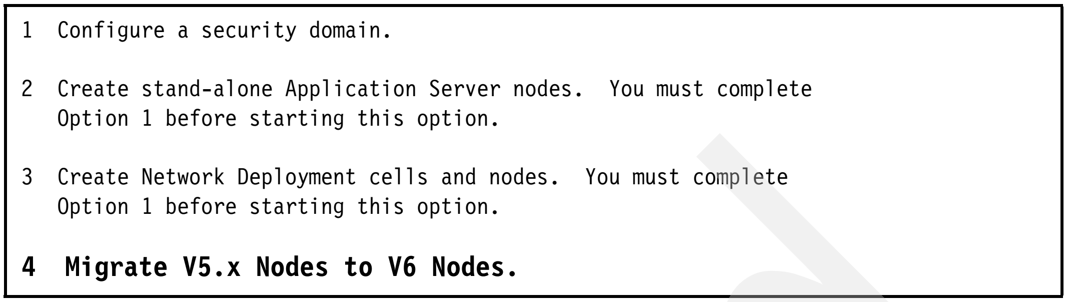
Invoke ISPF dialogs and customized migration jobs

1. Invoke the V6 ISPF customization dialogs:

   `'WAS600.WAS.SBBOCLIB(BBOWSTRT)' 'APPL(MIG)'`

   The `'APPL(MIG)'` provides a clean set of ISPF variables for this migration run.

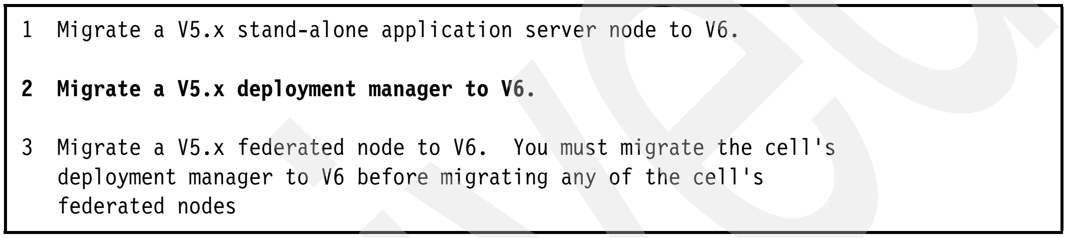
2. Select option **4** from the primary option menu as shown in Figure 13-2.



```
1  Configure a security domain.

2  Create stand-alone Application Server nodes.  You must complete
   Option 1 before starting this option.

3  Create Network Deployment cells and nodes.  You must complete
   Option 1 before starting this option.

4  Migrate V5.x Nodes to V6 Nodes.
```

Figure 13-2 Invoking customization dialogs for node migration

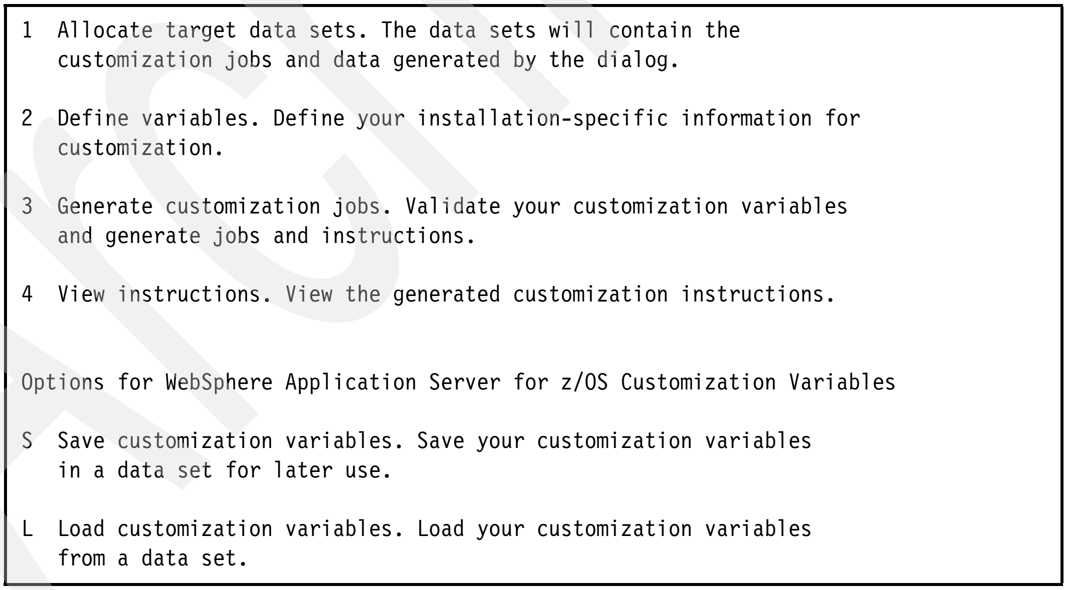
3. Selected option **2**, Migrate a V5.x deployment manager to V6 as shown in Figure 13-3.



```
1  Migrate a V5.x stand-alone application server node to V6.

2  Migrate a V5.x deployment manager to V6.

3  Migrate a V5.x federated node to V6.  You must migrate the cell's
   deployment manager to V6 before migrating any of the cell's
   federated nodes
```

Figure 13-3 Specifying deployment manager node

4. The next menu appears as shown in Figure 13-4. We navigate this dialog in a non-sequential order, referring back to this figure as we show you each step.

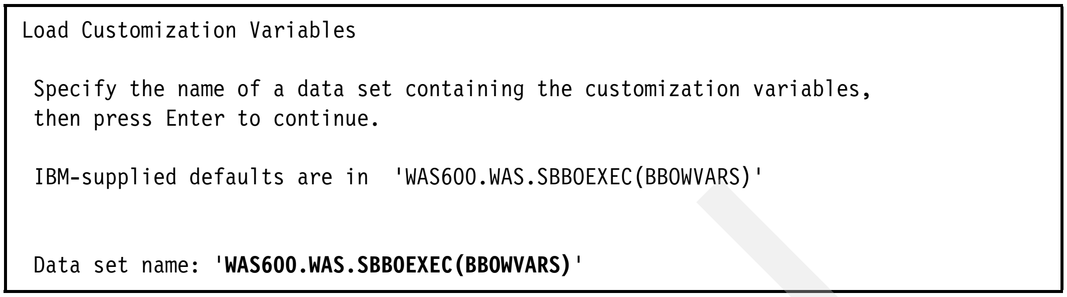


```
1  Allocate target data sets. The data sets will contain the
   customization jobs and data generated by the dialog.

2  Define variables. Define your installation-specific information for
   customization.

3  Generate customization jobs. Validate your customization variables
   and generate jobs and instructions.

4  View instructions. View the generated customization instructions.


Options for WebSphere Application Server for z/OS Customization Variables

S  Save customization variables. Save your customization variables
   in a data set for later use.

L  Load customization variables. Load your customization variables
   from a data set.
```

Figure 13-4 Options presented to create customized migration jobs

5. Load the default customization variables by selecting option **L** in Figure 13-4.

6. The next dialog is shown in Figure 13-5. For the application server nodes G5NODEC and G5NODED, use a copy of the saved variables from this customization run. That spares you from entering some of the same information several times. Load the default variables for the first node.



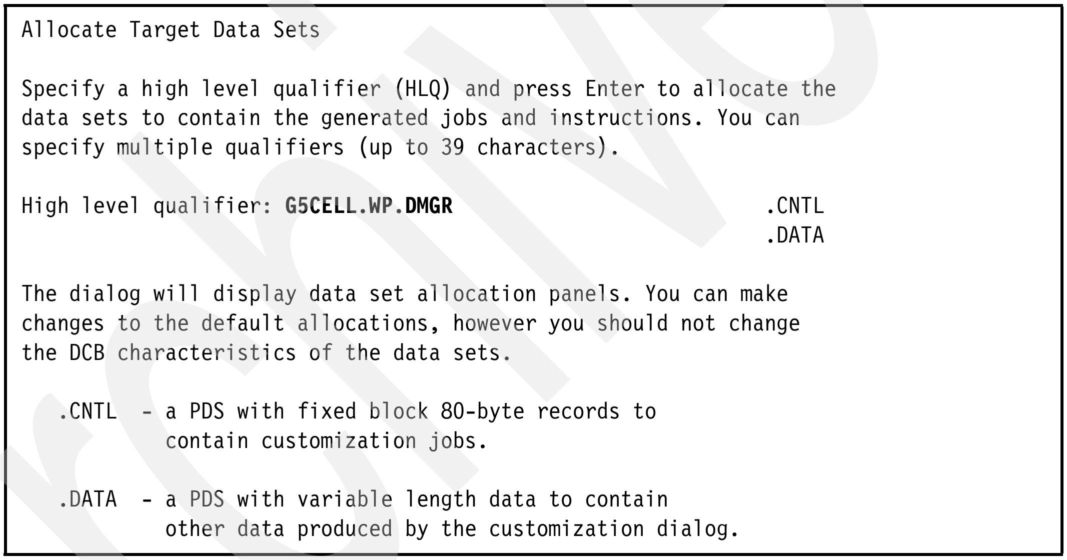
```
Load Customization Variables

 Specify the name of a data set containing the customization variables,
 then press Enter to continue.

 IBM-supplied defaults are in  'WAS600.WAS.SBBOEXEC(BBOWVARS)'


 Data set name: 'WAS600.WAS.SBBOEXEC(BBOWVARS)'
```

Figure 13-5 Default customization variables loaded

7. Choose option **1** in Figure 13-4, which results in display of the screen shown in Figure 13-6. This dialog allows you to allocate the target data sets where the customized jobs are stored. Use the default allocation parameters. Our scenario creates two data sets:

   – G5CELL.WP.DMGR.CNTL (a FB 80 PDS where the JCL is stored)

   – G5CELL.WP.DMGR.DATA (a VB 255 PDS where scripts are stored)

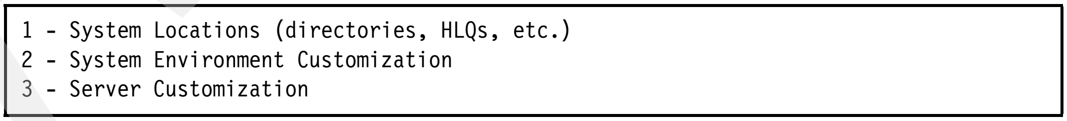


```
Allocate Target Data Sets

Specify a high level qualifier (HLQ) and press Enter to allocate the
data sets to contain the generated jobs and instructions. You can
specify multiple qualifiers (up to 39 characters).

High level qualifier: G5CELL.WP.DMGR                          .CNTL
                                                              .DATA

The dialog will display data set allocation panels. You can make
changes to the default allocations, however you should not change
the DCB characteristics of the data sets.

   .CNTL  - a PDS with fixed block 80-byte records to
            contain customization jobs.

   .DATA  - a PDS with variable length data to contain
            other data produced by the customization dialog.
```

Figure 13-6 Allocate target data sets

8. Select option **2**, Define Variables as shown in Figure 13-4.

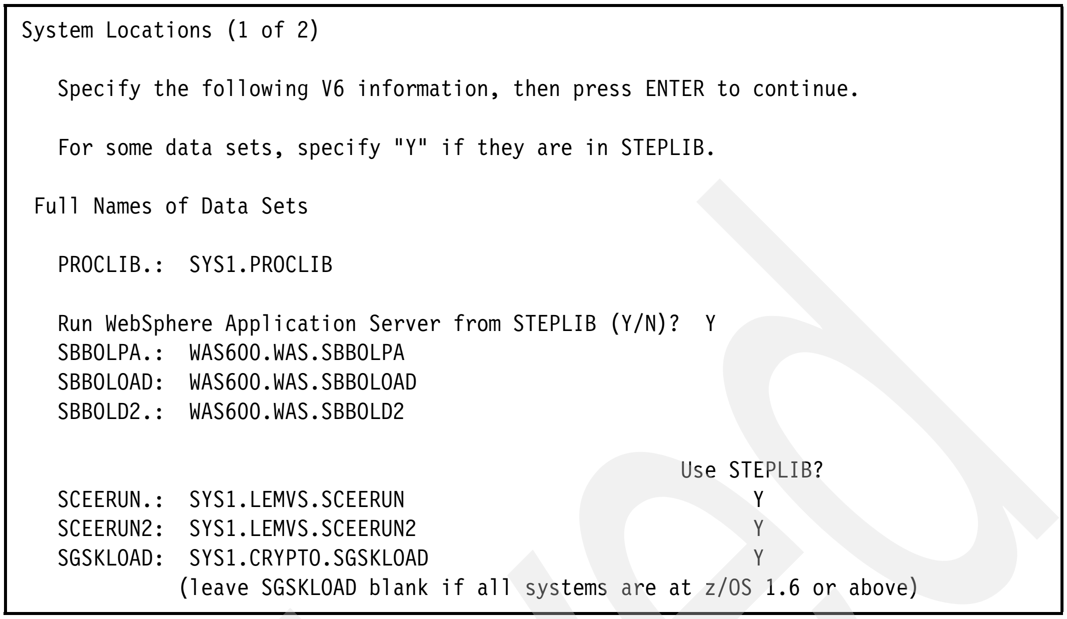
9. You now see the dialog shown in Figure 13-7.



```
1 - System Locations (directories, HLQs, etc.)
2 - System Environment Customization
3 - Server Customization
```

Figure 13-7 Defining variables

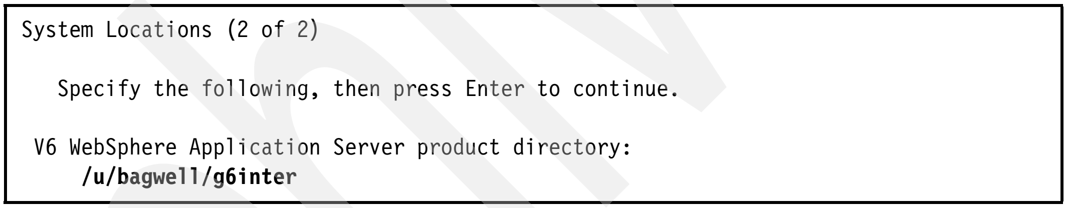
10. Select item **1**, which leads to the dialog in Figure 13-8. Choose to use or not use STEPLIB according to your local site preference.



```
System Locations (1 of 2)

   Specify the following V6 information, then press ENTER to continue.

   For some data sets, specify "Y" if they are in STEPLIB.

 Full Names of Data Sets

   PROCLIB.:  SYS1.PROCLIB

   Run WebSphere Application Server from STEPLIB (Y/N)?  Y
   SBBOLPA.:  WAS600.WAS.SBBOLPA
   SBBOLOAD:  WAS600.WAS.SBBOLOAD
   SBBOLD2.:  WAS600.WAS.SBBOLD2


                                                       Use STEPLIB?
   SCEERUN.:  SYS1.LEMVS.SCEERUN                             Y
   SCEERUN2:  SYS1.LEMVS.SCEERUN2                            Y
   SGSKLOAD:  SYS1.CRYPTO.SGSKLOAD                           Y
             (leave SGSKLOAD blank if all systems are at z/OS 1.6 or above)
```

Figure 13-8 Variables supplied for System Locations (1 of 2)

11. Populate the variables for System Locations (2 of 2) as shown in Figure 13-9.



```
System Locations (2 of 2)

   Specify the following, then press Enter to continue.

 V6 WebSphere Application Server product directory:
     /u/bagwell/g6inter
```

Figure 13-9 Variables supplied for System Locations (2 of 2)

Specify the directory mount point of the V6 SMP/E HFS, in other words, where WAS600.WAS.SBBOHFS was mounted. For our example, the actual location is /shared/zWebSphere/V6R0. You have the option of entering either the actual location, or a symbolic link.

The value /u/bagwell/g6inter represents an intermediate symbolic link we created that pointed to the real V6 product directory.

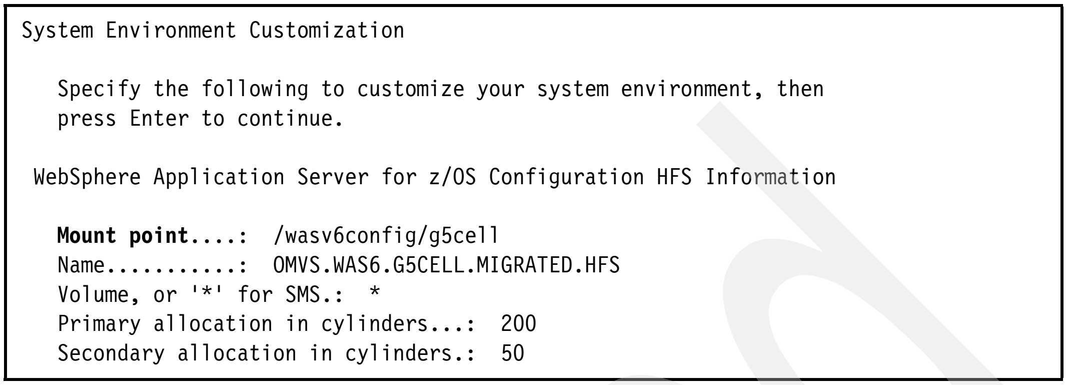
By pointing to an intermediate symbolic link here, the configuration symbolic links created during the migration all point to this intermediate symbolic link, which resolves to the actual mount point. We recommend that you use the symbolic link method because it provides the flexibility to isolate nodes for purposes like test, production, and maintenance.

For more details on the use of intermediate symbolic links and their role in providing flexibility see the white paper titled *WebSphere Application Server for z/OS V5 - Planning for Test, Production and Maintenance* located at this address:

`http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100396`

12. The System Environment Customization variables screen is accessed from option **2** in the dialog in Figure 13-7. Figure 13-10 shows the System Environment Customization dialog. This information populates the BBOWMDMT job, which is designed to create and mount the HFS in which the migrated configuration resides. The primary and secondary allocation values we use are lower than the default in order to save space.

You may decide to manually allocate the HFS and mount it yourself. Running the BBOWMDMT is not a strict requirement for migrating the node.

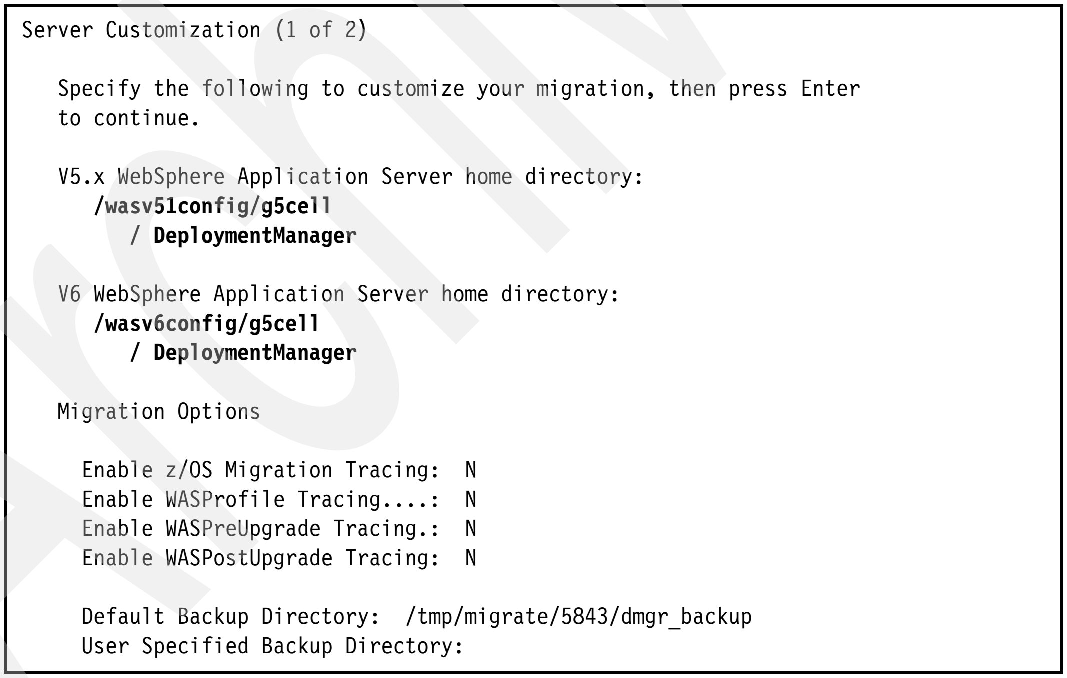


```
System Environment Customization

   Specify the following to customize your system environment, then
   press Enter to continue.

 WebSphere Application Server for z/OS Configuration HFS Information

   Mount point....:  /wasv6config/g5cell
   Name...........:  OMVS.WAS6.G5CELL.MIGRATED.HFS
   Volume, or '*' for SMS.:  *
   Primary allocation in cylinders...:  200
   Secondary allocation in cylinders.:  50
```

Figure 13-10 System Environment Customization variables - HFS file system information

Important: It is always necessary to provide the correct Mount point value on this panel, even if you don't plan to run the BBOWMDMT job. This is an important variable used throughout the migration jobs.

13. The first Server Customization screen is next, as shown in Figure 13-11.



```
Server Customization (1 of 2)

   Specify the following to customize your migration, then press Enter
   to continue.

   V5.x WebSphere Application Server home directory:
      /wasv51config/g5cell
         / DeploymentManager

   V6 WebSphere Application Server home directory:
      /wasv6config/g5cell
         / DeploymentManager

   Migration Options

     Enable z/OS Migration Tracing:  N
     Enable WASProfile Tracing....:  N
     Enable WASPreUpgrade Tracing.:  N
     Enable WASPostUpgrade Tracing:  N

     Default Backup Directory:  /tmp/migrate/5843/dmgr_backup
     User Specified Backup Directory:
```

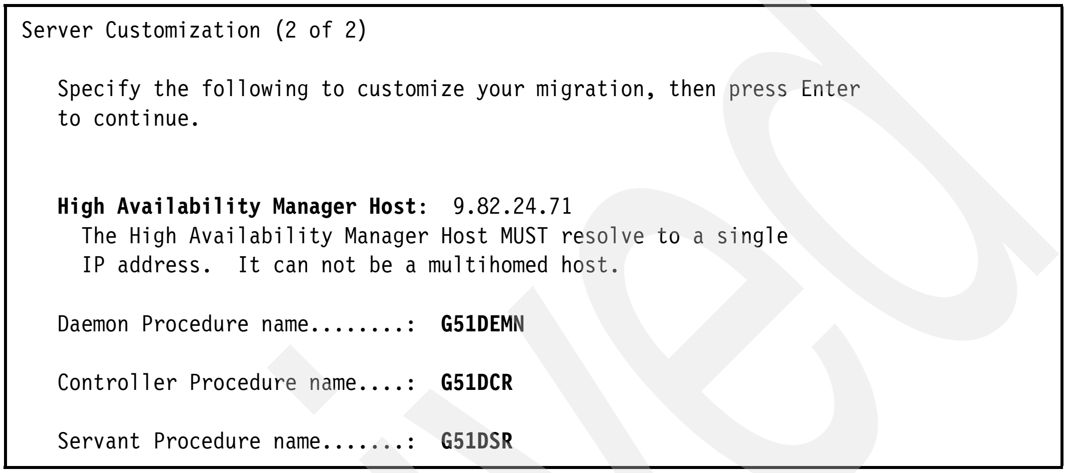
Figure 13-11 Server Customization variables - Screen 1

Note: If you are following along with a migration to V6.1, the screens in Figure 13-11 and Figure 13-12 have changed a bit. See Figure 15-1 on page 327 and Figure 15-2 on page 328 for examples of how these screens look for V6.1.

a. V5.x WebSphere Application Server home directory represents the mount point and node directory for the V5 deployment manager.

b. V6 WebSphere Application Server home directory is the intended node directory for the migrated V6 deployment manager. Notice how the mount point (/wasv6config/g5cell) is not open for update. That value was the HFS mount point provided on the previous panel.

14. The second Server Customization screen is displayed next, as shown in Figure 13-12.



```
Server Customization (2 of 2)

   Specify the following to customize your migration, then press Enter
   to continue.


   High Availability Manager Host:  9.82.24.71
     The High Availability Manager Host MUST resolve to a single
     IP address.  It can not be a multihomed host.

   Daemon Procedure name........:  G51DEMN

   Controller Procedure name....:  G51DCR

   Servant Procedure name.......:  G51DSR
```

Figure 13-12 Server Customization variables - Screen 2

Specify the following values:

a. The High Availability Manager Host is a new function of V6 servers. The field on the panel accepts either a DNS host name or an IP address. If you enter a host name, it must resolve to a single IP address.

   For the SYSC system on which this deployment manager node resided, all our hosts are multi-homed. For our example, we enter the IP address of the SYSC MVS image.

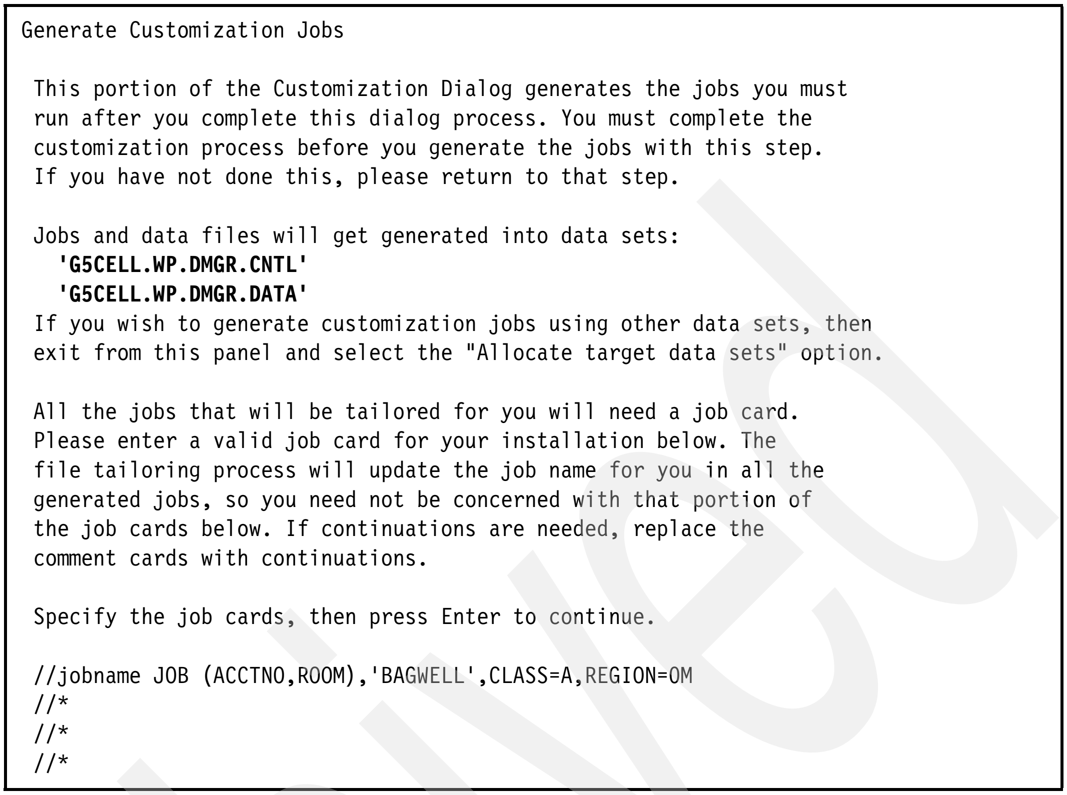
b. New V6 JCL procedures are placed in the CNTL target data set. The BBOMDCP member contains a job that copies them to your specified PROCLIB, specified in Figure 13-8 on page 248, and renames them to whatever values you supply here.

   You have two choices:

   - Specify new names, which requires that you create new STARTED profiles for the controllers. The new profiles are required just for the controllers, but not required for the servant procedures. The instruction member BBOMDINS has an example of the command required to create profiles.
   - Specify the same names as used with the V5 configuration. That eliminates the need for new STARTED profiles, but requires you to back up the V5 procedures since this job would overwrite them.

   For the scenario here, we kept the same names. However, keeping the same names has a cost. See 13.1.4, “Daemon migration problem” on page 273 for a discussion on the repercussions of re-using procedure names.

15. With the variables defined, go to the main panel, shown in Figure 13-4, and select option **3**, Generate customization jobs. The next dialog is shown in Figure 13-13.

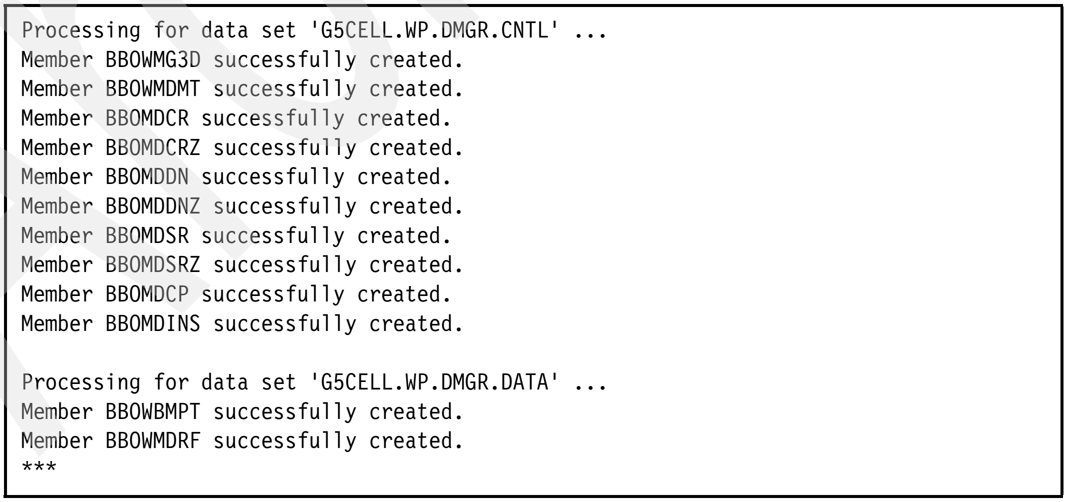


```
Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please return to that step.

 Jobs and data files will get generated into data sets:
   'G5CELL.WP.DMGR.CNTL'
   'G5CELL.WP.DMGR.DATA'
 If you wish to generate customization jobs using other data sets, then
 exit from this panel and select the "Allocate target data sets" option.

 All the jobs that will be tailored for you will need a job card.
 Please enter a valid job card for your installation below. The
 file tailoring process will update the job name for you in all the
 generated jobs, so you need not be concerned with that portion of
 the job cards below. If continuations are needed, replace the
 comment cards with continuations.

 Specify the job cards, then press Enter to continue.

 //jobname JOB (ACCTNO,ROOM),'BAGWELL',CLASS=A,REGION=0M
 //*
 //*
 //*
```

Figure 13-13 Generating the customization jobs

The data set values are filled in automatically based on what you provided for the Target Data Sets in Figure 13-6 on page 247. But you should always check this to make sure it is truly where you want the jobs to be placed.

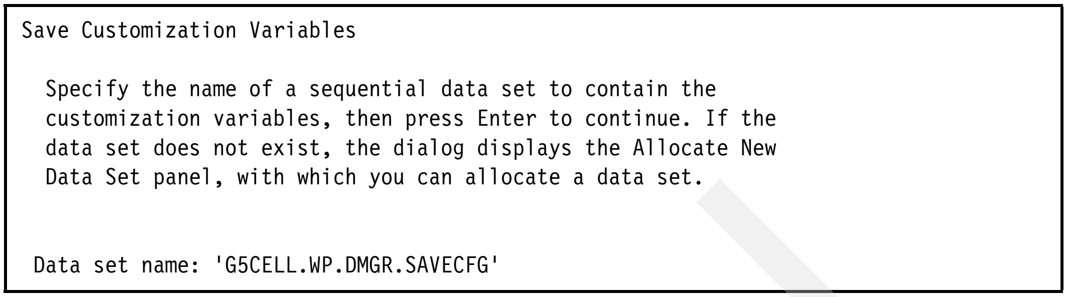
16. Press Enter. The jobs are written out to the target data sets. The output is shown in Figure 13-14.



```
Processing for data set 'G5CELL.WP.DMGR.CNTL' ...
Member BBOWMG3D successfully created.
Member BBOWMDMT successfully created.
Member BBOMDCR successfully created.
Member BBOMDCRZ successfully created.
Member BBOMDDN successfully created.
Member BBOMDDNZ successfully created.
Member BBOMDSR successfully created.
Member BBOMDSRZ successfully created.
Member BBOMDCP successfully created.
Member BBOMDINS successfully created.

Processing for data set 'G5CELL.WP.DMGR.DATA' ...
Member BBOWBMPT successfully created.
Member BBOWMDRF successfully created.
***
```

Figure 13-14 Customized jobs written out to the target data sets

17. Select option **S** and save the customization variables in the screen in Figure 13-4. This results in display of the screen shown in Figure 13-15.



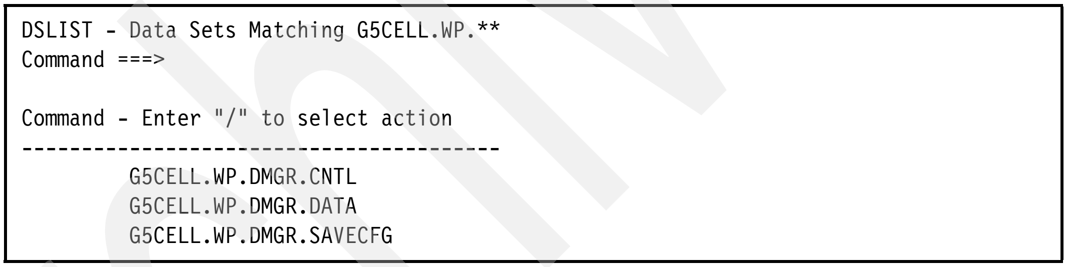
```
Save Customization Variables

  Specify the name of a sequential data set to contain the
  customization variables, then press Enter to continue. If the
  data set does not exist, the dialog displays the Allocate New
  Data Set panel, with which you can allocate a data set.


 Data set name: 'G5CELL.WP.DMGR.SAVECFG'
```

Figure 13-15 Customized jobs written out to the target data sets

We recommend you accept the default allocation parameters.

> **Tip:** When coming up with names for your data sets, keep a few things in mind:
>
> - Have a separate set of migration data sets for each node being migrated.
> - The high-level qualifier should indicate in some fashion the node being migrated.
> - The high-level qualifier for the CNTL, DATA, and SAVECFG data sets for a node should be the same.

18. After exiting the customization dialogs, list the data sets as shown in Figure 13-16.



```
DSLIST - Data Sets Matching G5CELL.WP.**
Command ===>

Command - Enter "/" to select action
----------------------------------------
         G5CELL.WP.DMGR.CNTL
         G5CELL.WP.DMGR.DATA
         G5CELL.WP.DMGR.SAVECFG
```

Figure 13-16 Customized data sets for the migration of the deployment manager node

These are the contents of the data sets:

– CNTL contains the JCL jobs, the new start procedures and an instruction member.

– DATA contains two members with customized scripts.

– SAVECFG contains the variables you supplied during this customization run.

19. Review instruction member BBOMDINS in the CNTL data set.

The BBOMDINS member of the CNTL data set contains a list of instructions for migrating the node. The instructions are organized according to the diagram in Figure 13-17.

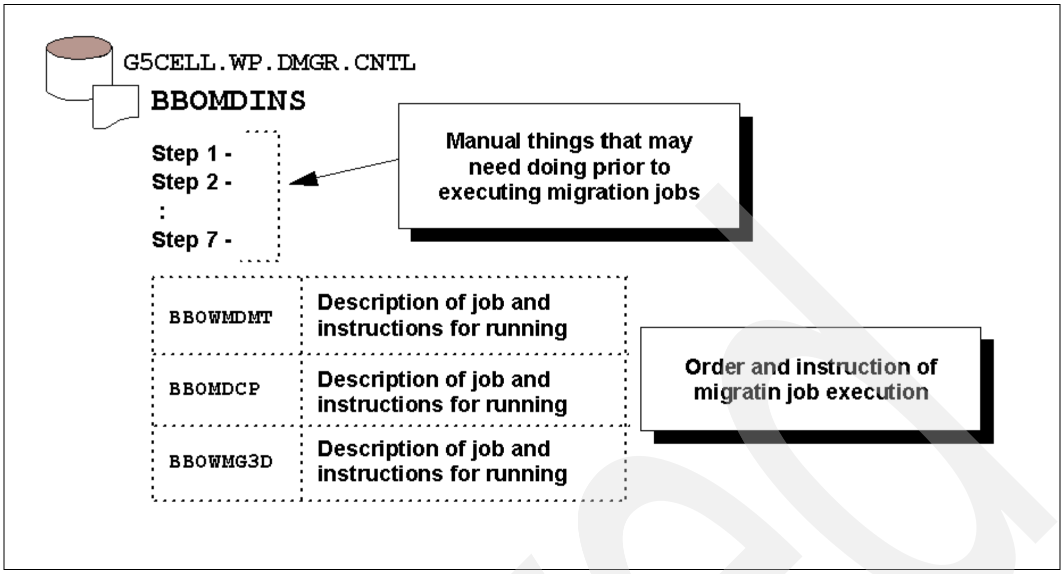

G5CELL.WP.DMGR.CNTL
BBOMDINS
Step 1 -
Step 2 -
:
Step 7 -
Manual things that may need doing prior to executing migration jobs
BBOWMDMT
Description of job and instructions for running
BBOMDCP
Description of job and instructions for running
BBOWMG3D
Description of job and instructions for running
Order and instruction of migratin job execution

Figure 13-17 How the BBOMDINS instruction member information is organized

> **Important:** Review the contents of the BBOMDINS instruction member and follow the instructions. These instructions are specific to your particular system. Do not rely exclusively on our example. In particular, those instructions describe what needs to be done to your MVS system so WebSphere Application Server for z/OS Version 6 can operate correctly. For example, you may need to perform manual steps for APF authorization and HFS auto-mount.

20. Stop the deployment manager. The servers in a node being migrated must be stopped while the node is being migrated.

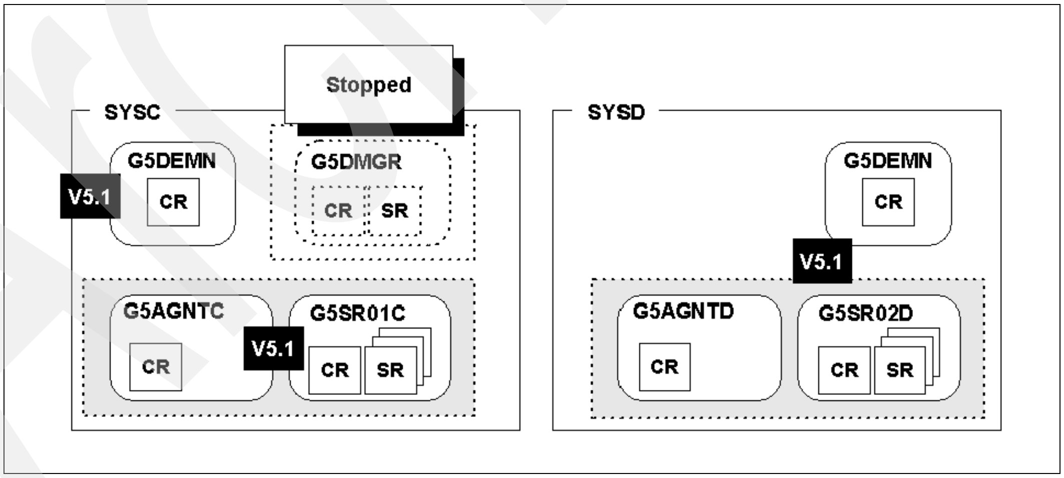


Figure 13-18 Deployment manager stopped; Daemon and other servers left started

Issue a stop command against the deployment manager controller:

```
/P G5DMGR
```

This stops only the deployment manager but not its daemon. If you make the mistake of stopping the daemon, the servers in the G5NODEC node would come down as well. To maximize the up time of those servers, leave the daemon up. Later on, you will restart the daemon, which results in the node agent and G5SR01C server coming down.

21. Run customized jobs.

> **Important:** You can run through the ISPF customization dialogs on any MVS system in the sysplex, but the migration jobs must be run on the same MVS image on which the servers of the migrated node run. For our deployment manager node, that means the jobs are run on the SYSC system.

For the migration of a deployment manager node, three customization jobs are generated:

– BBOWMDMT allocates and mounts the new V6 configuration HFS.

– BBOMDCP copies new V6 JCL start procedures to the PROCLIB.

– BBOWMG3D performs the migrations.

Details about these jobs follow.

a. The BBOWMDMT job

Add a USER= and PASSWORD= value to the JOB card so it runs under a userid that has the authority to mount an HFS file system. Figure 13-19 shows successful completion.

```
$HASP373 BBOWMDMT STARTED - INIT 1    - CLASS A - SYS SYSC
Jobname  Procstep Stepname  CPU Time      EXCPs     RC
BBOWMDMT --None-- ALLOC     00:00:00          0     00
BBOWMDMT --None-- MNTHFS    00:00:00         87     00
```

Figure 13-19 BBOWMDMT job

Figure 13-20 shows that the mount point is not owned by the correct ID. The configuration mount point should be owned by the WebSphere Admin ID and WebSphere Configuration Group for the cell configuration mounted there.

```
BAGWELL:/u/bagwell-> cd /wasv6config
BAGWELL:/wasv6config-> ls -al
drwxrwxr-x  20 G6ADMIN  G6CFG      8192 Mar 19 10:44 .
drwxr-xr-x  34 QWER01   SYS1       8192 Mar 17 07:32 ..
drwxrwxr-x   2 QWER01   GROUP0     8192 Mar 19 10:44 g5cell
```

Figure 13-20 Mount point after BBOWMDMT job

Switch to superuser and from the /wasv6config directory issue a **chown** command against the g5cell mount point directory as shown in Figure 13-21.

```
chown g5admin:g5cfg ./g5cell
BAGWELL:/wasv6config-> ls -al
total 320
drwxrwxr-x  20 G6ADMIN  G6CFG      8192 Mar 19 10:44 .
drwxr-xr-x  34 QWER01   SYS1       8192 Mar 17 07:32 ..
drwxrwxr-x   2 G5ADMIN  G5CFG      8192 Mar 19 10:44 g5cell
```

Figure 13-21 Mount point after correcting owner

You have the alternative to skip BBOWMDMT and manually allocate and mount the HFS data set. Either way works fine. It is entirely your choice whether you run BBOWMDMT and subsequently change the ownership, or you manually allocate the HFS data set.

b. The BBOMDCP job

The BBOMDCP job copies six members to PROCLIB as shown in Figure 13-22.

```
//SYSIN     DD *
 C INDD=INPUT,OUTDD=OUTPUT
 S M=((BBOMDDN,G51DEMN,R))
 S M=((BBOMDDNZ,G51DEMNZ,R))
 S M=((BBOMDCR,G51DCR,R))
 S M=((BBOMDCRZ,G51DCRZ,R))
 S M=((BBOMDSR,G51DSR,R))
 S M=((BBOMDSRZ,G51DSRZ,R))
/*
```

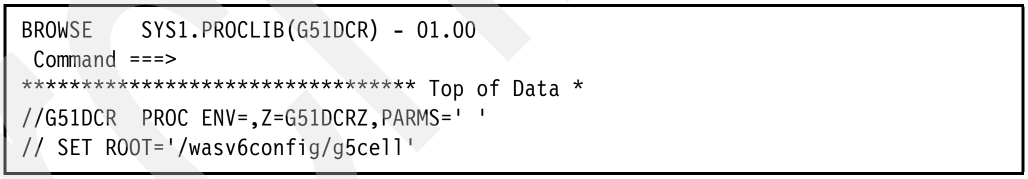
Figure 13-22 BBOMDCP job

Recall that in step 3 on page 245, those same members are backed up from our SYS1.PROCLIB data set. We are using the same procedure names so we can avoid having to create new STARTED profiles for new controller procedure names.

You should try not to re-use V5 procedures with a V6 configuration. You would need to update the SET ROOT= and any STEPLIBS. It is better to simply allow the migration utility to create new procedures and copy them into PROCLIB.

Add a USER= and PASSWORD= value to the JOB card so it runs under a userid that has the authority to copy members into PROCLIB.

Figure 13-23 shows that the new procedures have the V6 mount point for SET ROOT=.



```
BROWSE    SYS1.PROCLIB(G51DCR) - 01.00
 Command ===>
********************************* Top of Data *
//G51DCR  PROC ENV=,Z=G51DCRZ,PARMS=' '
// SET ROOT='/wasv6config/g5cell'
```

Figure 13-23 Verify copy members

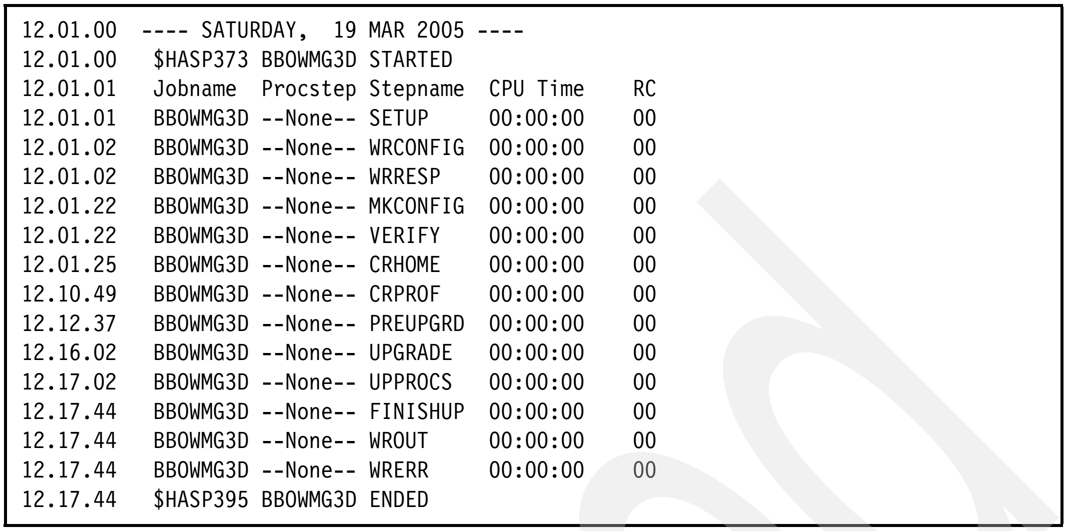
c. The BBOWMG3D job

Be sure the deployment manager is stopped.

For this example, we leave the daemon associated with the deployment manager up while BBOWMG3D runs. However, it is stopped right after the BBOWMG3D job completes. The servers in the G5NODEC node still rely on this daemon. To maximize up time during the migration job, leave it up. But before the newly migrated deployment manager is re-started, you should stop and restart the daemon.

Be sure the JOB card had a USER= and PASSWORD= so this job runs under the ID of the WebSphere Admin ID of G5ADMIN. This job must run under that ID.

Our example runs with all steps successful in sixteen minutes. The results are shown in Figure 13-24.



```
12.01.00  ---- SATURDAY,  19 MAR 2005 ----
12.01.00   $HASP373 BBOWMG3D STARTED
12.01.01   Jobname  Procstep Stepname  CPU Time    RC
12.01.01   BBOWMG3D --None-- SETUP     00:00:00    00
12.01.02   BBOWMG3D --None-- WRCONFIG  00:00:00    00
12.01.02   BBOWMG3D --None-- WRRESP    00:00:00    00
12.01.22   BBOWMG3D --None-- MKCONFIG  00:00:00    00
12.01.22   BBOWMG3D --None-- VERIFY    00:00:00    00
12.01.25   BBOWMG3D --None-- CRHOME    00:00:00    00
12.10.49   BBOWMG3D --None-- CRPROF    00:00:00    00
12.12.37   BBOWMG3D --None-- PREUPGRD  00:00:00    00
12.16.02   BBOWMG3D --None-- UPGRADE   00:00:00    00
12.17.02   BBOWMG3D --None-- UPPROCS   00:00:00    00
12.17.44   BBOWMG3D --None-- FINISHUP  00:00:00    00
12.17.44   BBOWMG3D --None-- WROUT     00:00:00    00
12.17.44   BBOWMG3D --None-- WRERR     00:00:00    00
12.17.44   $HASP395 BBOWMG3D ENDED
```

Figure 13-24   BBOWMG3D completion

22. Perform post-migration RACF work.

    The BBOMDINS instruction member contains two sets of post-migration RACF work that may be needed:

    a. Create new STARTED profiles.

       If you provided new JCL start procedure names, then you must create new STARTED profiles. In our example, we backed up our V5 JCL procedures and copied the V6 procedures with the same names, so no new STARTED profiles are needed.

       The BBOMDINS instruction member has an example of the command needed.

    b. Create a new keyring for the deployment manager servant ID.

       One of the instructions found under the BBOWMG3D job says the following:

       > *WebSphere for z/OS V6 requires that the deployment manager servant has a WASKeyring with the WebSphere CA certificate. If your WebSphere V5 deployment manager uses a different userid for the deployment manager servant than it does for the controller you must update your RACF configuration.*

       For our example, the deployment manager controller runs under a different userid than the servant. The deployment manager controller runs as G5DCRU while the servant runs as G5DSRU. Therefore, it is necessary to issue RACF commands to provide the G5DSRU ID a keyring and connect the WebSphere certificate to it.

       First submit the following RACF command to add a keyring to the G5DSRU ID:

       ```
       RACDCERT ADDRING(WASKeyring) ID(G5DSRU)
       ```

       Then submit the following RACF command to connect the Certificate Authority's certificate to the new keyring just created (command entered as one line):

       ```
       RACDCERT ID(G5DSRU) CONNECT(RING(WASKeyring)LABEL('WebSphereCA') CERTAUTH)
       ```

23. Stop the daemon on SYSC

    Before restarting the migrated deployment manager, first stop the daemon server. Recall we had left the daemon running to maximize the up time of the application servers on SYSC. During the migration of the deployment manager, the daemon on SYSC is migrated as well.

Issue a **stop** command against the SYSC daemon:

```
/P G5DEMN
```

This results in all the servers on SYSC stopping. The servers on SYSD remain up.

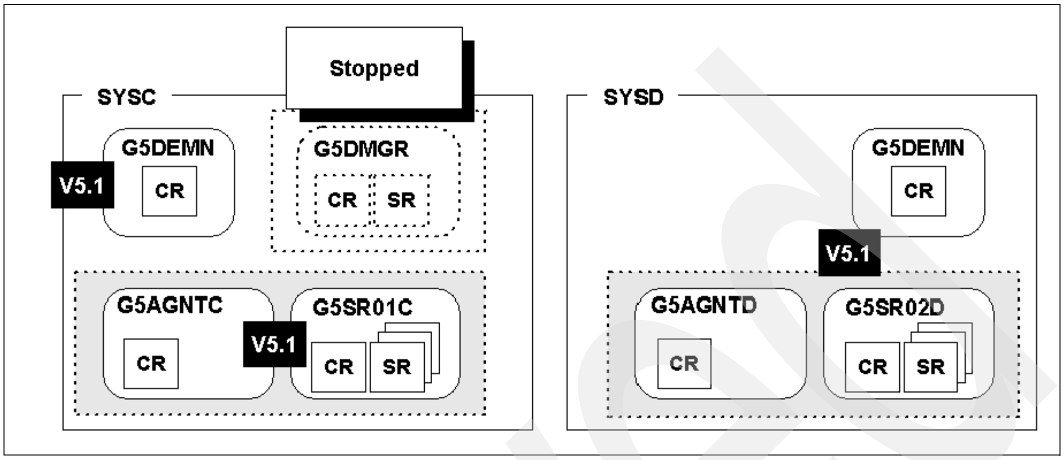


Figure 13-25 Daemon server on SYSC stopped; all servers for cell on SYSC stopped as well

Note: Any servers for other cells on SYSC would be unaffected by stopping the daemon server. It is only servers for that cell on the same MVS image as the daemon that are affected when a daemon is stopped.

24. Start the deployment manager with this start command:

```
S G51DCR,ENV=G5CELL.G5NODE.G5DMGR,JOBNAME=G5DMGR
```

This is the same start command syntax that you would use to start the V5.1 deployment manager.

The daemon server is started automatically by the deployment manager.

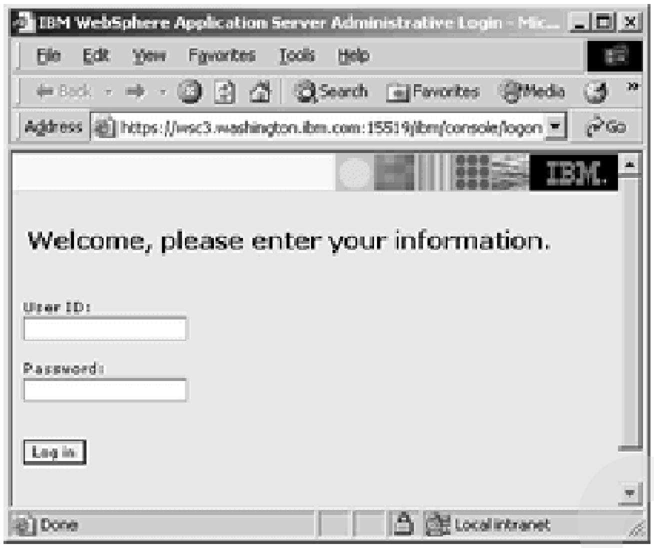
Check the servant region's SYSPRINT for this message:

```
WSVR0001I: Server SERVANT PROCESS dmgr open for e-business
```

Verify you can access the HTTP port of the newly-migrated deployment manager administrative console. This is the address for our example:

```
http://wsc3.washington.ibm.com:15518/admin
```

Figure 13-26 shows the resultant browser screen. Since there are two fields showing for user and password, this shows that global security is enabled. The deployment manager before the migration also had global security enabled, so this is a consistent sign that the migration has worked successfully.


IBM WebSphere Application Server Administrative Login - Mic...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
IBM.
Log in
Done
Local intranet

https://wsc3.washington.ibm.com:15519/ibm/console/logon

Welcome, please enter your information.

User ID:

Password:

Figure 13-26 Deployment manager administrative console

25. Start servers in G5NODEC on SYSC.

    Start the node agent:

    ```
    S G51ACR,ENV=G5CELL.G5NODEC.G5AGNTC,JOBNAME=G5AGNTC
    ```

    Start the G5SR01C server:

    ```
    S G51ACR,ENV=G5CELL.G5NODEC.G5SR01C,JOBNAME=G5SR01C
    ```

    G5NODEC is still a V5.1 node running with a V6 deployment manager. Notice the status of the cell, which is shown in Figure 13-27.

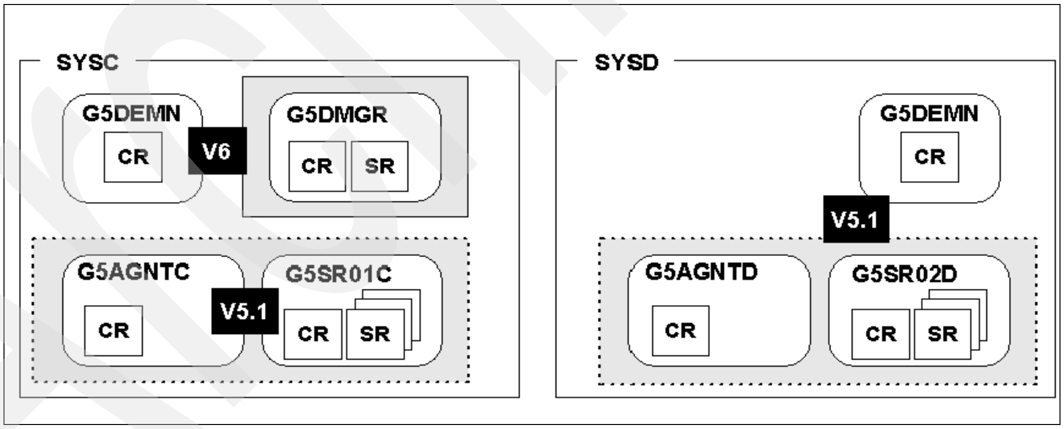
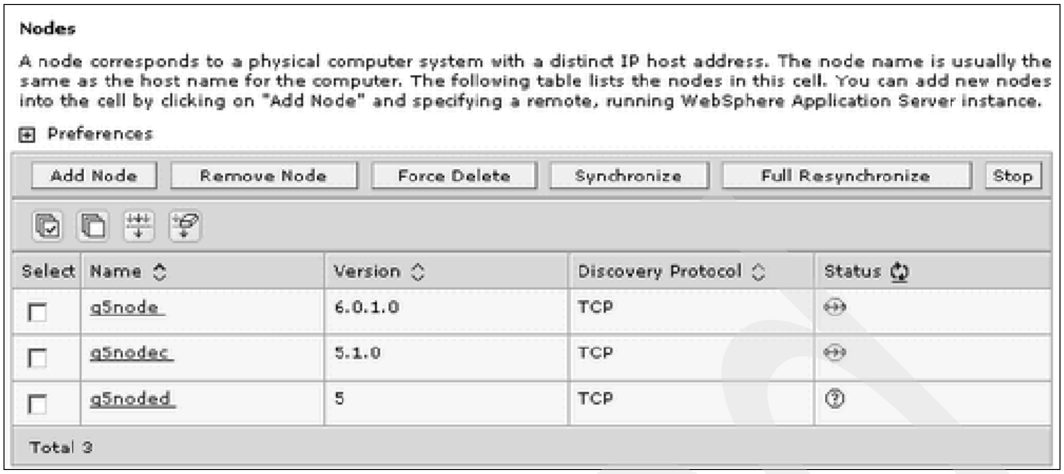


Figure 13-27 Partially migrated cell

Figure 13-28 shows the administrative console view of the nodes.



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

⊞ Preferences

Select	Name	Version	Discovery Protocol	Status
□	g5node	6.0.1.0	TCP	↔
□	g5nodec	5.1.0	TCP	↔
□	g5noded	5	TCP	?

Total 3

Figure 13-28 Node status in administrative console

The V5 copy of the deployment manager is now in a “disabled” state. You would not be able to start it if you tried. You would get a message indicating that the serverindex.xml file is not found. However, if you really need to restart the V5 image, you can re-enable the V5 deployment manager. See 13.1.6, “Reverting a node back to V5” on page 274 for details on how to re-enable the V5 deployment manager.

13.1.2 Migrating the application server node on SYSD

The application server node on SYSD is migrated next. We could just as easily migrate SYSC next, but, because our nodes are at V5.1, we want to illustrate having a node coexist with a V6 daemon on the same MVS image.

Had our nodes been at the V5.0 level, then G5NODEC would be the next node to migrate. A server at the V5.0 level could not have been started when the daemon server that supported it on SYSC was now at V6. The alternative, migrating SYSC next, would require shutting down any V5.0 servers on SYSC.

In part, our decision was based on the dilemma we faced due to using shared procedures and our decision to re-use the same procedure names for the migrated nodes. It didn't create a hard requirement to migrate SYSD next, but it was enough of a nagging worry that we went ahead with the SYSD migration next. For more on this dilemma, see 13.1.4, “Daemon migration problem” on page 273.

1. Perform preliminary work.
   a. The inventory of the node on SYSD was taken when the deployment manager was migrated. It is not necessary to do it again.
   b. The HFS does not need to be backed up since this cell is contained within a single shared HFS.
   c. Back up JCL start procedures used by the application server node for these JCL members:
      - G51ACR
      - G51ACRZ
      - G51ASR
      - G51ASRZ

We back up the JCL because we intend to use the same names for the newly created V6 JCL start procedures in order to reduce the amount of post-migration work required.

d. The daemon JCL need not be backed up since it was already backed up. Recall that we are using a common set of JCL for all daemon instances.

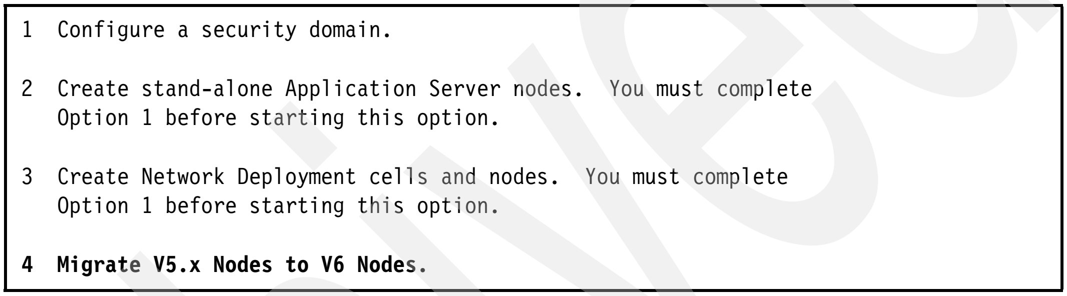
e. Be sure the G5CELL Admin ID can work in the /tmp directory. This is simply a matter of checking to see that the /tmp/migrate directory is still owned by the G5ADMIN ID, and the bbomigrt2.sh file was also owned by G5ADMIN. The simplest fix is to delete the /tmp/migrate directory so this migration will have a clear path to execute.

2. Invoke the V6 ISPF customization dialogs:

```
'WAS600.WAS.SBBOCLIB(BBOWSTRT)' 'APPL(MIG)'
```

The `'APPL(MIG)'` provides a set of ISPF variables for this migration run that are related to the ones used for the deployment manager.

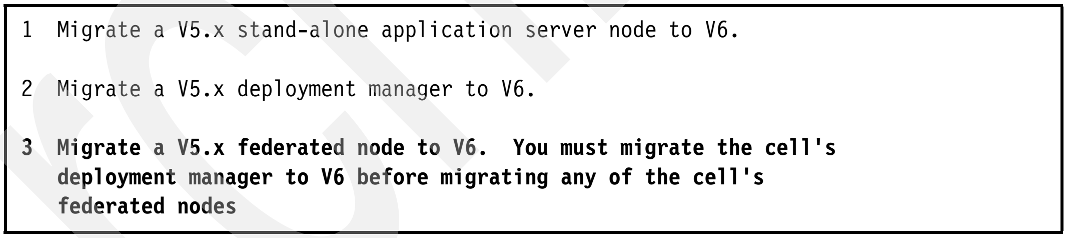
3. Select option **4** from the primary option dialog shown in Figure 13-29.



```
1  Configure a security domain.

2  Create stand-alone Application Server nodes.  You must complete
   Option 1 before starting this option.

3  Create Network Deployment cells and nodes.  You must complete
   Option 1 before starting this option.

4  Migrate V5.x Nodes to V6 Nodes.
```

Figure 13-29 Option 4 - Invoke migration customization dialogs

4. Select option **3**, Migrate a V5.x federated node to V6, as shown in Figure 13-30.



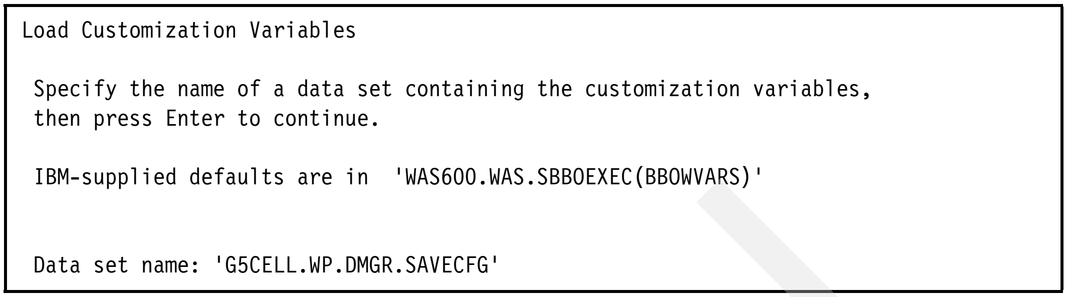
```
1  Migrate a V5.x stand-alone application server node to V6.

2  Migrate a V5.x deployment manager to V6.

3  Migrate a V5.x federated node to V6.  You must migrate the cell's
   deployment manager to V6 before migrating any of the cell's
   federated nodes
```

Figure 13-30 Option 3 -- migrate a federated node

5. The dialog shown in Figure 13-4 on page 246 is displayed. Follow the same steps as you did for the deployment manager.

6. Load the deployment manager saved customization variables (the SAVECFG member in the G5CELL.WP.DMGR.* data set) by selecting option **L**. Using the deployment manager variables is a handy way to avoid having to re-specify variables common to all migrations. This is a shortcut, but you could easily just re-specify the common values.

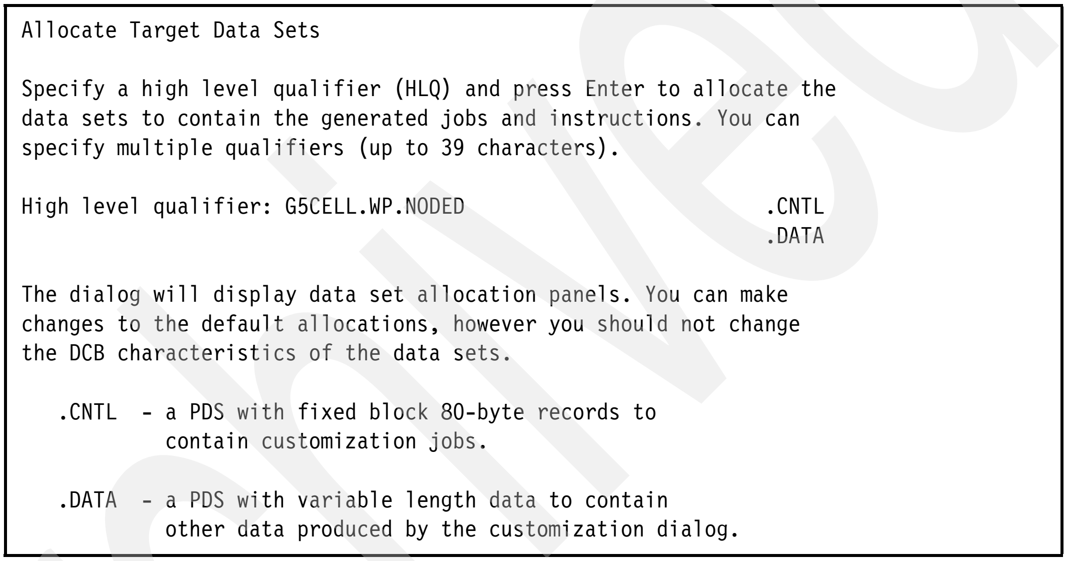
7. The dialog shown in Figure 13-31 is displayed.



```
Load Customization Variables

 Specify the name of a data set containing the customization variables,
 then press Enter to continue.

 IBM-supplied defaults are in  'WAS600.WAS.SBBOEXEC(BBOWVARS)'


 Data set name: 'G5CELL.WP.DMGR.SAVECFG'
```

Figure 13-31 Deployment manager customization variables loaded into SYSD migration

8. Allocate the target data sets where the customized jobs are stored, as shown in Figure 13-32. These are different from the ones created for the deployment manager.

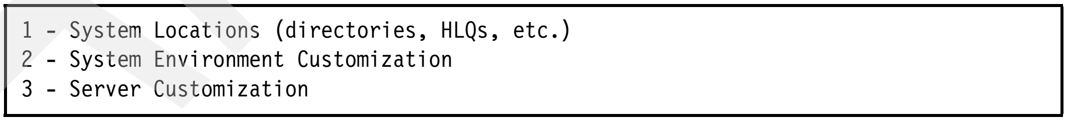


```
Allocate Target Data Sets

Specify a high level qualifier (HLQ) and press Enter to allocate the
data sets to contain the generated jobs and instructions. You can
specify multiple qualifiers (up to 39 characters).

High level qualifier: G5CELL.WP.NODED                         .CNTL
                                                              .DATA

The dialog will display data set allocation panels. You can make
changes to the default allocations, however you should not change
the DCB characteristics of the data sets.

   .CNTL  - a PDS with fixed block 80-byte records to
            contain customization jobs.

   .DATA  - a PDS with variable length data to contain
            other data produced by the customization dialog.
```

Figure 13-32 Allocated target data sets

We suggest using the default allocation parameters. This step results in the creation of two data sets. G5CELL.WP.NODED.CNTL is a FB 80 PDS where the JCL was stored. G5CELL.WP.NODED.DATA is a VB 255 PDS where scripts are stored.

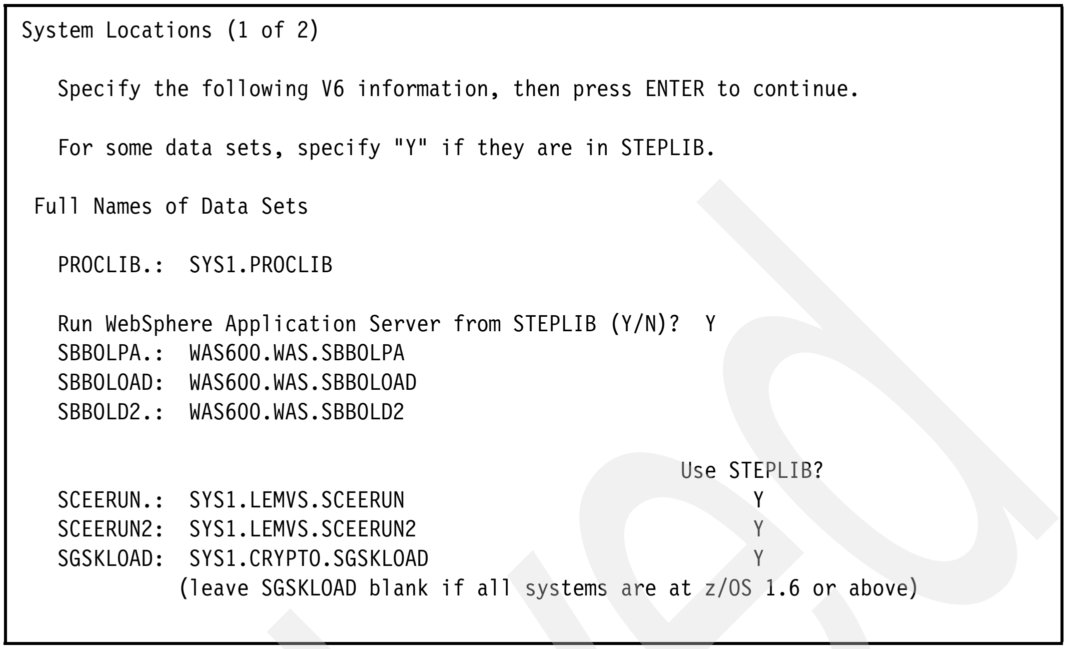
9. Select option **2**, Define Variables, which results in the dialog shown in Figure 13-33.



```
1 - System Locations (directories, HLQs, etc.)
2 - System Environment Customization
3 - Server Customization
```

Figure 13-33 Defining variables

10. Select **1**, which leads to the dialog shown in Figure 13-34.

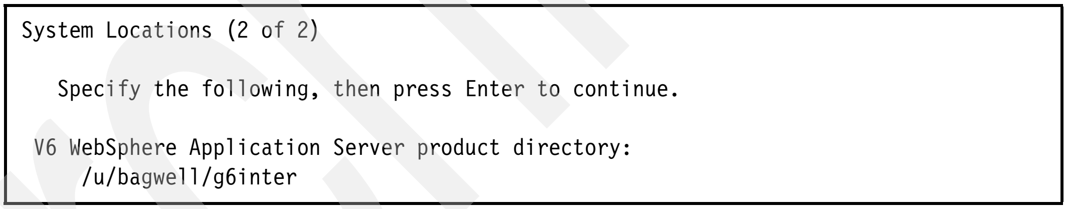


```
System Locations (1 of 2)

   Specify the following V6 information, then press ENTER to continue.

   For some data sets, specify "Y" if they are in STEPLIB.

 Full Names of Data Sets

   PROCLIB.:  SYS1.PROCLIB

   Run WebSphere Application Server from STEPLIB (Y/N)?  Y
   SBBOLPA.:  WAS600.WAS.SBBOLPA
   SBBOLOAD:  WAS600.WAS.SBBOLOAD
   SBBOLD2.:  WAS600.WAS.SBBOLD2


                                                       Use STEPLIB?
   SCEERUN.:  SYS1.LEMVS.SCEERUN                             Y
   SCEERUN2:  SYS1.LEMVS.SCEERUN2                            Y
   SGSKLOAD:  SYS1.CRYPTO.SGSKLOAD                           Y
             (leave SGSKLOAD blank if all systems are at z/OS 1.6 or above)
```

Figure 13-34 Variables supplied for System Locations 1

The variables for the first System Locations screen are the same as those used for the deployment manager. You don't need to make any changes to this screen.

11.The next screen, shown in Figure 13-35, shows more variables to specify. These are also the same as for the deployment manager.

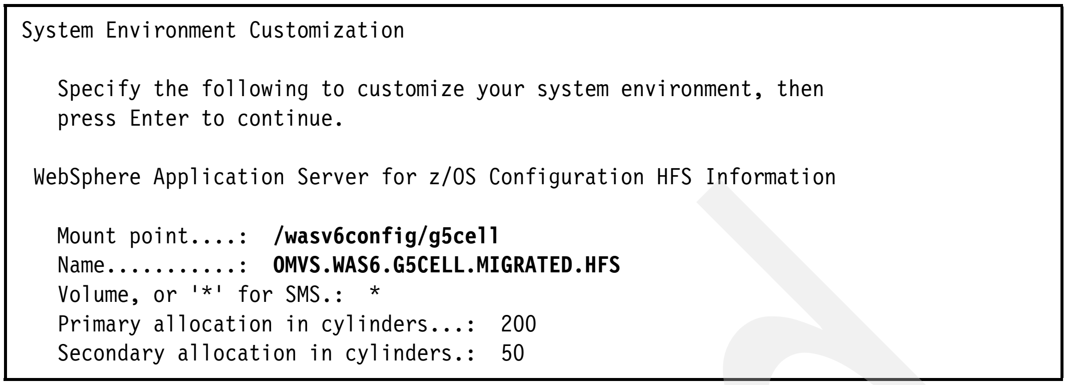


```
System Locations (2 of 2)

   Specify the following, then press Enter to continue.

 V6 WebSphere Application Server product directory:
     /u/bagwell/g6inter
```

Figure 13-35 Variables supplied for System Locations 2

Here you would specify the directory mount point of the V6 SMP/E HFS, where WAS600.WAS.SBBOHFS is mounted. Our value of /u/bagwell/g6inter represents an intermediate symbolic link we created that points to the real V6 product directory.

12.The System Environment Customization variables shown in Figure 13-36 are the same as provided for the migration of the deployment manager.

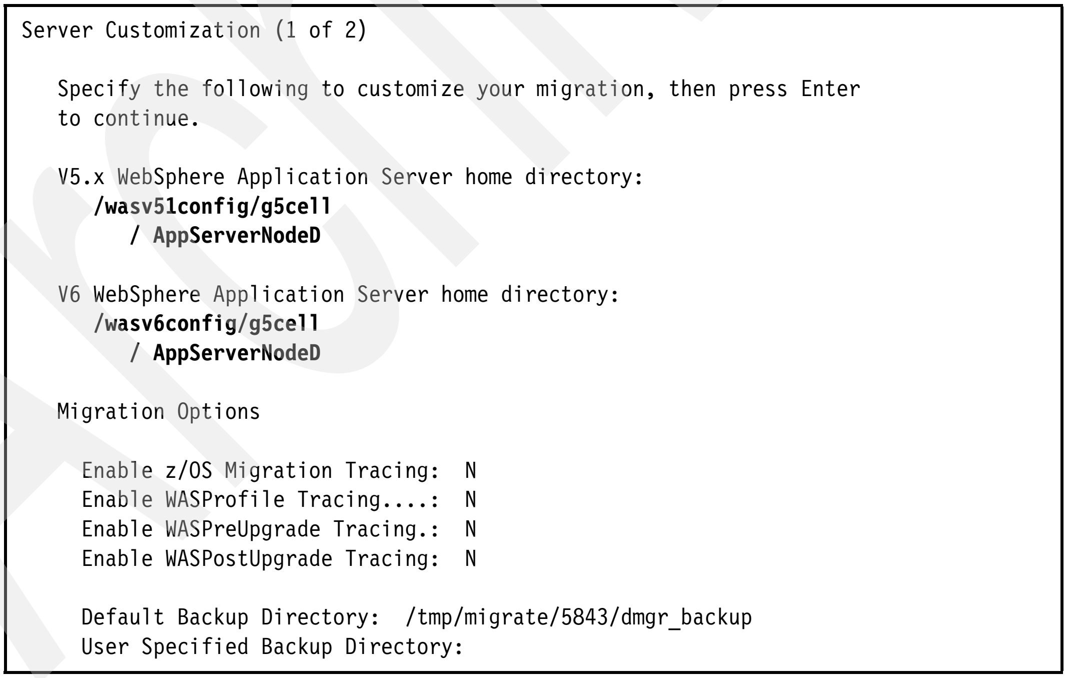


```
System Environment Customization

   Specify the following to customize your system environment, then
   press Enter to continue.

 WebSphere Application Server for z/OS Configuration HFS Information

   Mount point....:  /wasv6config/g5cell
   Name...........:  OMVS.WAS6.G5CELL.MIGRATED.HFS
   Volume, or '*' for SMS.:  *
   Primary allocation in cylinders...:  200
   Secondary allocation in cylinders.:  50
```

Figure 13-36 System Environment Customization variables: HFS file system information

> **Note:** Because the shared HFS is used for all the nodes in the cell, you must supply the mount point values. The variable must be set for the migration to run. However, you do not need to run BBOWMMMT because the HFS is already mounted.
>
> If you have this node in a different HFS, then you would specify that different mount point, and you would need to run BBOWMMMT.

13.To this point, you have been able to use the default variable values that were specified for the deployment manager. On the next dialog, shown in Figure 13-37, you must specify non-default values.



```
Server Customization (1 of 2)

   Specify the following to customize your migration, then press Enter
   to continue.

   V5.x WebSphere Application Server home directory:
      /wasv51config/g5cell
         / AppServerNodeD

   V6 WebSphere Application Server home directory:
      /wasv6config/g5cell
         / AppServerNodeD

   Migration Options

     Enable z/OS Migration Tracing:  N
     Enable WASProfile Tracing....:  N
     Enable WASPreUpgrade Tracing.:  N
     Enable WASPostUpgrade Tracing:  N

     Default Backup Directory:  /tmp/migrate/5843/dmgr_backup
     User Specified Backup Directory:
```

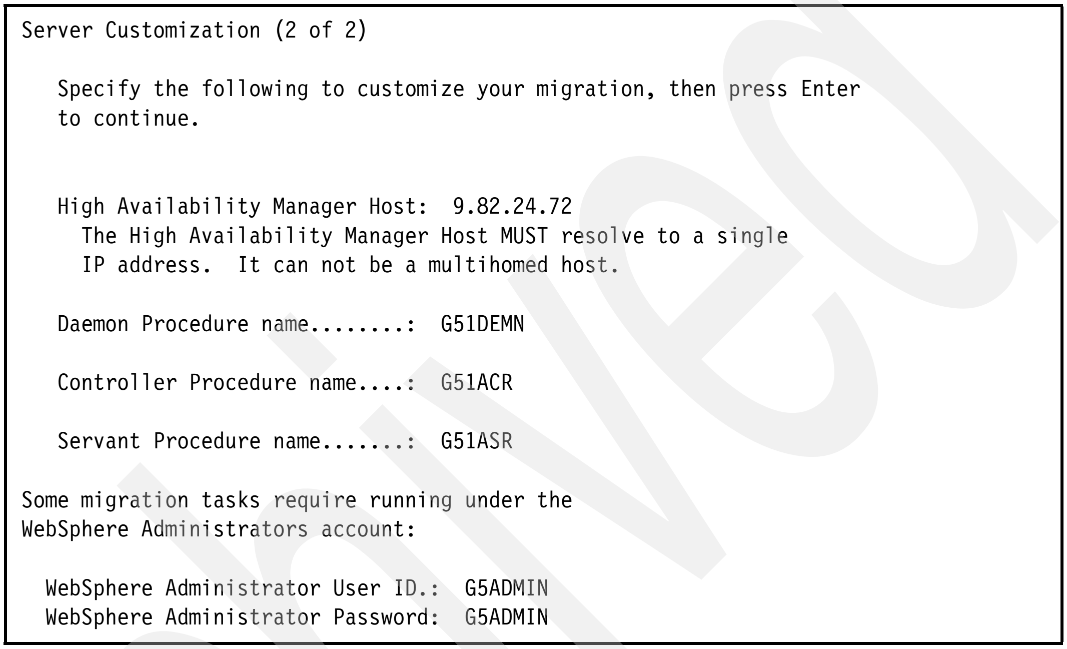
Figure 13-37 Server Customization variables: Screen 1

> **Note:** If you are following along with a migration to V6.1, the screens in Figure 13-37 and Figure 13-38 have changed a bit. See Figure 15-1 on page 327 and Figure 15-2 on page 328 for examples of how these screens look for V6.1

You must specify the source mount point and node directory for the V5 application server node. In this case, the mount point is /was51config/g5cell and the node is AppServerNodeD.

You also specify the target directory for the migration. The mount point /wasv6config/g5cell is the same value as that specified in Figure 13-10 on page 249. The mount point is not open for update. The node value for the target matches the node value for the source.

14. The second Server Customization screen is shown in Figure 13-38.



```
Server Customization (2 of 2)

   Specify the following to customize your migration, then press Enter
   to continue.


   High Availability Manager Host:  9.82.24.72
     The High Availability Manager Host MUST resolve to a single
     IP address.  It can not be a multihomed host.

   Daemon Procedure name........:  G51DEMN

   Controller Procedure name....:  G51ACR

   Servant Procedure name.......:  G51ASR

Some migration tasks require running under the
WebSphere Administrators account:

  WebSphere Administrator User ID.:  G5ADMIN
  WebSphere Administrator Password:  G5ADMIN
```

Figure 13-38 Server Customization variables: Screen 2

The High Availability Manager Host is a new function of V6 servers. You can enter either a DNS host name or an IP address. If you enter a host name, it must resolve to a single IP address.

In our example, all our hosts are multi-homed. Therefore, we specify the IP address of SYSD MVS image.

> **Note:** Make sure the IP address is unique within the cell. If you specify the wrong address, the server will not start.

The migration creates new V6 JCL procedures and places them in the CNTL target data set. The BBOMMCP member contains a job that copies these procedures to your specified PROCLIB and renames them to whatever values you supply here.

The procedure names for the application server nodes are different from the deployment manager node. In our case the procedure name is G51ACR. Since the name is different, you must run the BBOMMCP job.

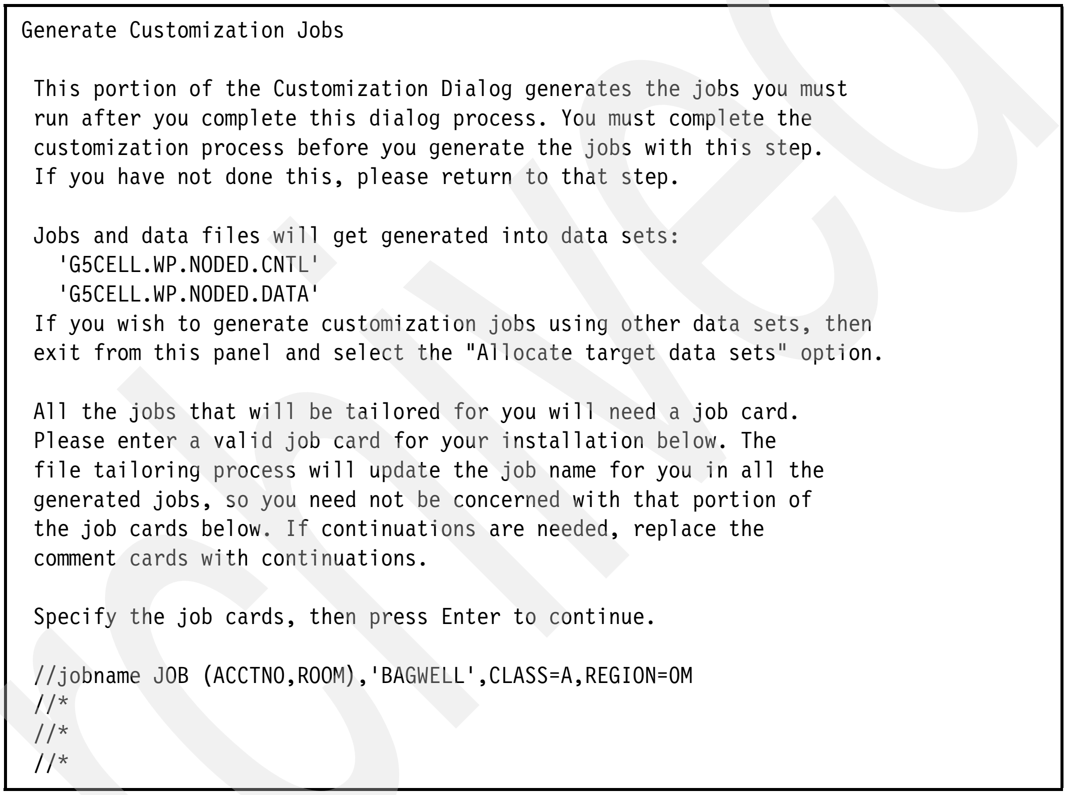
Just as with the deployment manager, we reused the same procedure names across the migration. See 13.1.4, “Daemon migration problem” on page 273 for more details on how to handle this case.

Note that the daemon procedure name of G51DEMN is the same name we used for the deployment manager. (All our daemon instances use the same procedure name.)

Specifying the same name here means that the V6 procedure is copied into SYS1.PROCLIB again. This is not a problem, but it is redundant. If you are concerned about the redundancy, simply remove the reference to G51DEMN and G51DEMNZ from the BBOMMCP job. That prevents the members from being copied a second time.

You must specify the WebSphere Admin ID and password here. This is only really used if global security is enabled, but the panels require the fields to be completed. This information is used during the migration process so the migration process can connect to the deployment manager and invoke a node synchronization. If global security is enabled, it needs to be able to authenticate itself properly to the deployment manager.

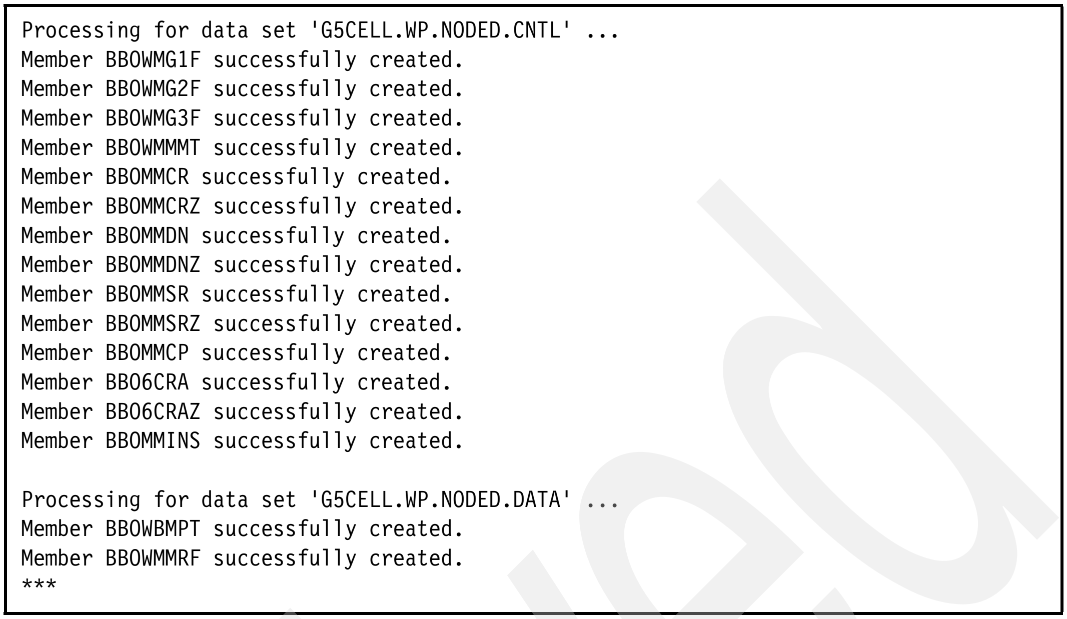
15. Select option **3** from the screen in Figure 13-4 on page 246, which results in display of the Generate Customization Jobs panel shown in Figure 13-39.



```
Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please return to that step.

 Jobs and data files will get generated into data sets:
   'G5CELL.WP.NODED.CNTL'
   'G5CELL.WP.NODED.DATA'
 If you wish to generate customization jobs using other data sets, then
 exit from this panel and select the "Allocate target data sets" option.

 All the jobs that will be tailored for you will need a job card.
 Please enter a valid job card for your installation below. The
 file tailoring process will update the job name for you in all the
 generated jobs, so you need not be concerned with that portion of
 the job cards below. If continuations are needed, replace the
 comment cards with continuations.

 Specify the job cards, then press Enter to continue.

 //jobname JOB (ACCTNO,ROOM),'BAGWELL',CLASS=A,REGION=0M
 //*
 //*
 //*
```

Figure 13-39 Generating the customization jobs

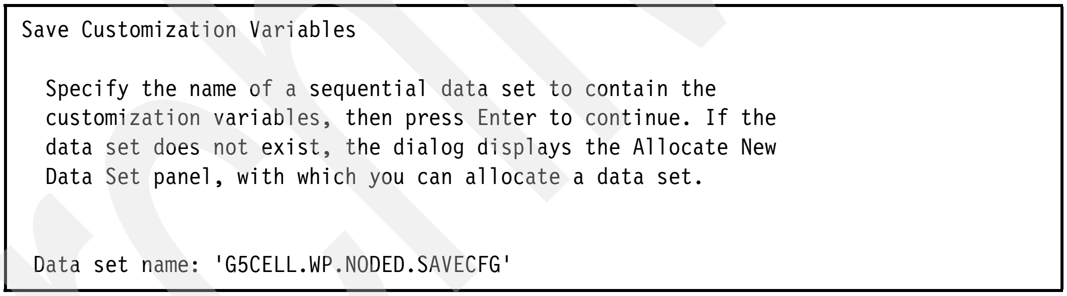
16. Press Enter. The jobs are written to the target data sets as shown in Figure 13-40.



```
Processing for data set 'G5CELL.WP.NODED.CNTL' ...
Member BBOWMG1F successfully created.
Member BBOWMG2F successfully created.
Member BBOWMG3F successfully created.
Member BBOWMMMT successfully created.
Member BBOMMCR successfully created.
Member BBOMMCRZ successfully created.
Member BBOMMDN successfully created.
Member BBOMMDNZ successfully created.
Member BBOMMSR successfully created.
Member BBOMMSRZ successfully created.
Member BBOMMCP successfully created.
Member BBO6CRA successfully created.
Member BBO6CRAZ successfully created.
Member BBOMMINS successfully created.

Processing for data set 'G5CELL.WP.NODED.DATA' ...
Member BBOWBMPT successfully created.
Member BBOWMMRF successfully created.
***
```

Figure 13-40 Customized jobs written out to the target data sets

17. Select option **S** in Figure 13-4 on page 246; the screen shown in Figure 13-41 is displayed. Save the customization variables to a different SAVECFG data set than used by the deployment manager. Accept the default allocation parameters.



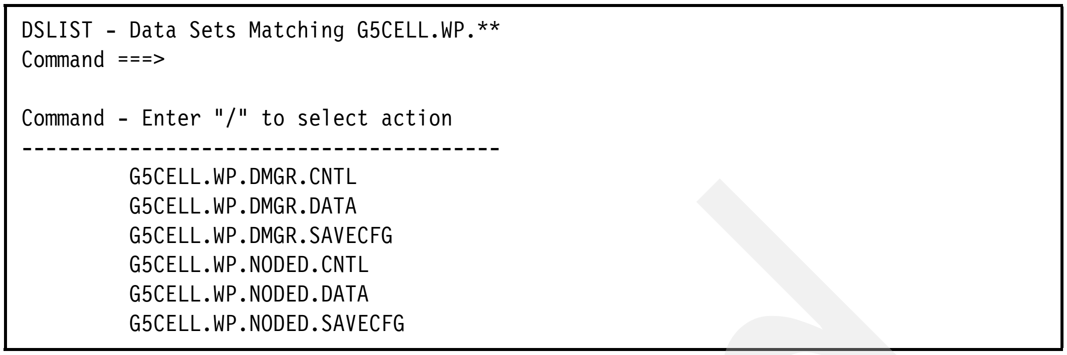
```
Save Customization Variables

  Specify the name of a sequential data set to contain the
  customization variables, then press Enter to continue. If the
  data set does not exist, the dialog displays the Allocate New
  Data Set panel, with which you can allocate a data set.


 Data set name: 'G5CELL.WP.NODED.SAVECFG'
```

Figure 13-41 Saving customization variables

When coming up with names for your data sets, keep a few things in mind:

- Have a separate set of migration data sets for each node being migrated.
- The high-level qualifier should indicate in some fashion the node being migrated.
- The high-level qualifier for the CNTL, DATA and SAVECFG data sets for a node should be the same.

18. Exit the customization dialogs and check the new data sets with a data set list command. Figure 13-42 shows new SYSD node data sets and the deployment manager node data sets.



```
DSLIST - Data Sets Matching G5CELL.WP.**
Command ===>

Command - Enter "/" to select action
----------------------------------------
         G5CELL.WP.DMGR.CNTL
         G5CELL.WP.DMGR.DATA
         G5CELL.WP.DMGR.SAVECFG
         G5CELL.WP.NODED.CNTL
         G5CELL.WP.NODED.DATA
         G5CELL.WP.NODED.SAVECFG
```

Figure 13-42 Customized data sets for the migration of the SYSD application server node

19. Review instruction member BBOMMINS in the CNTL data set.

    The BBOMMINS member of the CNTL data set contains a list of instructions for migrating the node. The instructions are organized according to the diagram in Figure 13-43.

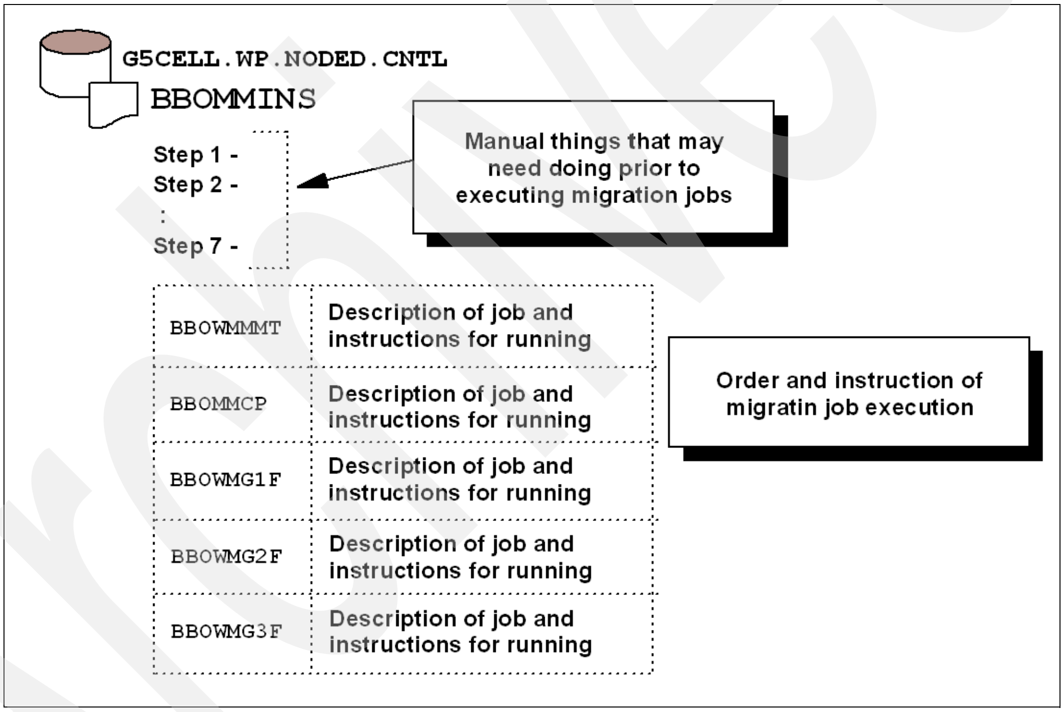

G5CELL.WP.NODED.CNTL
BBOMMINS
Step 1 -
Step 2 -
:
Step 7 -
Manual things that may need doing prior to executing migration jobs
BBOWMMMT
Description of job and instructions for running
BBOMMCP
Description of job and instructions for running
BBOWMG1F
Description of job and instructions for running
BBOWMG2F
Description of job and instructions for running
BBOWMG3F
Description of job and instructions for running
Order and instruction of migratin job execution

Figure 13-43 How the BBOMMINS instruction member information is organized

20. Make sure the deployment manager is up and running. This is a very important prerequisite to migrating a federated node. During migration, the migration utility establishes a connection to the SOAP port of the deployment manager and initiates a node synchronization process. The deployment manager has to be running or the migration process fails.

21. Shut down the daemon server and all G5NODED servers.

    On the SYSD system issue the following command:

    ```
    /P G5DEMN
    ```

    This results in the daemon coming down and all the servers in the G5NODED node coming down as well. The servers in a node must be down to migrate the node. Be very careful to make sure you issue the shutdown command against the correct system. The

JOBNAME value for daemon servers in a Network Deployment configuration cell all have the same value. In our example, the name is G5DEMN. If you run that command against the wrong system, you will bring down the deployment manager and the servers as well.

22. Run customized jobs on the SYSD system for application server node G5NODED.

> Important: You can run through the ISPF customization dialogs on any MVS system in the Sysplex, but the migration jobs must be run on the same MVS image on which the servers of the migrated node will run.

For the migration of a federated node, five customization jobs are generated:

a. BBOWMMMT allocates and mounts the new V6 configuration HFS. You do not need to run this job if you ran the BBOWMDMT job for the deployment manager node. If you are hosting this node in a separate HFS then you need to run this job.

b. BBOMMCP copies new start procedures to the PROCLIB.

```
//SYSIN     DD *
 C INDD=INPUT,OUTDD=OUTPUT
 S M=((BBOMMDN,G51DEMN,R))
 S M=((BBOMMDNZ,G51DEMNZ,R))
 S M=((BBOMMCR,G51ACR,R))
 S M=((BBOMMCRZ,G51ACRZ,R))
 S M=((BBOMMSR,G51ASR,R))
 S M=((BBOMMSRZ,G51ASRZ,R))
/*
```

Recall that under “Preliminary work” we backed up those same members from our SYS1.PROCLIB data set. We are using the same procedure names so we can avoid having to create new STARTED profiles for new controller procedure names. It is better to simply allow the migration utility to create new procedures and copy them into PROCLIB.

The G51DEMN and G51DEMNZ members are copied into PROCLIB by the BBOMDCP job for the deployment manager; therefore copying them again is redundant but causes no harm.

To run the job, add a USER and PASSWORD value to the JOB card so it runs under a userid that has authority to copy members into PROCLIB.

After a successful completion, verify that the new procedures have the V6 mount point configured.

```
BROWSE    SYS1.PROCLIB(G51ACR) - 01.00
 Command ===>
******************************** Top of Data *
//G51ACR  PROC ENV=,Z=G51ACRZ,PARMS=' '
// SET ROOT='/wasv6config/g5cell'
```

c. BBOWMG1F enables PRR (Peer Resource Recovery) processing.

This job modifies the V5 configuration and updates an XML setting so that when the server is started PRR can take place. Running this job is only required when you have XA connectors installed.

After running this job, one of the application servers in the node must be started so PRR can take place. When things work correctly, the server comes up and then automatically shuts down after a minute or so.

d. BBOWMG2F disables PRR processing. This job modifies the V5 configuration XML and disables the change made by BBOWMG1F. If BBOWMG1F was not run, then you do not need to run BBOWMG2F. But if BBOWMG1F was run, then running BBOWMG2F is mandatory.

   This job should be run when no servers in the node are running. And after job completion, no server in the node should be started until the node has been migrated.

e. BBOWMG3F performs the migration.

   Be sure the JOB card has a USER= and PASSWORD= so this job can run under the ID of the WebSphere Admin ID of G5ADMIN. This job must run under that ID.

   Submit the job and wait for completion. This job takes a long time to complete.

   In our case, the job took 31 minutes.

```
15.28.35 ---- SATURDAY,  19 MAR 2005 ----
15.28.35  $HASP373 BBOWMG3F STARTED
15.28.38  Jobname  Procstep Stepname  CPU Time   RC
15.28.38  BBOWMG3F --None-- SETUP     00:00:00   00
15.28.38  BBOWMG3F --None-- WRCONFIG  00:00:00   00
15.28.38  BBOWMG3F --None-- WRRESP    00:00:00   00
15.28.50  BBOWMG3F --None-- MKCONFIG  00:00:00   00
15.28.50  BBOWMG3F --None-- VERIFY    00:00:00   00
15.29.08  BBOWMG3F --None-- CRHOME    00:00:00   00
15.30.54  BBOWMG3F --None-- PREUPGRD  00:00:00   00
15.36.15  BBOWMG3F --None-- CRPROF    00:00:00   00
15.55.15  BBOWMG3F --None-- UPGRADE   00:00:00   00
15.58.39  BBOWMG3F --None-- UPPROCS   00:00:00   00
15.59.47  BBOWMG3F --None-- FINISHUP  00:00:00   00
15.59.47  BBOWMG3F --None-- WROUT     00:00:00   00
15.59.47  BBOWMG3F --None-- WRERR     00:00:00   00
15.59.47  $HASP395 BBOWMG3F ENDED
```

23. Perform post-migration RACF work by following these three steps:

   a. Create new STARTED profiles. This is mentioned under Step 7 of the pre-job instructions. If we had provided new JCL start procedure names this would have been necessary. Our example backs up the V5 JCL procedures and copies in the V6 procedures with the same names, so no new STARTED profiles are needed. The BBOMMINS instruction member provides an example for the controller JCL only.

   b. Permitting the servant region ID access to BPX.WLMSERVER. One of the instructions found under the BBOWMG3F job says the following:

      > *WebSphere for z/OS V6 requires that the servant's address space is authorized to interface with the z/OS Workload Manager. The following RACF commands must be issued to permit the Application Server's started task userid to have Read access to the WLM Server profile.*

      Submit the following RACF commands to permit the servant ID (G5ASRU) access to the BPX.WLMSERVER class:

      ```
      PE BPX.WLMSERVER CLASS(FACILITY) ID(G5ASRU) ACCESS(READ)
      SETROPTS RACLIST(FACILITY) REFRESH
      ```

   c. Create a keyring for the servant ID and add the CA certificate to the keyring. Version 6 has the requirement that servant regions have access to a keyring with the CA certificate. If your servant region ID is identical to the controller region ID, you are all set. But if the servant ID is different from the controller ID, then a new keyring and certificate for that keyring is required.

Submit the following RACF command to add a keyring to the G5DSRU ID:

```
RACDCERT ADDRING(WASKeyring) ID(G5ASRU)
```

Submit the following RACF command to connect the Certificate Authority's certificate to the new keyring just created.

```
RACDCERT ID(G5ASRU) CONNECT(RING(WASKeyring)LABEL('WebSphereCA') CERTAUTH
```

24. Start the node agent on SYSD first.

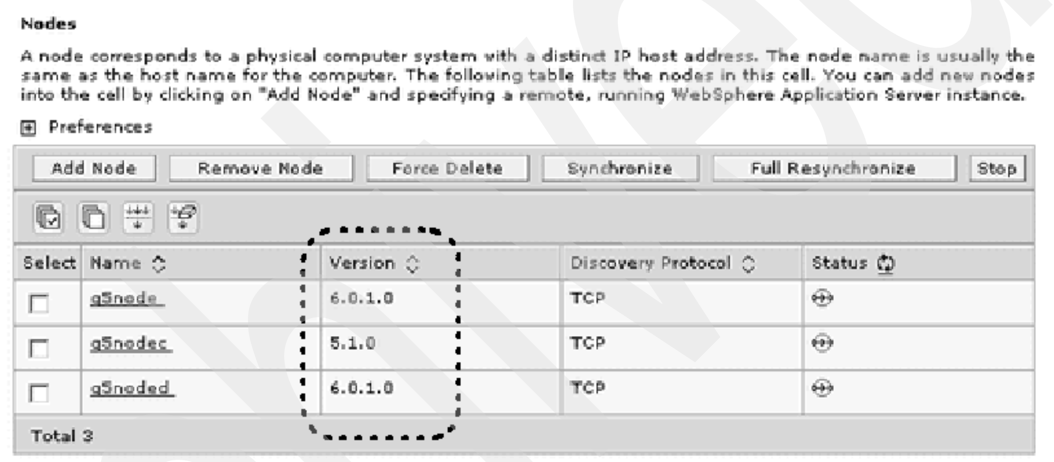
    Issue the start command for the node agent:

    ```
    S G51ACR,ENV=G5CELL.G5NODED.G5AGNTD,JOBNAME=G5AGNTD
    ```

    You then see this message in the node agent controller SYSPRINT:

    ```
    WSVR0001I: Server CONTROL PROCESS nodeagent open for e-business
    ```

    The administrative console in Figure 13-44 shows that g5noded is identified as a V6 node and g5nodec is identified as a v5.1 node.



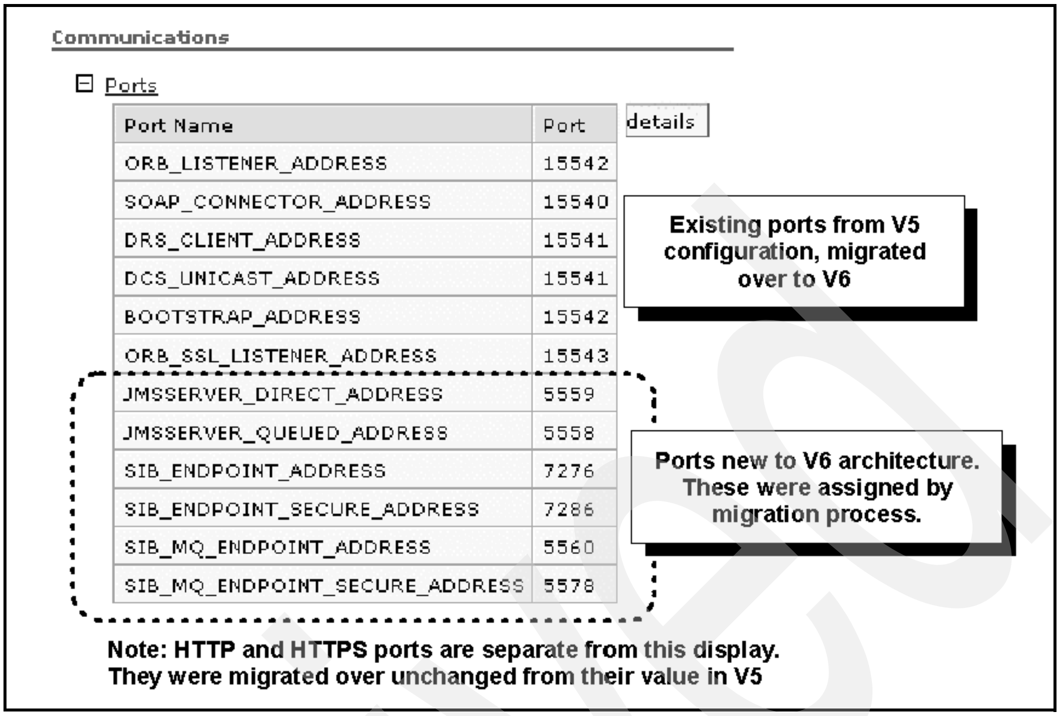
Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

Select	Name	Version	Discovery Protocol	Status
□	g5node	6.0.1.0	TCP	⊕
□	g5nodec	5.1.0	TCP	⊕
□	g5noded	6.0.1.0	TCP	⊕
Total 3				

Figure 13-44 Administrative console after node D migration

25. View the new ports that the migration has created for the V6 application server. The deployment manager and node agent do not have new ports added, so you only have to examine migrated application servers. Figure 13-45 shows the ports as assigned by the migration process.



Communications

⊟ Ports

Port Name	Port
ORB_LISTENER_ADDRESS	15542
SOAP_CONNECTOR_ADDRESS	15540
DRS_CLIENT_ADDRESS	15541
DCS_UNICAST_ADDRESS	15541
BOOTSTRAP_ADDRESS	15542
ORB_SSL_LISTENER_ADDRESS	15543
JMSSERVER_DIRECT_ADDRESS	5559
JMSSERVER_QUEUED_ADDRESS	5558
SIB_ENDPOINT_ADDRESS	7276
SIB_ENDPOINT_SECURE_ADDRESS	7286
SIB_MQ_ENDPOINT_ADDRESS	5560
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5578

Existing ports from V5 configuration, migrated over to V6

Ports new to V6 architecture. These were assigned by migration process.

Note: HTTP and HTTPS ports are separate from this display. They were migrated over unchanged from their value in V5

Figure 13-45 Application port assignments after migration

You could leave these as the migration process assigned them, but the migration process has no knowledge of TCP port usage elsewhere on the communication stack, so these numbers may or may not already be in use. You must determine whether to re-map these to numbers you know are good.

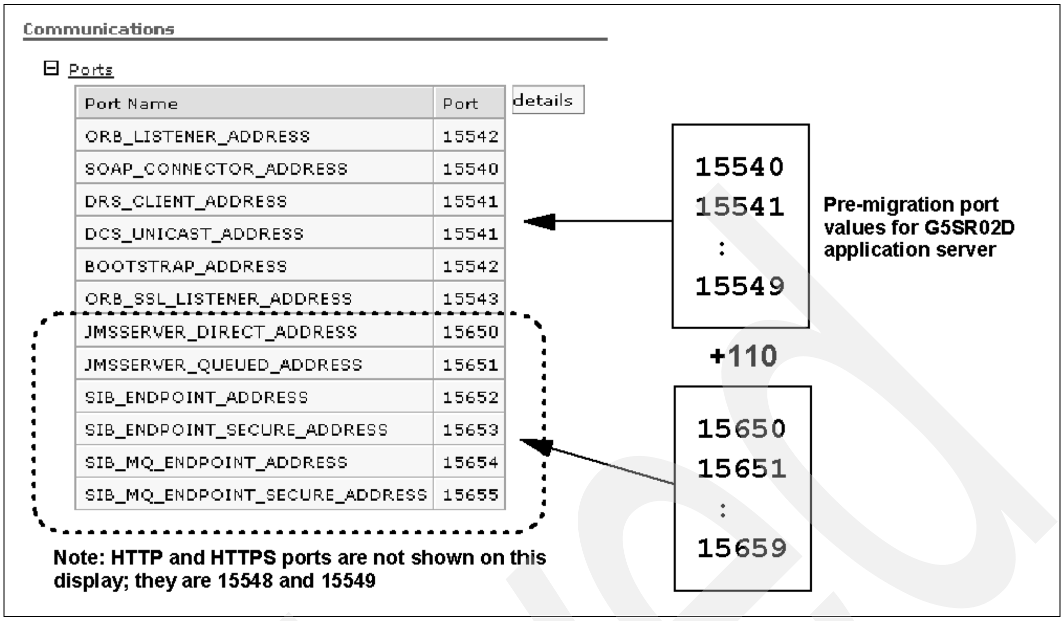
One very simple approach is to add 110 to the start of the server's port range and use that result as a secondary range for the new ports.

a. Under g5sr02d server, select Ports in the administrative console.

b. Change the port values one by one to the new values.

c. Save the changes to the master configuration and synchronize.

Figure 13-46 shows the same port list after assigning the new ports to a range that is unlikely to conflict with other servers.



Communications

⊟ Ports

Port Name	Port
ORB_LISTENER_ADDRESS	15542
SOAP_CONNECTOR_ADDRESS	15540
DRS_CLIENT_ADDRESS	15541
DCS_UNICAST_ADDRESS	15541
BOOTSTRAP_ADDRESS	15542
ORB_SSL_LISTENER_ADDRESS	15543
JMSSERVER_DIRECT_ADDRESS	15650
JMSSERVER_QUEUED_ADDRESS	15651
SIB_ENDPOINT_ADDRESS	15652
SIB_ENDPOINT_SECURE_ADDRESS	15653
SIB_MQ_ENDPOINT_ADDRESS	15654
SIB_MQ_ENDPOINT_SECURE_ADDRESS	15655

Note: HTTP and HTTPS ports are not shown on this display; they are 15548 and 15549

Figure 13-46 Revised port assignments

26. Start Application server G5SR02D.

Issue the start command for the G5SR02D application server on SYSD:

```
S G51ACR,ENV=G5CELL.G5NODED.G5SR02D,JOBNAME=G5SR02D
```

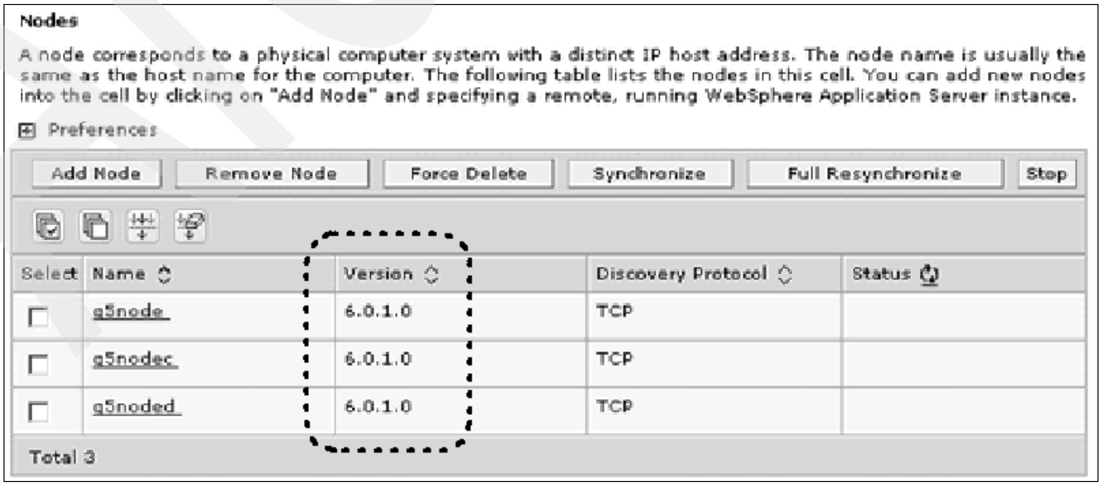
Watch for this message:

```
WSVR0001I: Server SERVANT PROCESS g5sr02d open for e-business
```

13.1.3 Migrate the application server node on SYSC

The steps to migrate the node on SYSC are identical to those for the node on SYSD. Follow the steps in 13.1.2, "Migrating the application server node on SYSD" on page 259.

Upon completion of all the steps, viewing the node status shows that all the nodes are now migrated to V 6.0, as shown in Figure 13-47.



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

⊞ Preferences

Select	Name	Version	Discovery Protocol	Status
□	g5node	6.0.1.0	TCP	
□	g5nodec	6.0.1.0	TCP	
□	g5noded	6.0.1.0	TCP	
Total 3				

Figure 13-47 Node status after completing all node migration

13.1.4 Daemon migration problem

If you have this particular coincidence of configuration specifics, you should read this section for a preview of what might occur and how to remedy the situation.

1. Use of shared procedures between servers, particularly the daemon servers, and to a lesser degree the application servers.
2. Re-use of the same JCL procedure names for the V6 configuration.
3. Use of STEPLIB statements to point to the product libraries.

At the point where the migration of the deployment manager is complete, our configuration has new copies of the following JCL start procedures:

- G51DCR - the deployment manager controller start procedure
- G51ACR - the deployment manager servant start procedure
- G51DEMN - the daemon server's start procedure

The deployment manager procedures, G51DCR and G51ACR, are not a problem because there is only one deployment manager. The problem at this point was G51DEMN, the daemon JCL start procedure. The copy of that in SYS1.PROCLIB was now updated to the V6.0 level, with STEPLIB pointers to the V6 code libraries.

At the point where the deployment manager is migrated, the SYSD daemon is not yet migrated, yet the JCL start procedure that would be used to start that daemon is already migrated over to a V6 copy of the procedure. If you do not have to restart the daemon, there is no problem. But if that daemon is ever restarted, it would pick up the V6 copy of the JCL and try to start that daemon as a V6 daemon.

There are several remedies or work-arounds. Choosing any one of the following remedies will avoid the problem entirely:

- Do not configure the daemon servers on both MVS images to use the same procedure name.
- In the customization panels for the deployment manager migration, we chose to use the same JCL start procedure names. See Figure 13-12 on page 250 for the dialog where this is specified. If you use a new JCL procedure name for the daemon server, the problem is avoided. Specifying a new procedure name would have required a new STARTED profile.
- Configure your system to have a procedure library higher in the concatenation than SYS1.PROCLIB. You could then copy your backup of the V5.1 daemon JCL procedure to that system-specific proclib for SYSD. Thus, whenever the G51DEMN proc is invoked it would pick up that system's copy, which would be the V5.1 copy.
- Replace the V6 copy of G51DEMN in SYS1.PROCLIB with the V5.1 backup copy of the procedure in the event you need to restart the SYSD daemon. As long as the SYSC daemon is up and running, it wouldn't need to access the JCL in proclib.
- If you have the opportunity to load V6 modules into LPA/LNKLST on SYSC and V5 modules into LPA/LNKLST on SYSD, you could use the same daemon procedure (minus STEPLIB statements) and avoid the problem. For the daemon, this is relatively easy, since the daemon server does not have a setupCmdLine.sh shell script as part of its configuration. The setupCmdLine.sh file for a node may have STEPLIB statements in it that have to be taken into account.

The application servers also face a similar problem with procedure sharing. The problem in this case is slightly different. The Work Load Manager for z/OS (WLM) is in control of starting servant regions. With a common servant region procedure, G51ASR, WLM tries to start another servant region in a not-yet-migrated node using the G51ASR procedure that has been migrated to V6. The result would be a server controller running at V5.1 and a servant at V6.

The work-arounds are similar to what we saw for the daemon:

- Avoid using a common procedure between nodes on different MVS images.
- Create new procs with new names for the migrated node and pay the cost of creating new STARTED profiles for them.
- System-specific proclibs and the careful management of what version of procedures reside in which system-specific proclibs.
- Careful management of what is in LPA/LNKLST for each system and the removal of STEPLIB from the JCL and the setupCmdLine.sh file for the node.

13.1.5 Known migration issues

There are known issues that you should be aware of if you encounter problems migrating, including the following:

1. When performing V5.0 to V5.1 migration, it is recommended that you *not* have any region constraints in place, such as IEFUSI limits, because these can cause unpredictable JVM errors. The IEFUSI in place can cause a limit on the amount of region space allowed by a system, which can limit memory in a fashion of which the user is not aware. This memory limit could stop the JVM from initializing completely during servant startup, which can account for the following error:

   ```
   BBOO0072E Shasta Runtime function loadAndInitVM detected
   ```

2. The instruction member for the migration of a federated application server node or a stand-alone server node fails to call out the potential need to create a keyring for the servant ID and to add the CA certificate to that keyring. In Version 6 the servant regions need access to a keyring with the CA certificate. There are cases where this step is not necessary:

   a. If the servant region is using the same RACF ID as the controller, then it has access to the keyring and CA certificate that was created when the V5 cell was built.

   b. If the servant region for a server is using the same ID as other servant regions, then you have already created the new keyring and added the CA certificate for that ID. In our example, the deployment manager servant ID is different from the deployment manager controller ID, so we must issue the commands to create the keyring and add the CA certificate. The application server node servant ID is different from the deployment manager servant ID, so we must do the same when we migrate the node on SYSD. But we do not have to when we migrated the SYSC node because that node uses the same servant ID as SYSD. Similarly, we did not have to do it when we migrated the application server node because that servant also used the same ID as the servants for the cell.

   V6 requires servant regions to have access to a keyring and CA certificate. The keyring name should be the same one used by the V5 controllers, and the CA certificate label should be the same as well. If you are using unique IDs for each server, then every server needs new keyrings built for the servants. But if you are sharing an ID between multiple servers, then you may need to do it only once.

13.1.6 Reverting a node back to V5

When the BBOWMG3D/F job is run, it disables the source node configuration structure such that it cannot be started.

The file serverindex.xml name is changed to serverindex.xml_disabled. Every node has a serverindex.xml file. It is located in the *node level* directory for that node. Figure 13-48 shows an example of where the serverindex.xml file is for the G5CELL's deployment manager node.

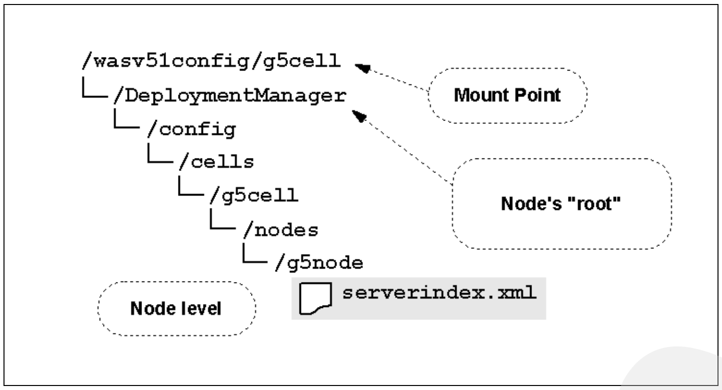


Figure 13-48 Location of serverindex.xml

In addition to renaming, the migration copies a JACL script into the bin directory. Figure 13-49 shows how to find the bin directory for the node. The JACL script is a simple input file for WSADMIN script interface processing. The JACL script is called migrationDisablementReversal.jacl.

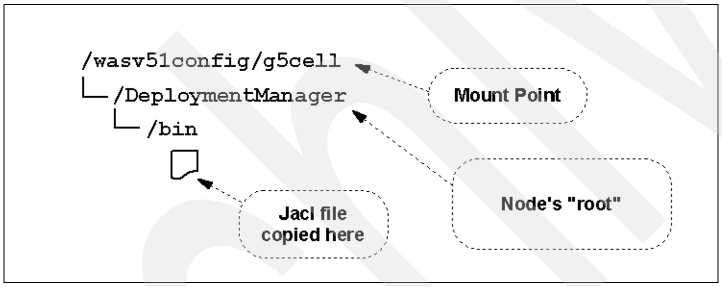


Figure 13-49 Location of bin directory

Restoring a migrated configuration manually

The process is relatively simple:

1. Stop the V6 servers. They cannot be up at the same time the V5.x copy of the server is up.
2. Unmount the V6 configuration HFS. It is best not to have this around to confuse matters. You may wish to delete the HFS file system altogether after unmounting.
3. Copy back to PROCLIB the V5.x JCL you backed up earlier.
4. Change serverindex.xml file name. Go into the node level directory of each node that was migrated and rename serverindex.xml_disabled to serverindex.xml.
5. Start the V5.x server.

Restoring a migrated configuration using a script

The steps are:

1. Stop the V6 servers. They can't be up at the same time the V5.x copy of the server is up.
2. Unmount the V6 configuration HFS. It is best not to have this around to confuse matters. You may wish to delete the HFS file system altogether after unmounting.
3. Restore procedures; copy back to PROCLIB the V5.x JCL you backed up earlier.

4. From a Telnet or OMVS session, go to the /bin directory of your V5 node.
5. Issue the following command:

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

6. Start the V5.x servers as you normally would.

13.1.7 Recovering from BBOWMG3* job failure

The BBOWMG3* job has 13 steps, each of which must complete successfully for the total migration to be successful. If any of the steps fail, do the following:

1. Under the V6 configuration HFS mount point, delete the node's "root" (or "home") directory and all subdirectories. Pay close attention to where you are and what you delete when you do this step. Make sure you're at the V6 HFS and not the V5, and make sure you delete the node you wish to delete and not some other already-migrated and working node. For instance, a deployment manager node's home is /DeploymentManager. For the G5CELL the application server node on SYSC was /AppServerNodeC. You only have to delete the just the failed node's home, but not the entire HFS.
2. Fix whatever caused BBOWMG3* to fail.
3. Resubmit the BBOWMG3* job.

13.2 Manual migration: Installing a V6 stand-alone server

This procedure walks you through the steps to build a simple stand-alone server configuration. We assume that the SMP/E installation work has been accomplished. In addition we assume that the prerequisite work has been done to allow WebSphere to operate on the MVS image. Instructions on what is involved for that are included in Appendix B, "Z/OS preparation steps" on page 341.

13.2.1 Planning worksheets

You should make a copy of the following worksheets and then fill in the empty fields according to the instructions. Filling out the information in advance will make the navigation of the ISPF dialog screens go more smoothly with less confusion over what data should go into which fields.

WebSphere for z/OS SMP/E information

Table 13-6 SMP/E information

	Enter your values here
High level qualifier of target data sets	
Mount point of SBBOHFS data set	

MVS system information

Table 13-7 MVS system information

	Enter your values here
MVS sysname where configuration will run	
Sysplex name[1]	
Location of PARMLIB data set	
Location of PROCLIB data set	
Location of SYSEXEC data set	
Location of SCEERUN data set	
Location of SCEERUN2 data set	
Location of SGSKLOAD data set [2]	
IP address of adapter on LPAR [3]	
Host name of LPAR where server will reside	
Userid with UID=0 authority and password	

Notes:

1. Specify the sysplex name even if you have a monoplex.
2. You do not need the SGSKLOAD data set if you have z/OS 1.6 or above.
3. You must specify the IP address in certain places where the DNS host name will not suffice, so record the IP address in dotted decimal notation.

WebSphere names

Create a unique two-character uppercase string that is to be your “cell identifier.” RACF userids and groups are based on this cell identifier. Populate your cell identifier into the open cells in Figure 13-50 and Figure 13-51.

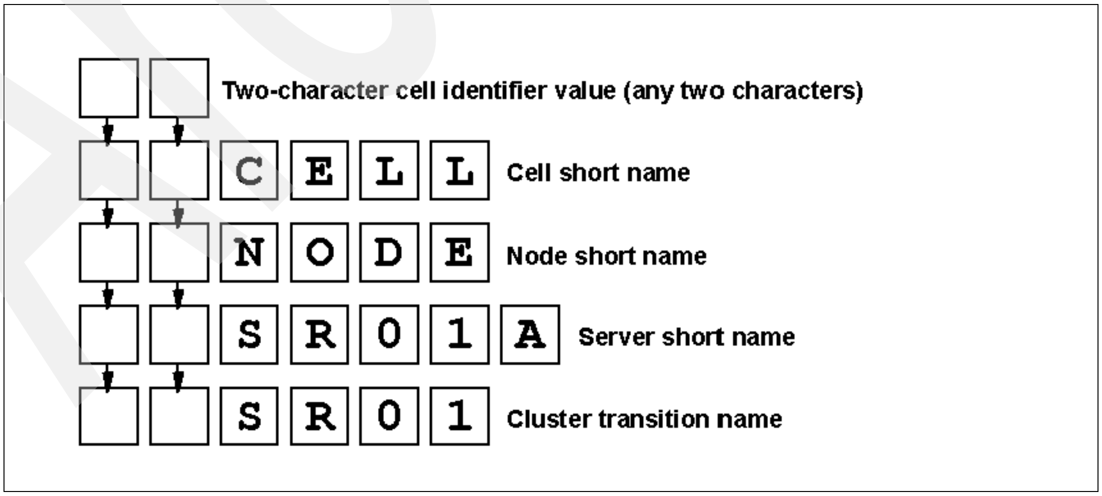


Figure 13-50 Creating WebSphere names using cell indenter

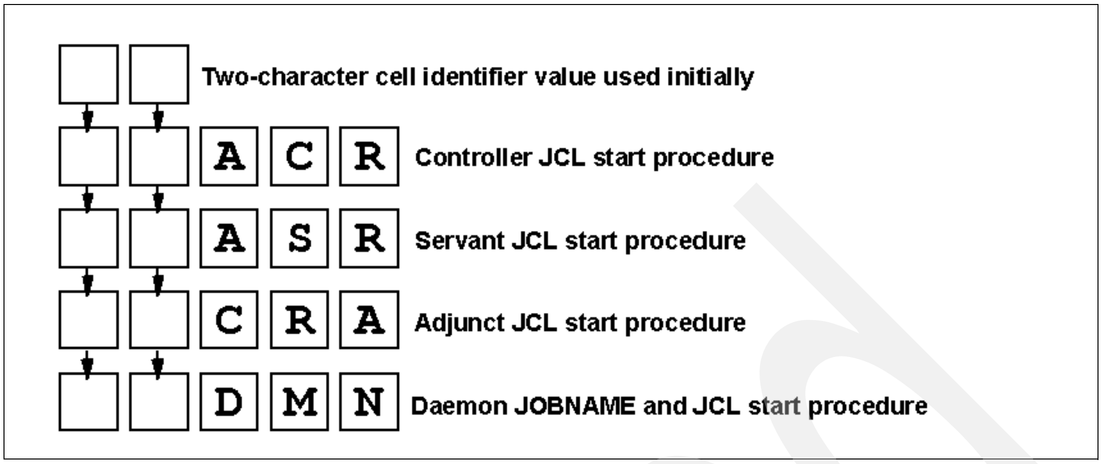

Two-character cell identifier value used initially
A C R
Controller JCL start procedure
A S R
Servant JCL start procedure
C R A
Adjunct JCL start procedure
D M N
Daemon JOBNAME and JCL start procedure

Figure 13-51 Creating JCL start procedure names using cell identifier

Choose a two-number prefix between 10 and 60 for TCP port assignments. The objective is to make these ports unique so there are no port conflicts. Pick a two-number prefix you are sure is not used by anything else on the MVS image. Enter this prefix in the empty cells in Figure 13-52.

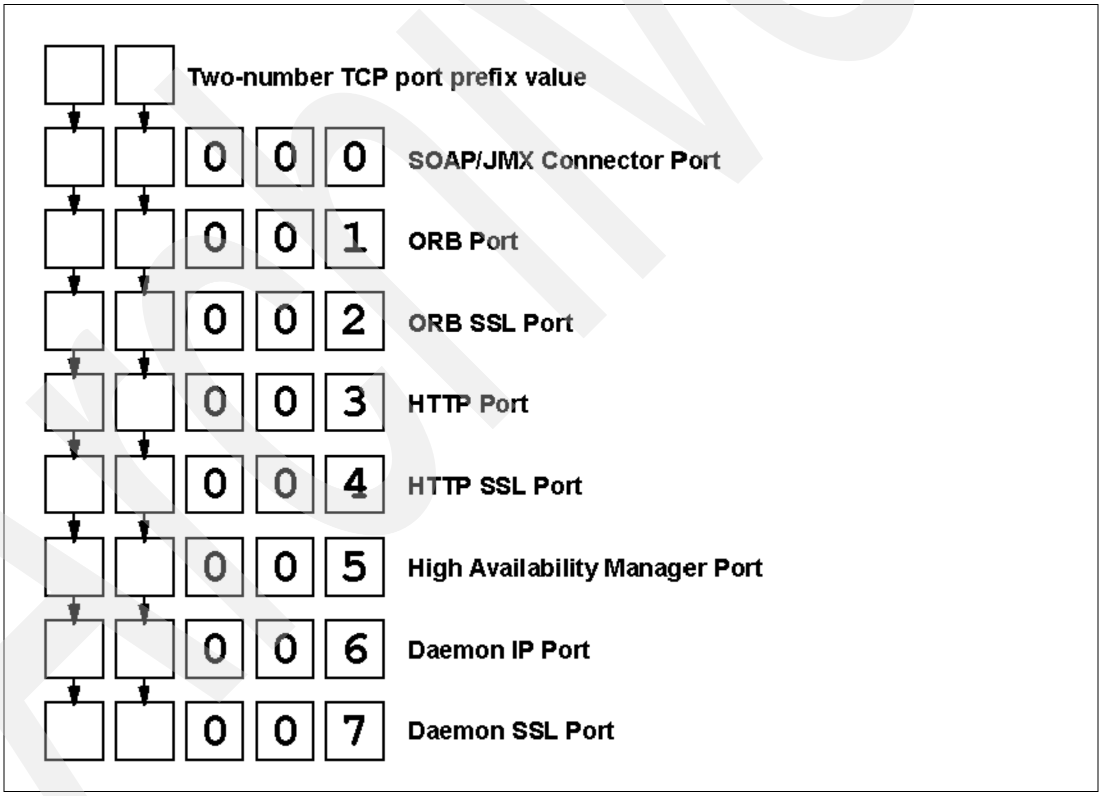


Figure 13-52 TCP port assignments

13.2.2 ISPF customization dialogs

This section takes you through all the steps necessary to run ISPF to create the customization jobs.

1. Log on to TSO on the system on which the server will run. This is not absolutely necessary, but it is better if you do. From option **6** invoke the following command:

   ```
   'hlq.SBBOCLIB(BBOWSTRT)' 'APPL(cell_id)'
   ```

   where:

   – `hlq` is the high-level qualifier of the target data sets you specified in Table 13-6.

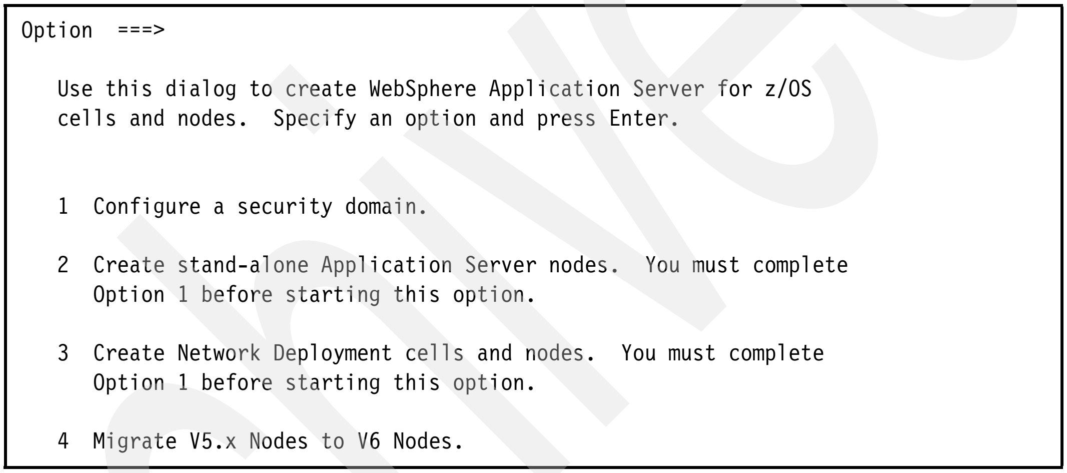
   – `cell_id` is the cell identifier you chose in Figure 13-50.

   – The command argument '`APPL(cell_id)`' is a way to instruct ISPF to create a brand new set of variables for this customization run.

   Here is what the command looks like for our system scenario:

   ```
   'WAS600.WAS.SBBOCLIB(BBOWSTRT)' 'APPL(XX)'
   ```

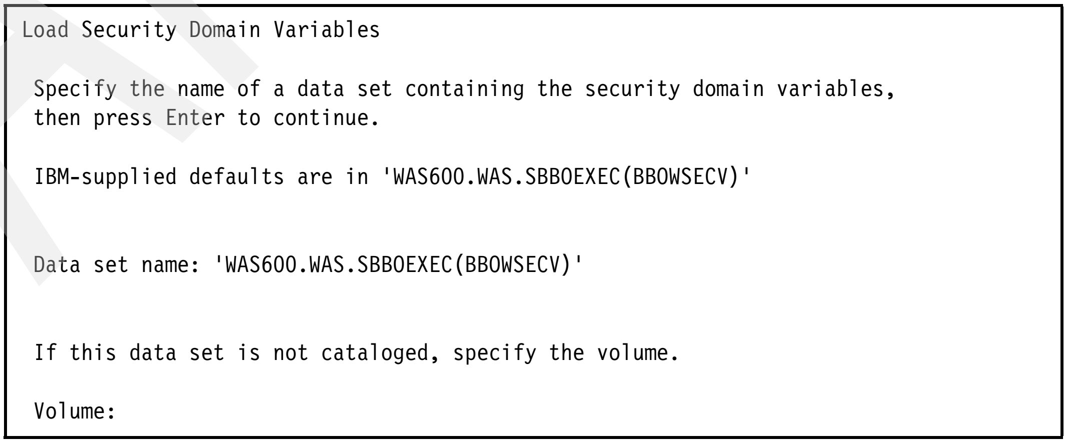
2. Press Enter to clear the first panel you see.
3. Press F3 to clear the license information. You should now see a panel that looks like Figure 13-53.



```
Option  ===>

   Use this dialog to create WebSphere Application Server for z/OS
   cells and nodes.  Specify an option and press Enter.


   1  Configure a security domain.

   2  Create stand-alone Application Server nodes.  You must complete
      Option 1 before starting this option.

   3  Create Network Deployment cells and nodes.  You must complete
      Option 1 before starting this option.

   4  Migrate V5.x Nodes to V6 Nodes.
```

Figure 13-53 Initial customization dialog

4. Select option **1**, Configure a security domain and press Enter.
5. Select option **L**, Load security domain variables.
6. Specify the values in the screen shown in Figure 13-54.



```
Load Security Domain Variables

 Specify the name of a data set containing the security domain variables,
 then press Enter to continue.

 IBM-supplied defaults are in 'WAS600.WAS.SBBOEXEC(BBOWSECV)'


 Data set name: 'WAS600.WAS.SBBOEXEC(BBOWSECV)'


 If this data set is not cataloged, specify the volume.

 Volume:
```

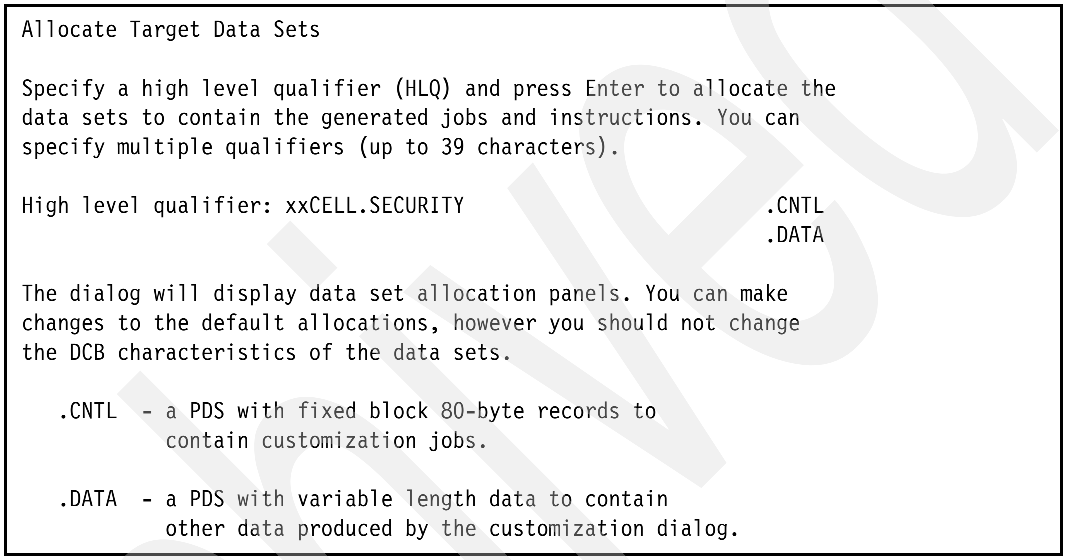
Figure 13-54 Specify IBM defaults for security domain

Make certain the IBM-supplied defaults data set is used for data set name. You specify your specific high-level qualifier in Table 13-6 on page 276. Press Enter.

7. Select option **1**, Allocate target data sets.
8. In the next dialog, shown in Figure 13-55, provide a high-level qualifier for the target data sets of `xxCELL.SECURITY`, where `xx` is the two-character cell identifier you wrote into the worksheet in Figure 13-50.

   Press Enter.

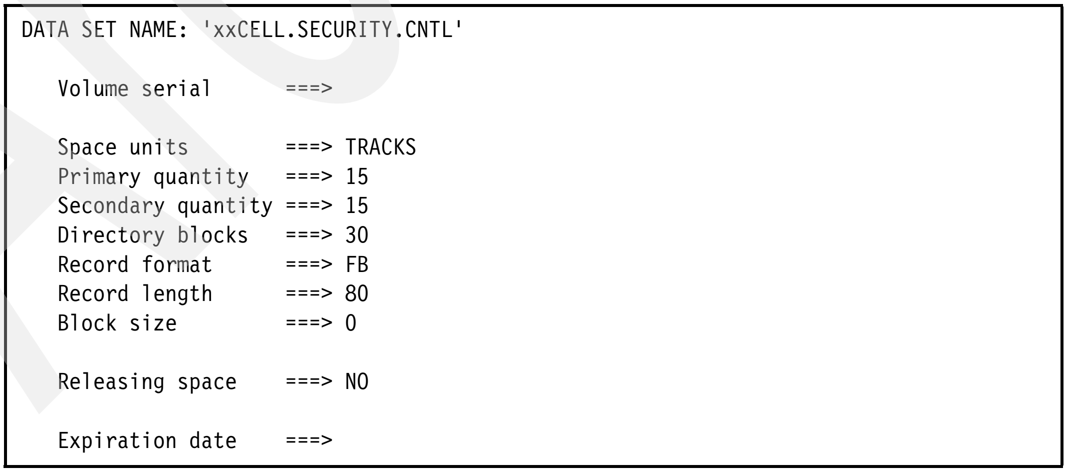
   This results in two data sets being created. One data set ends in CNTL and the other ends in DATA. The *target data sets* are where the customized jobs are written when you generate the jobs.



```
Allocate Target Data Sets

Specify a high level qualifier (HLQ) and press Enter to allocate the
data sets to contain the generated jobs and instructions. You can
specify multiple qualifiers (up to 39 characters).

High level qualifier: xxCELL.SECURITY                         .CNTL
                                                              .DATA

The dialog will display data set allocation panels. You can make
changes to the default allocations, however you should not change
the DCB characteristics of the data sets.

   .CNTL  - a PDS with fixed block 80-byte records to
            contain customization jobs.

   .DATA  - a PDS with variable length data to contain
            other data produced by the customization dialog.
```

Figure 13-55 Allocating the target data sets

9. The next dialog, shown in Figure 13-56, asks for allocation parameters for the **.CNTL data set. Accept the default values and press Enter.



```
DATA SET NAME: 'xxCELL.SECURITY.CNTL'

   Volume serial      ===>

   Space units        ===> TRACKS
   Primary quantity   ===> 15
   Secondary quantity ===> 15
   Directory blocks   ===> 30
   Record format      ===> FB
   Record length      ===> 80
   Block size         ===> 0

   Releasing space    ===> NO

   Expiration date    ===>
```

Figure 13-56 Allocating CNTL data set

10. Accept the defaults for the **.DATA data set in the next dialog and press Enter.

11. You should be at the Security Domain main option panel. Select option **2**, Define variables.

12. In the next dialog, shown in Figure 13-57, fill in these fields:

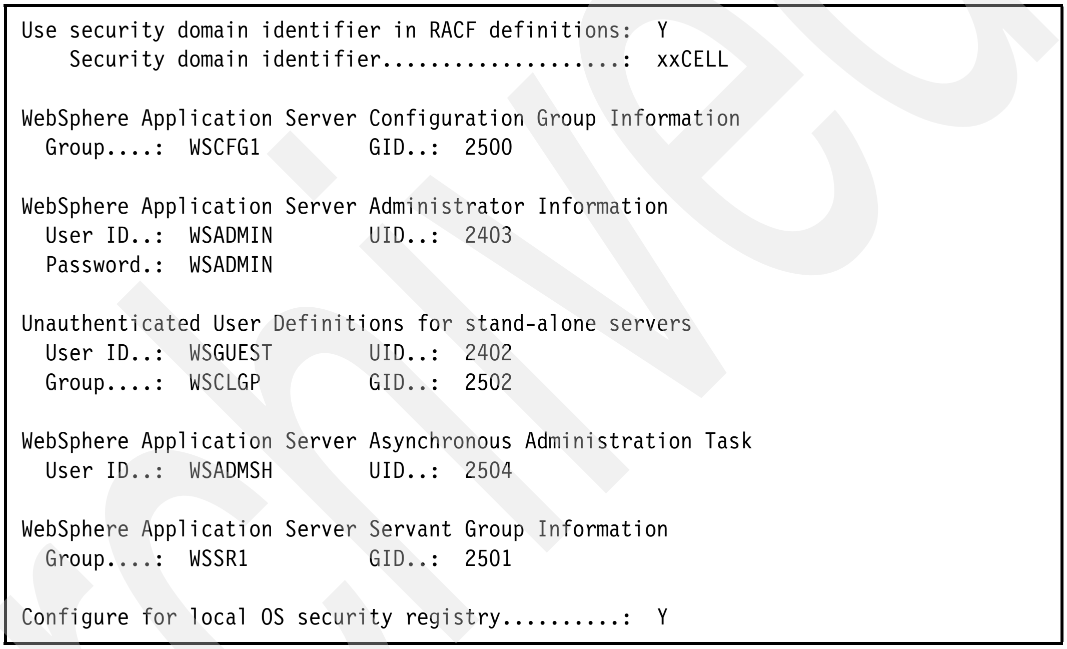
    a. Specify `Y` for the field “Use security domain identifier in RACF definitions.”

    b. Specify the cell short name as `xxCELL`, but replace `xx` with your two-character cell identifier.

    c. Replace the default `WS` prefix with your two-character cell identifier. Change all the user and group names to begin with your cell identifier.

    d. Examine the UID and GID values and determine if they are in use. If they are in use, you need to change them. If you are not sure, you may wish to consider pre-pending each with a 1 or perhaps the two-digit TCP port prefix you specified in Figure 13-52.

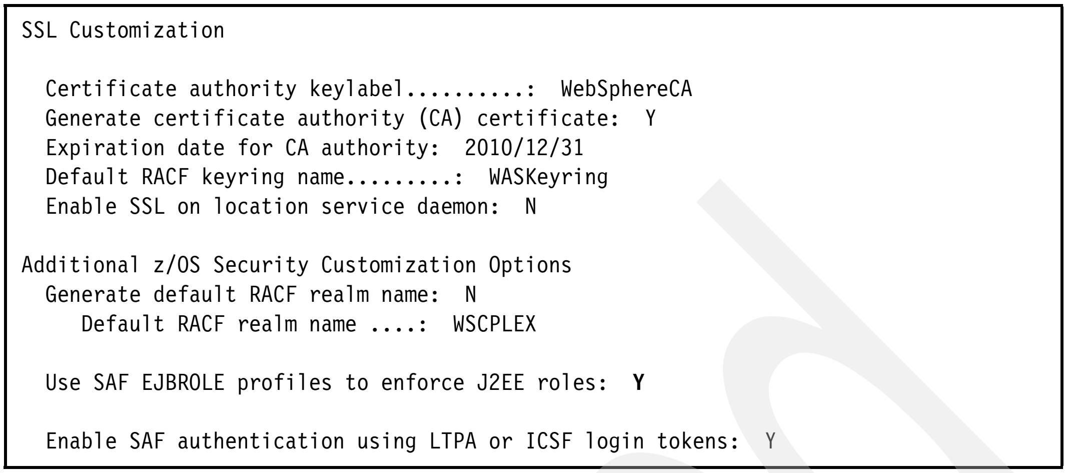
    Press Enter.



```
Use security domain identifier in RACF definitions:  Y
    Security domain identifier....................:  xxCELL

WebSphere Application Server Configuration Group Information
  Group....:  WSCFG1         GID..:  2500

WebSphere Application Server Administrator Information
  User ID..:  WSADMIN        UID..:  2403
  Password.:  WSADMIN

Unauthenticated User Definitions for stand-alone servers
  User ID..:  WSGUEST        UID..:  2402
  Group....:  WSCLGP         GID..:  2502

WebSphere Application Server Asynchronous Administration Task
  User ID..:  WSADMSH        UID..:  2504

WebSphere Application Server Servant Group Information
  Group....:  WSSR1          GID..:  2501

Configure for local OS security registry..........:  Y
```

Figure 13-57 Security domain, 1 of 2

13. In the next dialog, shown in Figure 13-58, specify `Y` for the field “Use SAF EJBROLE profiles to enforce J2EE roles” and press Enter.



```
SSL Customization

  Certificate authority keylabel..........:  WebSphereCA
  Generate certificate authority (CA) certificate:  Y
  Expiration date for CA authority:  2010/12/31
  Default RACF keyring name.........:  WASKeyring
  Enable SSL on location service daemon:  N

Additional z/OS Security Customization Options
  Generate default RACF realm name:  N
     Default RACF realm name ....:  WSCPLEX

  Use SAF EJBROLE profiles to enforce J2EE roles:  Y

  Enable SAF authentication using LTPA or ICSF login tokens:  Y
```

Figure 13-58 Security domain 2 of 2

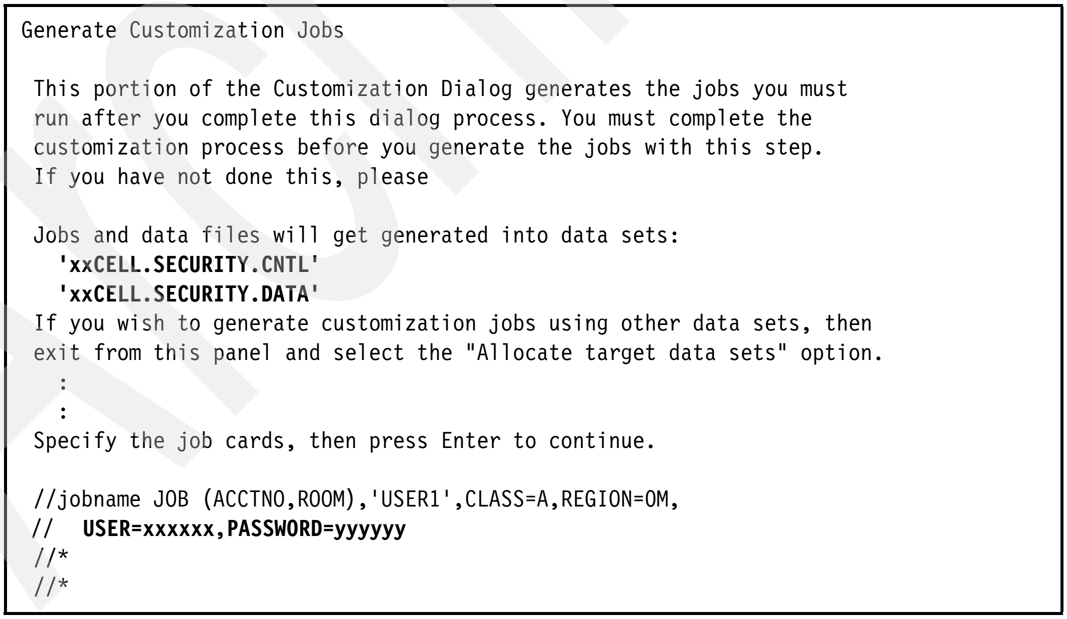
14. Select option **4**, Generate customized jobs.

15. In the next dialog, shown in Figure 13-59, verify or change the following fields:

    a. Verify your target data sets are shown under “Jobs and data files will get generated into data sets.”

    b. Add a comma to the end of the job card, then add a USER and PASSWORD so the customized jobs all run under the UID=0 ID. Specifying a USER= and PASSWORD= on the JOB card is not strictly required, provided you ensure the generated jobs run under the UID=0 ID.

    Press Enter.



```
Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please

 Jobs and data files will get generated into data sets:
   'xxCELL.SECURITY.CNTL'
   'xxCELL.SECURITY.DATA'
 If you wish to generate customization jobs using other data sets, then
 exit from this panel and select the "Allocate target data sets" option.
   :
   :
 Specify the job cards, then press Enter to continue.

 //jobname JOB (ACCTNO,ROOM),'USER1',CLASS=A,REGION=0M,
 //  USER=xxxxxx,PASSWORD=yyyyyy
 //*
 //*
```

Figure 13-59 Updating JOB card and generating customized jobs

16. Figure 13-60 shows the customized jobs being written out to the target data sets. When complete, press Enter.

```
Processing for data set 'XXCELL.SECURITY.CNTL' ...
Member BBOSBRAJ successfully created.
Member BBOSBRAK successfully created.
Member BBOSDFL successfully created.
Member BBOSDINS successfully created.

Processing for data set 'XXCELL.SECURITY.DATA' ...
Member BBOSBRAC successfully created.
***
```

Figure 13-60 Customized jobs written to target data sets

17. You should be at the Security Domain main option panel. Select option **3**, Save security domain variables.
18. Provide a name for the variable data set, as shown in Figure 13-61. The high-level qualifier should be `xxCELL.SECURITY`, where `xx` is your two-character cell identifier. The last qualifier should be SDCFG.

```
Data set name: 'xxCELL.SECURITY.SDCFG'
```

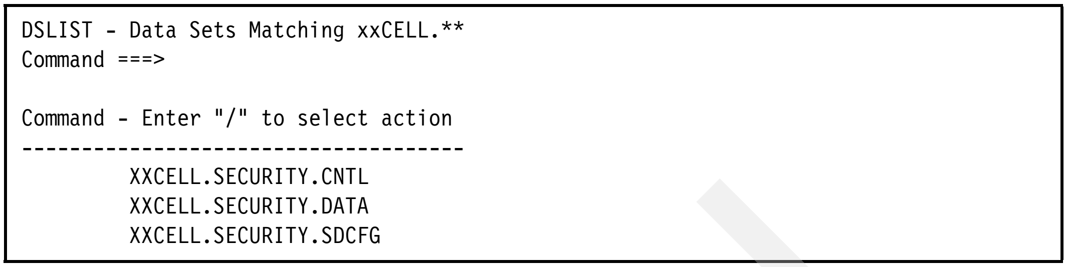
Figure 13-61 Data set where security variables are saved

19. Accept the default allocation parameters for the data set, as shown in Figure 13-62.

```
DATA SET NAME: 'XXCELL.SECURITY.SDCFG'

   Volume serial      ===>

   Space units        ===> TRACKS
   Primary quantity   ===> 2
   Secondary quantity ===> 2
   Directory blocks   ===> 0
   Record format      ===> VB
   Record length      ===> 255
   Block size         ===> 0

   Releasing space    ===> NO

   Expiration date    ===>
```

Figure 13-62 Allocation parameters for security variables data set

20. You should be at the Security domain main option panel. Press F3 to return to the primary option menu.
21. Press F3 to exit out of the ISPF customization dialogs.
22. Go to ISPF option **3.4** and list the data sets that start with `xxCELL.**`, where `xx` is your two-character cell identifier. You should see a data set list similar to that shown in Figure 13-63.



```
DSLIST - Data Sets Matching xxCELL.**
Command ===>

Command - Enter "/" to select action
------------------------------------
         XXCELL.SECURITY.CNTL
         XXCELL.SECURITY.DATA
         XXCELL.SECURITY.SDCFG
```

Figure 13-63 Listing of security data sets

The data sets that have been created are described in Table 13-8.

Table 13-8 Cell security data sets

Data set name	Data set type	Data set contents
XXCELL.SECURITY.CNTL	FB 80 PDS	Customized JCL
XXCELL.SECURITY.DATA	VB 255 PDS	RACF script
XXCELL.SECURITY.SDCFG	VB 255 SEQ	Security domain variables

13.2.3 Submit security customization jobs

It is important to run these jobs now, rather than waiting until the standalone server's jobs are generated. Because the SDCFG variable data set is an important input to the stand-alone server's customization run, it is best to make sure the Security Domain job runs correctly before going through the effort of customizing the stand-alone server. If something goes wrong with these jobs, you should correct the problems now, before proceeding with server customization.

Follow these steps to submit the customization jobs:

1. Go to the xxCELL.SECURITY.CNTL data set, where xx is your two-character cell identifier.
2. Submit the BBOSBRAJ job. This does nothing more than create the BBOSBRAK member in the DATA target data set.
3. Make certain that the ID under which the BBOSBRAK job runs has the RACF SPECIAL, which is necessary to create the profiles.
4. Submit the BBOSBRAK job. The BBOSBRAK job in the CNTL data set calls the just-created BBOSBRAK member in DATA. This creates the various RACF profiles.
5. Inspect the output and make sure all the profiles are properly created.

> **Note:** There is a chance that the BBOSBRAK job will encounter profiles that already exist in your RACF database, depending on whether any WebSphere V4 or V5 work had been done before. In particular, you may see the following profiles flagged as already created:
>
> - CB.BIND.* and CB.* in the CBIND class
> - CBS390 in the APPL class
> - IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING in the FACILITY class
>
> Errors related to those profiles are okay, but any errors related to profiles that start with the two-character cell ID you chose will be a problem when it comes time to start this server. Therefore you should correct those errors now.

13.2.4 Stand-alone server customization

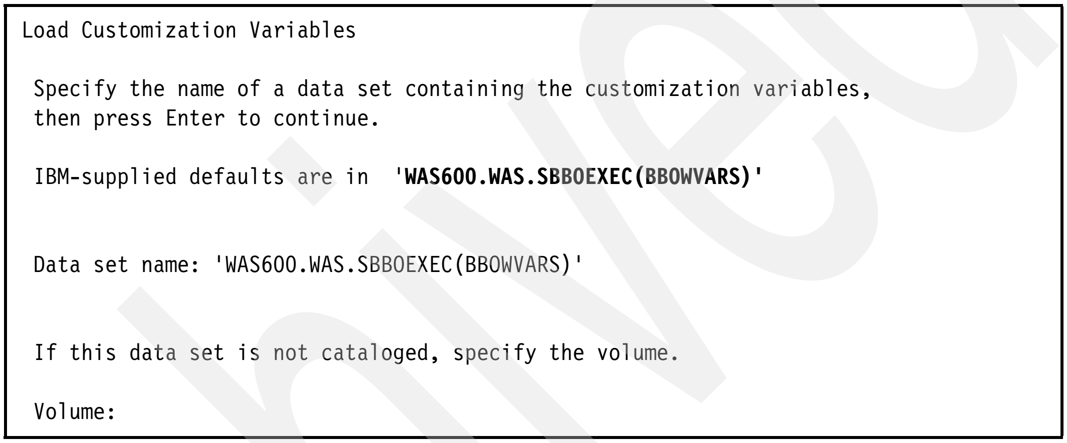
1. Go back to option **6** and re-invoke the ISPF dialogs with the same command you used before:

   ```
   'hlq.SBBOCLIB(BBOWSTRT)' 'APPL(cell_id)'
   ```

2. Selection option **2**, Create standalone application server node.
3. Select option **L**, Load customization variables.
4. In the next dialog, shown in Figure 13-64, verify the following field:

   Make certain the IBM-supplied defaults data set is used for the Data set name. The high-level qualifier would be the same qualifier you specified previously in 13.2.2, “ISPF customization dialogs” on page 278.

   Press Enter.

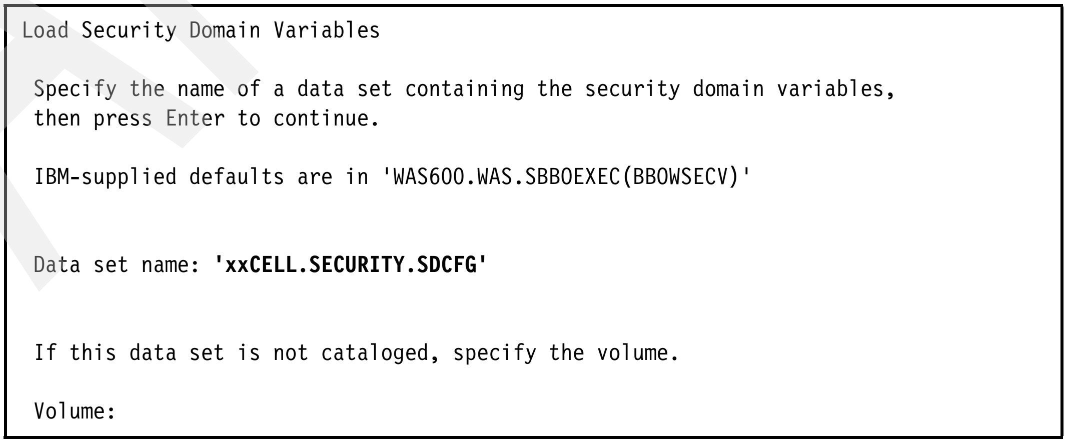


```
Load Customization Variables

 Specify the name of a data set containing the customization variables,
 then press Enter to continue.

 IBM-supplied defaults are in  'WAS600.WAS.SBBOEXEC(BBOWVARS)'


 Data set name: 'WAS600.WAS.SBBOEXEC(BBOWVARS)'


 If this data set is not cataloged, specify the volume.

 Volume:
```

Figure 13-64 Loading customization variables

5. Select option **1**, Load security domain variables.
6. In the next dialog, shown in Figure 13-65, specify the name of the Save variables file you created for the Security domain variables in Figure 13-61 on page 283 and press Enter.

   This is a very important step. The security domain variables connect the RACF IDs for this stand-alone server to the cell config group created during the security domain customization.

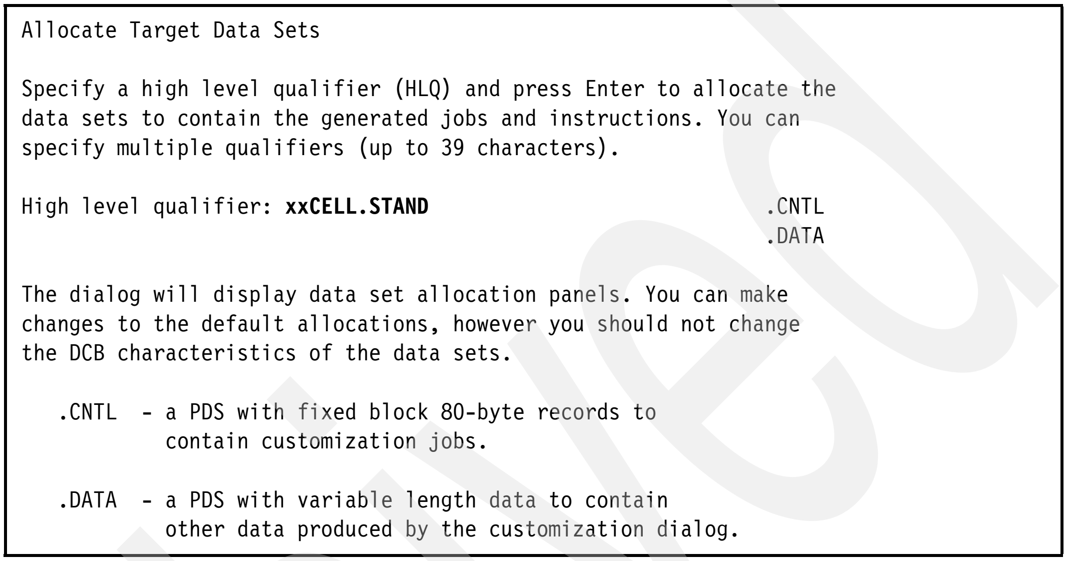


```
Load Security Domain Variables

 Specify the name of a data set containing the security domain variables,
 then press Enter to continue.

 IBM-supplied defaults are in 'WAS600.WAS.SBBOEXEC(BBOWSECV)'


 Data set name: 'xxCELL.SECURITY.SDCFG'


 If this data set is not cataloged, specify the volume.

 Volume:
```

Figure 13-65 Loading security domain variables

7. Select option **2**, Allocate target data sets.
8. In the next dialog, shown in Figure 13-66, specify the high-level qualifier for the target data sets of xxCell.STAND, where xx is your two-character cell identifier. Press Enter.

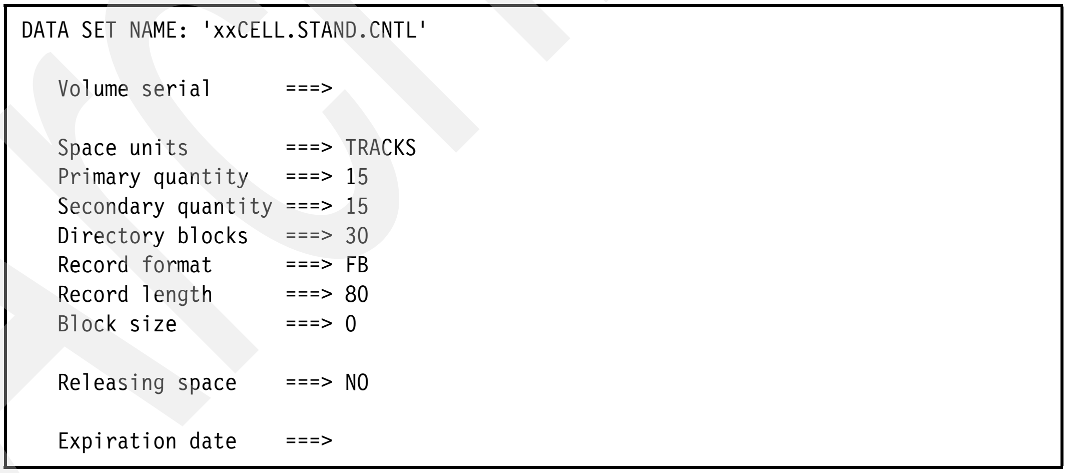
   This results in two data sets being created. The first data set ends in CNTL and the second ends in DATA. The customized jobs are written to the target data sets when you generate the job.



```
Allocate Target Data Sets

Specify a high level qualifier (HLQ) and press Enter to allocate the
data sets to contain the generated jobs and instructions. You can
specify multiple qualifiers (up to 39 characters).

High level qualifier: xxCELL.STAND                          .CNTL
                                                            .DATA

The dialog will display data set allocation panels. You can make
changes to the default allocations, however you should not change
the DCB characteristics of the data sets.

   .CNTL  - a PDS with fixed block 80-byte records to
            contain customization jobs.

   .DATA  - a PDS with variable length data to contain
            other data produced by the customization dialog.
```

Figure 13-66 Allocating target data set

9. The next dialog, shown in Figure 13-67, asks for allocation parameters for the **.CNTL data set. Accept the default values and press Enter.



```
DATA SET NAME: 'xxCELL.STAND.CNTL'

   Volume serial      ===>

   Space units        ===> TRACKS
   Primary quantity   ===> 15
   Secondary quantity ===> 15
   Directory blocks   ===> 30
   Record format      ===> FB
   Record length      ===> 80
   Block size         ===> 0

   Releasing space    ===> NO

   Expiration date    ===>
```

Figure 13-67 Allocating data set for server customization

10. In the next screen, accept the defaults for the **.DATA data set.
11. You should be back at the Stand-alone Server main option panel. Select option **2**, Define variables.
12. You should now see a menu with 5 options. Select option **1**, System locations.

13. You should see the next dialog as shown in Figure 13-68. Do the following:

a. The System name and Sysplex name fields should be filled in for you, based on what the dialogs pick up from your system. Check them to be sure. This is why logging on to the MVS image where the server will run is recommended; it ensures the correct system name is populated for you.

b. Refer back to Table 13-7 on page 277 for information about your PROCLIB, PARMLIB and SYSEXEC data sets. Provide your values in these fields.

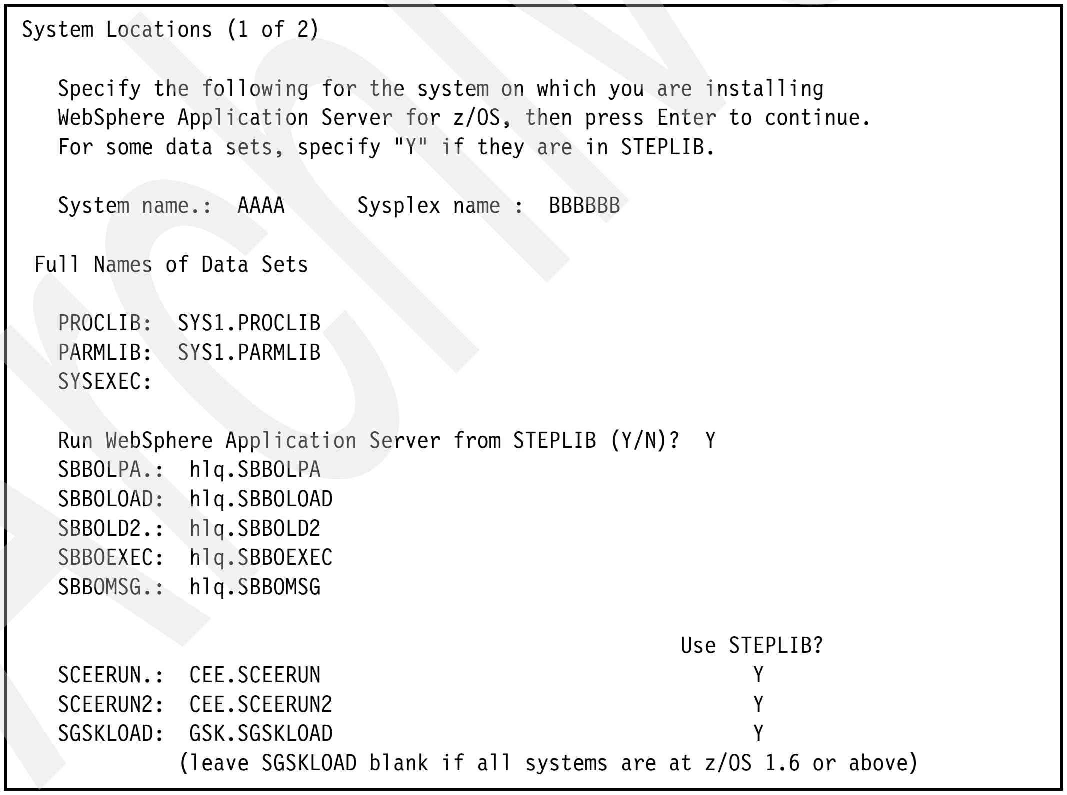
c. Specify `Y` for the STEPLIB question so the WebSphere for z/OS data sets are run from STEPLIB. Even if the data sets are in LPA/LNKLST, specifying `Y` here for this configuration provides the best chance for success.

d. The high-level qualifier values should be filled in properly based on the HLQ you used to invoke the ISPF dialogs. Check them to be sure by referring to Table 13-6 on page 276.

e. Refer to Table 13-7 on page 277 for information about your SCEERUN, SCEERUN2 and SGSKLOAD data sets. Provide the proper values for your system here. If you are on z/OS 1.6 or above, you can leave the SGSKLOAD field blank.

f. Specify `Y` for the STEPLIB question unless you are absolutely certain they reside in LPA/LNKLST. (Even if they do reside in LPA/LNKLST, specifying `Y` here will still allow things to work.)

g. Press Enter to move to the next panel.



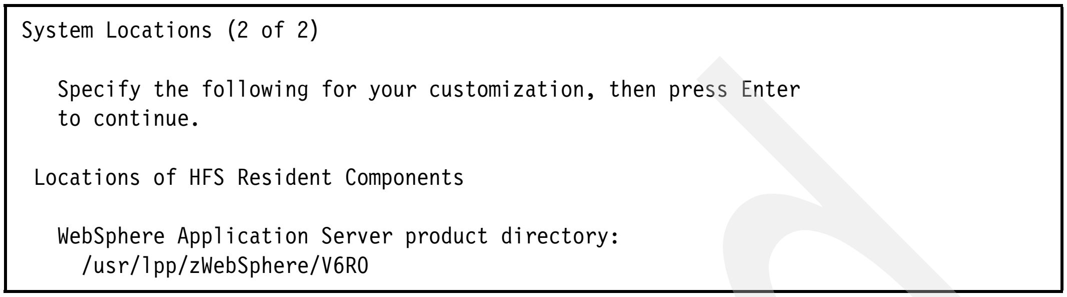
```
System Locations (1 of 2)

   Specify the following for the system on which you are installing
   WebSphere Application Server for z/OS, then press Enter to continue.
   For some data sets, specify "Y" if they are in STEPLIB.

   System name.:  AAAA      Sysplex name :  BBBBBB

 Full Names of Data Sets

   PROCLIB:  SYS1.PROCLIB
   PARMLIB:  SYS1.PARMLIB
   SYSEXEC:

   Run WebSphere Application Server from STEPLIB (Y/N)?  Y
   SBBOLPA.:  hlq.SBBOLPA
   SBBOLOAD:  hlq.SBBOLOAD
   SBBOLD2.:  hlq.SBBOLD2
   SBBOEXEC:  hlq.SBBOEXEC
   SBBOMSG.:  hlq.SBBOMSG

                                                       Use STEPLIB?
   SCEERUN.:  CEE.SCEERUN                                    Y
   SCEERUN2:  CEE.SCEERUN2                                   Y
   SGSKLOAD:  GSK.SGSKLOAD                                   Y
             (leave SGSKLOAD blank if all systems are at z/OS 1.6 or above)
```

Figure 13-68 System locations (Screen 1)

14. For the dialog shown in Figure 13-69 (and referring to Table 13-6 on page 276), provide the mount point where the WebSphere for z/OS hlq.SBBOHFS data set is mounted. The default is /usr/lpp/zWebSphere/V6R0, which may be where your **.SBBOHFS data set is

mounted. Verify that the value provided here is correct. Pay attention to the correct case because the mount point is case sensitive.

Press Enter.

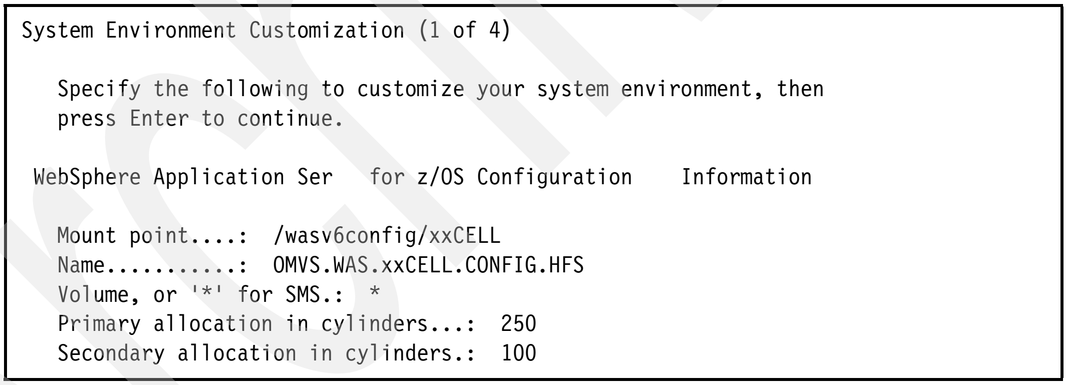


```
System Locations (2 of 2)

   Specify the following for your customization, then press Enter
   to continue.

 Locations of HFS Resident Components

   WebSphere Application Server product directory:
     /usr/lpp/zWebSphere/V6R0
```

Figure 13-69 System locations (Screen 2)

15. You should be back at the Define Variables option panel.
16. Select option **2**, System environment customization.
17. For the next dialog, provide the following information, which is used by the BBOWCHFS job to allocate and mount your configuration HFS:
    - A mount point of /wasv6config/xxCell where `xx` is your two-character cell identifier.
    - An HFS data set name of OMVS.WAS.xxCELL.CONFIG.HFS where `xx` is your two-character cell identifier.

    Press Enter.

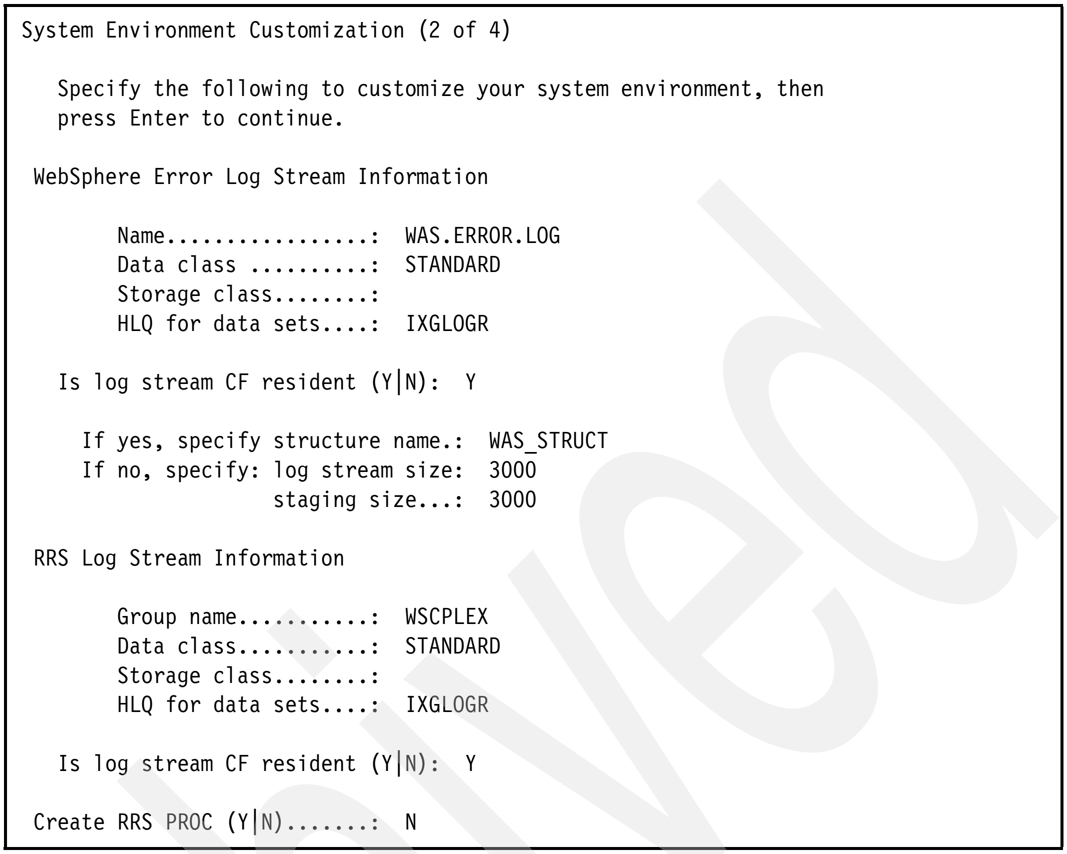


```
System Environment Customization (1 of 4)

   Specify the following to customize your system environment, then
   press Enter to continue.

 WebSphere Application Ser   for z/OS Configuration    Information

   Mount point....:  /wasv6config/xxCELL
   Name..........:  OMVS.WAS.xxCELL.CONFIG.HFS
   Volume, or '*' for SMS.:  *
   Primary allocation in cylinders...:  250
   Secondary allocation in cylinders.:  100
```

Figure 13-70 System environment customization (Screen 1)

18. For the next dialog, shown in Figure 13-71, change the RRS procedure value to `N`. We are assuming you are already using an RRS procedure, so this is one less member that the BBOWCPY1 job must copy into PROCLIB.

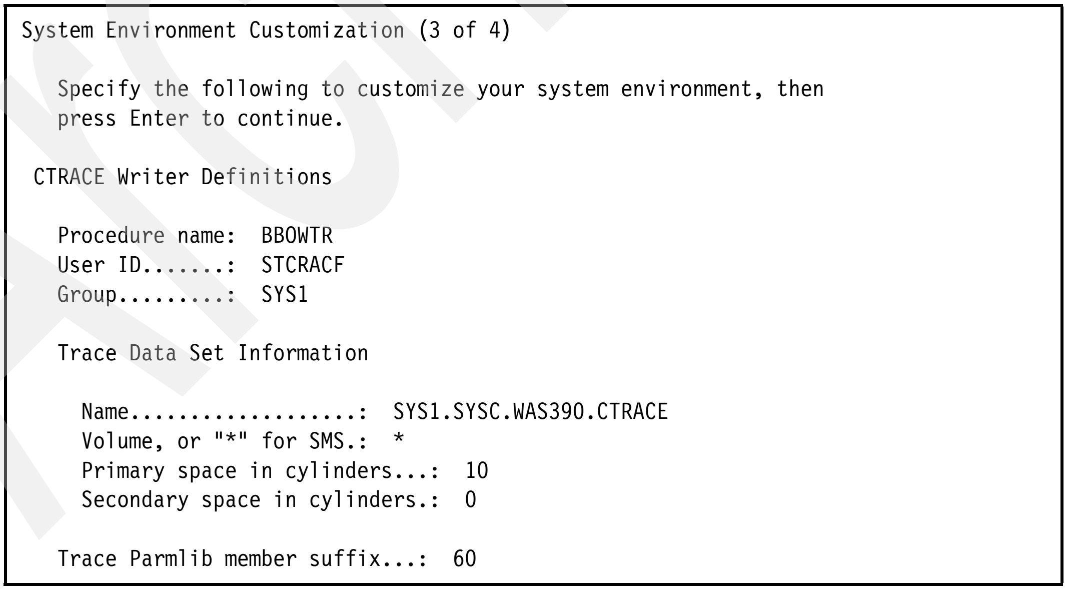
    Press Enter.



```
System Environment Customization (2 of 4)

   Specify the following to customize your system environment, then
   press Enter to continue.

 WebSphere Error Log Stream Information

        Name.................:  WAS.ERROR.LOG
        Data class ..........:  STANDARD
        Storage class........:
        HLQ for data sets....:  IXGLOGR

   Is log stream CF resident (Y|N):  Y

     If yes, specify structure name.:  WAS_STRUCT
     If no, specify: log stream size:  3000
                     staging size...:  3000

 RRS Log Stream Information

        Group name...........:  WSCPLEX
        Data class...........:  STANDARD
        Storage class........:
        HLQ for data sets....:  IXGLOGR

   Is log stream CF resident (Y|N):  Y

 Create RRS PROC (Y|N).......:  N
```

Figure 13-71 System environment customization (Screen 2)

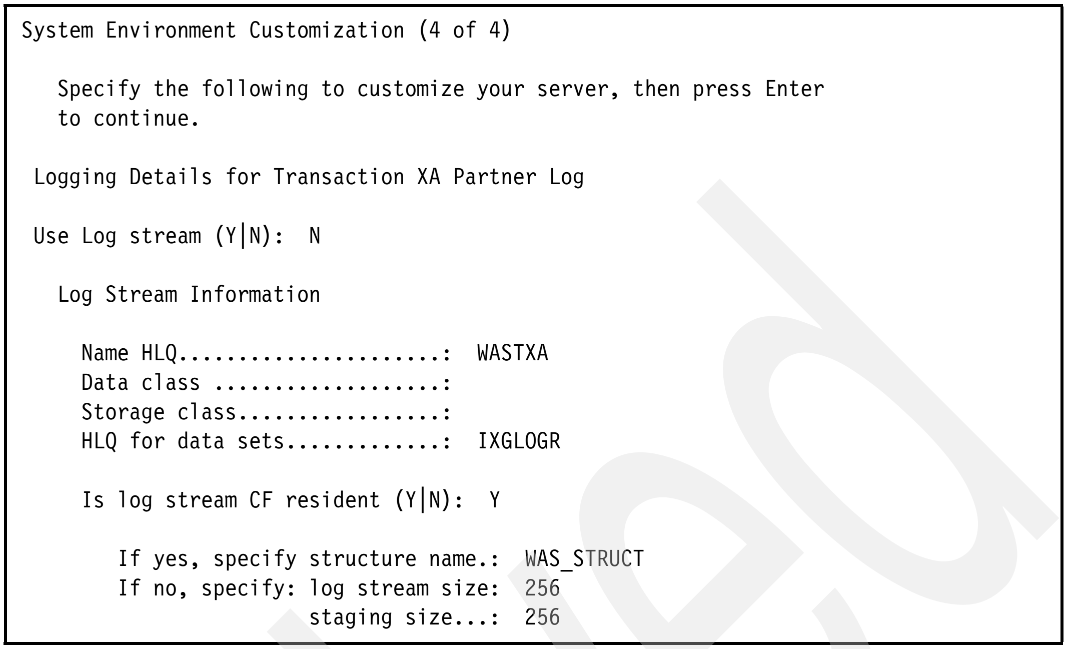
19. For the next dialog, shown in Figure 13-72, accept the defaults and press Enter.



```
System Environment Customization (3 of 4)

   Specify the following to customize your system environment, then
   press Enter to continue.

 CTRACE Writer Definitions

   Procedure name:  BBOWTR
   User ID.......:  STCRACF
   Group.........:  SYS1

   Trace Data Set Information

     Name...................:  SYS1.SYSC.WAS390.CTRACE
     Volume, or "*" for SMS.:  *
     Primary space in cylinders...:  10
     Secondary space in cylinders.:  0

   Trace Parmlib member suffix...:  60
```

Figure 13-72 System environment customization (Screen 3)

20. For the next dialog, shown in Figure 13-73, accept the defaults and press Enter.

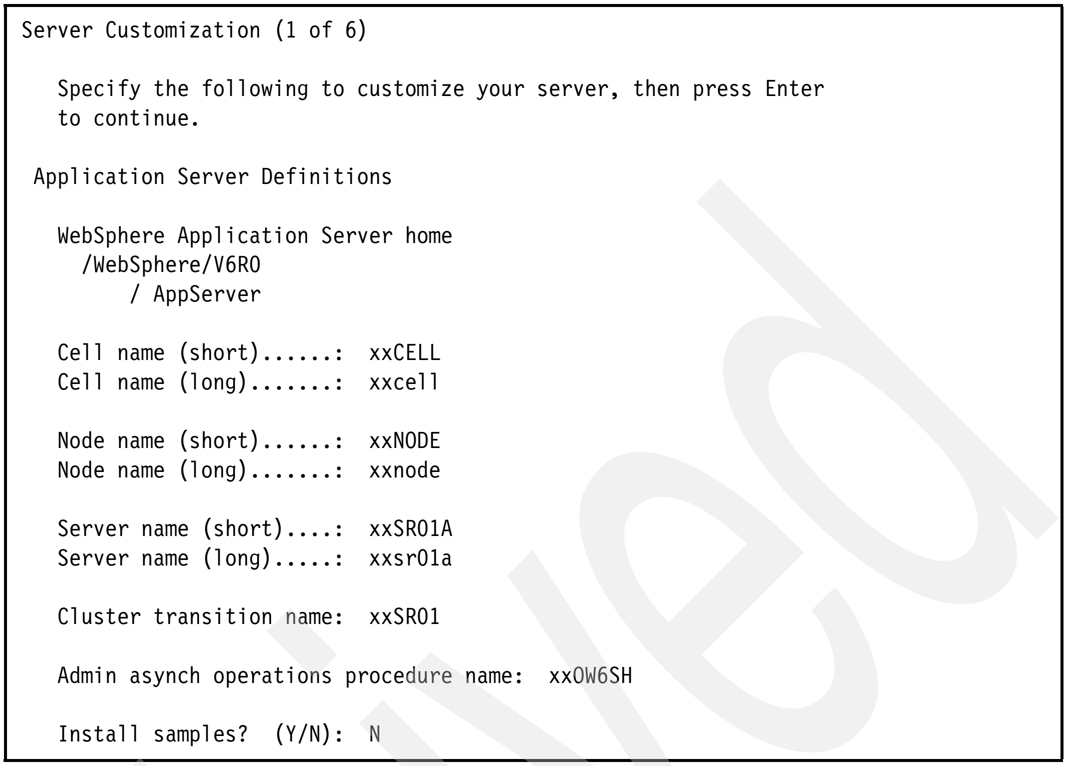


```
System Environment Customization (4 of 4)

   Specify the following to customize your server, then press Enter
   to continue.

 Logging Details for Transaction XA Partner Log

 Use Log stream (Y|N):  N

   Log Stream Information

     Name HLQ......................:  WASTXA
     Data class ...................:
     Storage class.................:
     HLQ for data sets.............:  IXGLOGR

     Is log stream CF resident (Y|N):  Y

        If yes, specify structure name.:  WAS_STRUCT
        If no, specify: log stream size:  256
                        staging size...:  256
```

Figure 13-73 System environment customization (Screen 4)

21. You should be back at the Define Variables option panel.
22. Select option **3,** Server customization.
23. For the next dialog, shown in Figure 13-74, do the following:
    a. Accept the default value for the WebSphere Application Server home field.
    b. Use the values you specified in Figure 13-50 on page 277 for cell name, node name and server name.
    c. Specify `xxOW6SH` in the Admin asynch operations procedure name field, where `xx` is your two-character cell identifier.
    d. For the Install samples field, specify `N`.

    Press Enter.

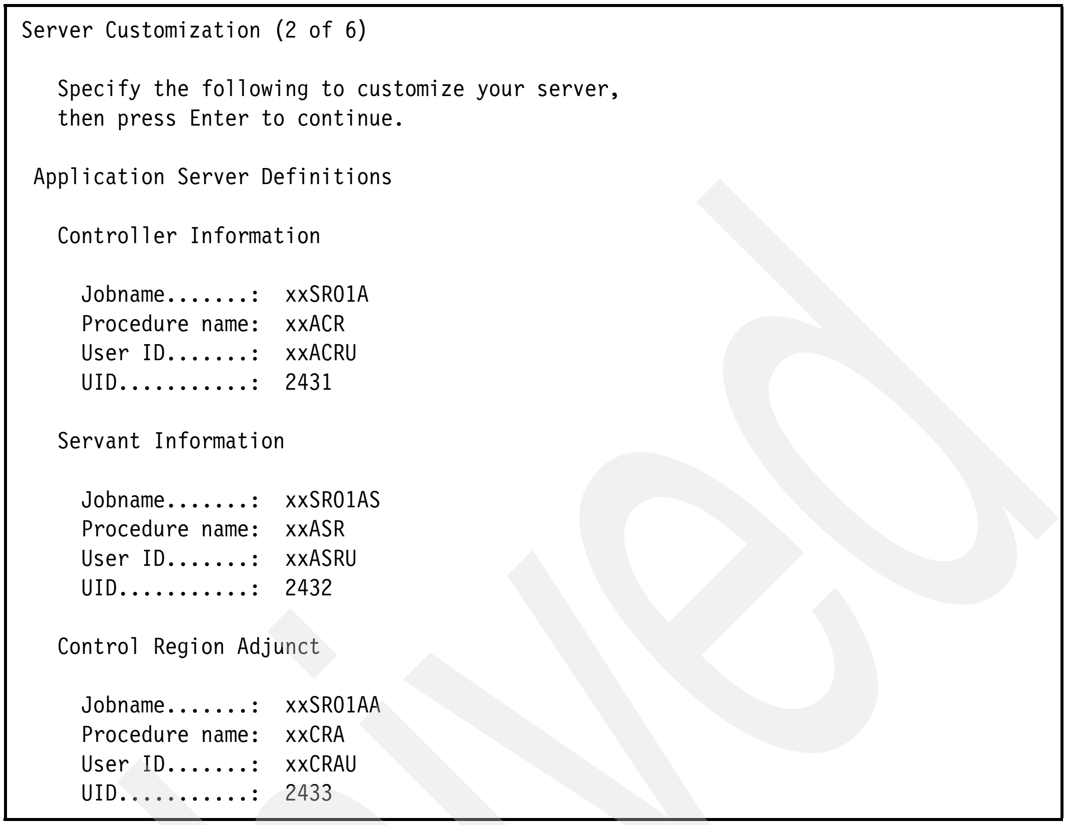


```
Server Customization (1 of 6)

   Specify the following to customize your server, then press Enter
   to continue.

 Application Server Definitions

   WebSphere Application Server home
     /WebSphere/V6R0
         / AppServer

   Cell name (short)......:  xxCELL
   Cell name (long).......:  xxcell

   Node name (short)......:  xxNODE
   Node name (long).......:  xxnode

   Server name (short)....:  xxSR01A
   Server name (long).....:  xxsr01a

   Cluster transition name:  xxSR01

   Admin asynch operations procedure name:  xxOW6SH

   Install samples?  (Y/N):  N
```

Figure 13-74   Server customization (Screen 1)

24. For the next dialog, shown in Figure 13-75, do the following:
    a. Provide the procedure name values that you specified in Figure 13-51 on page 278.
    b. Set the User ID value to the procedure name appended with `U`.
    c. Ensure the UID values are not already assigned in RACF.

    Press Enter.

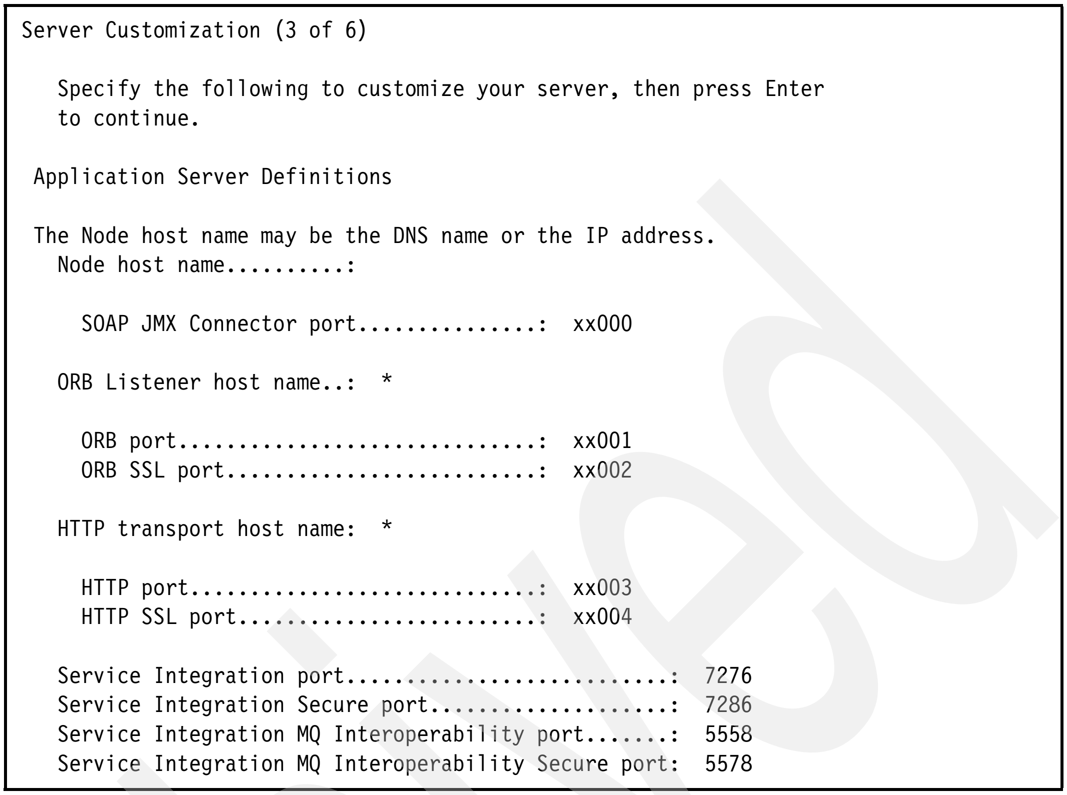


```
Server Customization (2 of 6)

   Specify the following to customize your server,
   then press Enter to continue.

 Application Server Definitions

   Controller Information

     Jobname.......:  xxSR01A
     Procedure name:  xxACR
     User ID.......:  xxACRU
     UID...........:  2431

   Servant Information

     Jobname.......:  xxSR01AS
     Procedure name:  xxASR
     User ID.......:  xxASRU
     UID...........:  2432

   Control Region Adjunct

     Jobname.......:  xxSR01AA
     Procedure name:  xxCRA
     User ID.......:  xxCRAU
     UID...........:  2433
```

Figure 13-75 Server customization (Screen 2)

25. For panel 3, shown in Figure 13-76, do the following:
    a. Provide the hostname as specified in Table 13-7 on page 277.
    b. Leave the ORB and HTTP host name values at the default value.
    c. Provide the port values you specified in Figure 13-52 on page 278.

    Press Enter.



```
Server Customization (3 of 6)

   Specify the following to customize your server, then press Enter
   to continue.

 Application Server Definitions

 The Node host name may be the DNS name or the IP address.
   Node host name..........:

     SOAP JMX Connector port...............:  xx000

   ORB Listener host name..:  *

     ORB port..............................:  xx001
     ORB SSL port..........................:  xx002

   HTTP transport host name:  *

     HTTP port.............................:  xx003
     HTTP SSL port.........................:  xx004

   Service Integration port...........................:  7276
   Service Integration Secure port....................:  7286
   Service Integration MQ Interoperability port.......:  5558
   Service Integration MQ Interoperability Secure port:  5578
```

Figure 13-76 Server customization (Screen 3)

Note: Figure 13-76 may look slightly different depending on your service level. If you have service level 6.0.2.1 or later, the dialog ends with five port fields beginning with High Availability Manager.

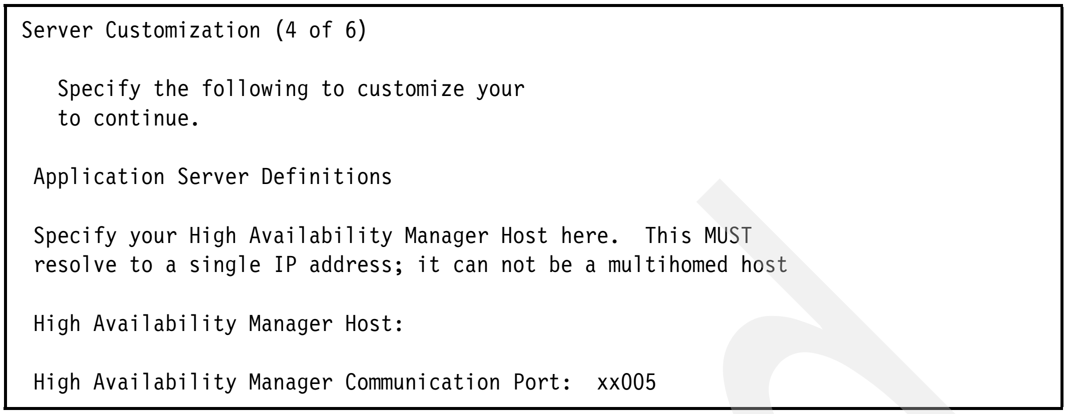
26. For the next dialog, shown in Figure 13-77, refer to Figure 13-52 on page 278, then specify the following:

    – The IP address of the High Availability Manager. This value must be a IP address in dotted decimal format.

    – The port value that you specified in Figure 13-52 on page 278.

    Press Enter.

Note: Figure 13-77 is not presented if you have service level 6.0.1.2 or later. In that case, Figure 13-76 would have the High Availability Manager port number and you would specify the port number there. The High Availability Manager host name is no longer needed with 6.0.1.2.



```
Server Customization (4 of 6)

   Specify the following to customize your
   to continue.

 Application Server Definitions

 Specify your High Availability Manager Host here.  This MUST
 resolve to a single IP address; it can not be a multihomed host

 High Availability Manager Host:

 High Availability Manager Communication Port:  xx005
```

Figure 13-77 Server customization (Screen 4)

27. For the next dialog, shown in Figure 13-78, do the following:

   a. Accept the default value for Daemon home directory.

   b. Provide the Daemon job name and procedure name according to Figure 13-51 on page 278.

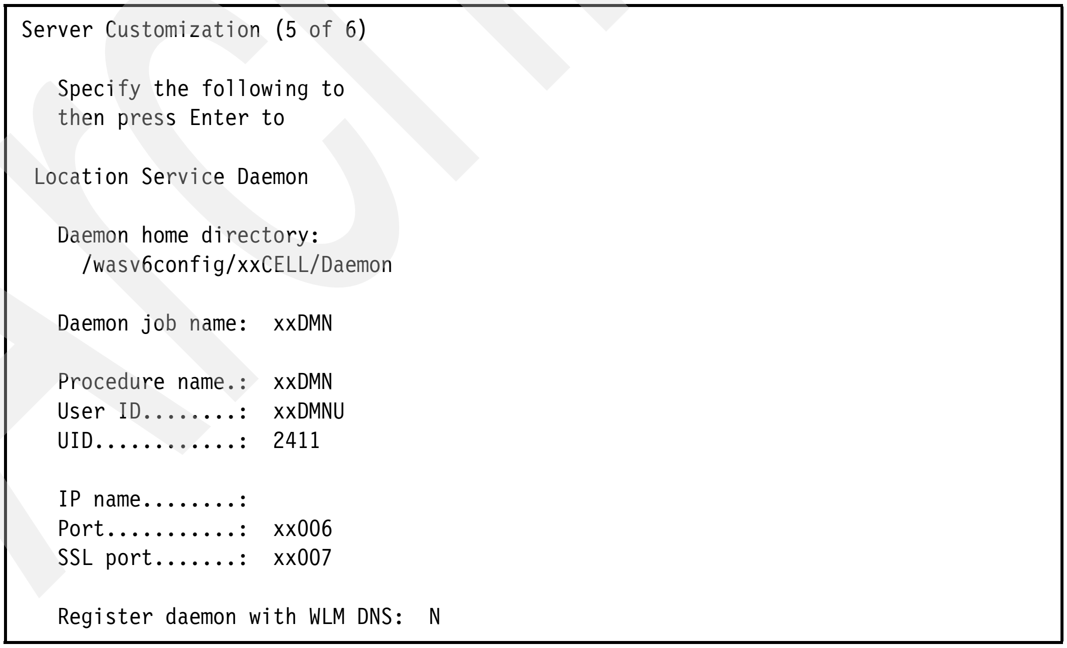
   c. Set the User ID values to the procedure name appended with `U`.

   d. Ensure the UID value is OK on your system.

   e. Provide DNS host name from Table 13-7 on page 277 for the IP name field. An IP address does not work here.

   f. Provide the daemon port values you specified in Figure 13-52 on page 278.

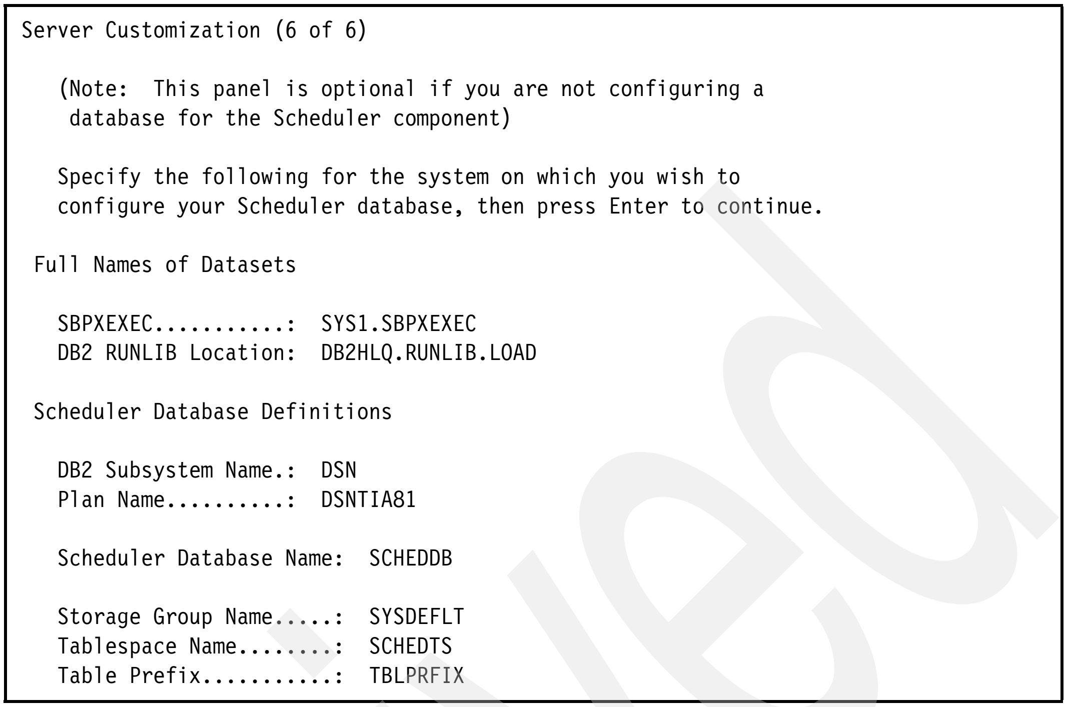
   Press Enter.



```
Server Customization (5 of 6)

   Specify the following to
   then press Enter to

 Location Service Daemon

   Daemon home directory:
     /wasv6config/xxCELL/Daemon

   Daemon job name:  xxDMN

   Procedure name.:  xxDMN
   User ID........:  xxDMNU
   UID............:  2411

   IP name........:
   Port...........:  xx006
   SSL port.......:  xx007

   Register daemon with WLM DNS:  N
```

Figure 13-78 Server customization (Screen 5)

28. For the next dialog, shown in Figure 13-79, accept the defaults and press Enter.

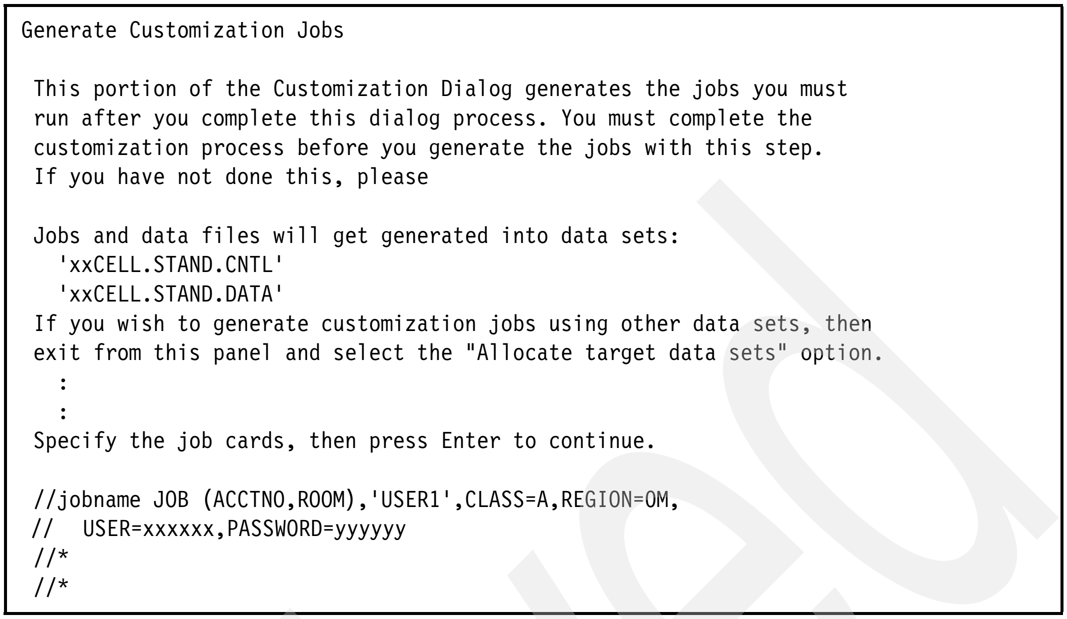


```
Server Customization (6 of 6)

   (Note:  This panel is optional if you are not configuring a
    database for the Scheduler component)

   Specify the following for the system on which you wish to
   configure your Scheduler database, then press Enter to continue.

 Full Names of Datasets

   SBPXEXEC...........:  SYS1.SBPXEXEC
   DB2 RUNLIB Location:  DB2HLQ.RUNLIB.LOAD

 Scheduler Database Definitions

   DB2 Subsystem Name.:  DSN
   Plan Name..........:  DSNTIA81

   Scheduler Database Name:  SCHEDDB

   Storage Group Name.....:  SYSDEFLT
   Tablespace Name........:  SCHEDTS
   Table Prefix...........:  TBLPRFIX
```

Figure 13-79 Server customization (Screen 6)

29. You should be back at the Define Variables option dialog.
30. Skip the options to configure a web server and to view security domain variables.
31. Press F3 to return to the Standalone Server primary option menu.
32. Select option **4**, Generate customized jobs.
33. On the next dialog, shown in Figure 13-80, do the following:
    a. Verify the correct target data sets are listed.
    b. Add a comma to the end of the job card, then add a USER and PASSWORD so the customized jobs all run under the UID=0 ID. Specifying a USER= and PASSWORD= on the JOB card is not strictly required, provided you ensure the generated jobs run under the UID=0 ID.

    Press Enter.

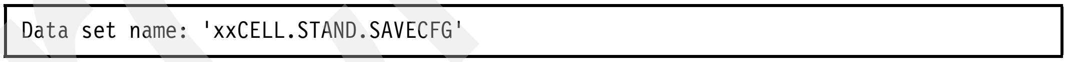


```
Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please

 Jobs and data files will get generated into data sets:
   'xxCELL.STAND.CNTL'
   'xxCELL.STAND.DATA'
 If you wish to generate customization jobs using other data sets, then
 exit from this panel and select the "Allocate target data sets" option.
   :
   :
 Specify the job cards, then press Enter to continue.

 //jobname JOB (ACCTNO,ROOM),'USER1',CLASS=A,REGION=0M,
 //   USER=xxxxxx,PASSWORD=yyyyyy
 //*
 //*
```

Figure 13-80 Generate customization jobs

34. Watch as the members are written out to the target data sets. When that process is complete, press Enter to return to the Standalone Server primary option menu.
35. Select option **S**, Save customization variables.
36. On the next dialog, shown in Figure 13-81, provide a name for the variable data set:
    a. The high-level qualifier should be xxCELL.STAND, where xx is your two character cell identifier.
    b. The last qualifier should be SAVECFG.

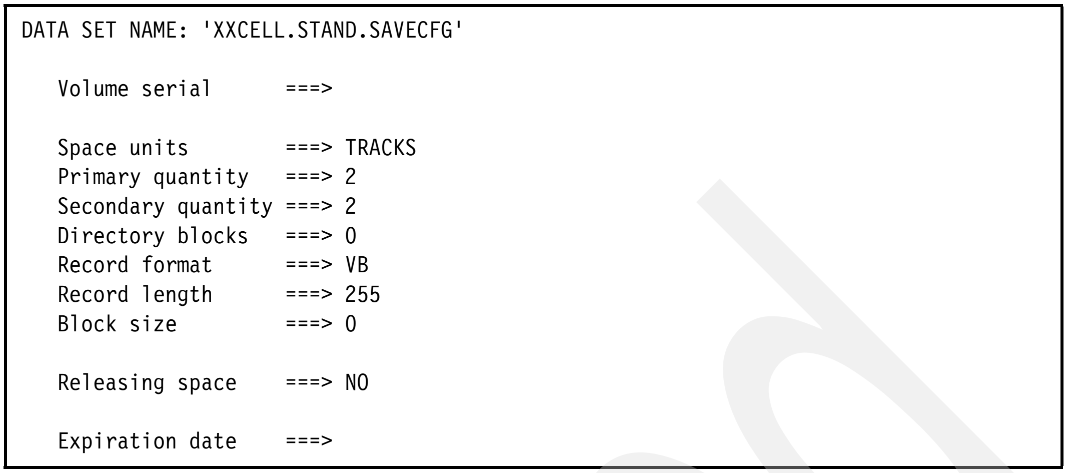
    Press Enter.



```
Data set name: 'xxCELL.STAND.SAVECFG'
```

Figure 13-81 Specify data set name

37. In the next dialog, shown in Figure 13-82, accept the default allocation parameters for the data set and press Enter.



```
DATA SET NAME: 'XXCELL.STAND.SAVECFG'

   Volume serial      ===>

   Space units        ===> TRACKS
   Primary quantity   ===> 2
   Secondary quantity ===> 2
   Directory blocks   ===> 0
   Record format      ===> VB
   Record length      ===> 255
   Block size         ===> 0

   Releasing space    ===> NO

   Expiration date    ===>
```

Figure 13-82 Data set allocation parameters

38. You should be at the Standalone Server main option panel.
39. Press F3 to return to the Primary option menu.
40. Press F3 to exit out of the ISPF customization dialogs.

13.2.5 Run customized jobs, start server

The xxCELL.STAND.CNTL data set is full of members, but not every one is a JCL job, and not every JCL job needs to be run. The BBOSSINS member contains a list of instructions if you are curious about what jobs do exist.

You should run the jobs shown in Table 13-9 in the order shown. Check each as it completes. Run these jobs on the same MVS image where you intend the WebSphere stand-alone server to reside. That applies regardless of any shared HFS arrangements you may have.

Table 13-9 List of customization jobs

Check when completed	Job name	Job description
	BBOCBRAJ	Creates a script full of RACF commands that is invoked by the BBOCBRAK job. This job just creates the script. The next job actually drives RACF and creates the IDs, groups.
	BBOCBRAK	Uses the script created by BBOCBRAJ to create the RACF objects needed to run your server. A RC=0 does not necessarily mean things ran okay. Read through the job listing.
	BBOWCHFS	Allocates the HFS file system data set and mounts it at the mount point you specified.
	BBOWHFSA	Creates the directory structure inside the HFS just created.
	BBOWCPY1	Copies tailored members to PROCLIB, PARMLIB and SYSEXEC. Take a look at the job itself to see what members are copied.

Check when completed	Job name	Job description
	BBOWWPFA	Copies a bunch of shell scripts, XML files and trace data into the WebSphere HFS directory structure. Creates the was.env from the information stored in many different places throughout the Standalone Server's HFS. Batch-installs the administrative application into your server. This job will run at least 20 minutes.
	BBOWHFSB	Finalizes the HFS. Very important! Do not overlook this job

Note: There is a chance that the BBOCBRAK will encounter profiles that already exist in your RACF database, depending on whether any WebSphere V4 or V5 work had been done before. In particular, you may see the following profiles flagged as already created:

- STCRACF User ID
- WAS.ERROR.LOG
- Any CBIND or SERVER class profiles with the string BBO somewhere in them
- BPX.WLMSERVER
- A STARTED profile of BBO*.*

Errors related to those profiles are okay. Any errors related to profiles that start with the two-character cell ID you chose will be a problem when it comes time to start this server.

Use ISHELL and go to your configuration mount point /wasv6config/xxCELL, where xx is your two-character cell identifier. You should see the details shown in Figure 13-83.

```
Type  Perm  Owner      Filename
Dir    775  xxADMIN    .
Dir    775  xxADMIN    ..
Dir    775  xxADMIN    AppServer
Dir    775  xxADMIN    Daemon
Syml   777  QWER01     xxCELL.xxCELL.xxDMN
Syml   777  QWER01     xxCELL.xxNODE.xxSR01A
Syml   777  QWER01     xxCELL.xxNODE.xxSR01A.HOME
```

Figure 13-83 Directory and symbolic links under mount point

13.2.6 Start the server

Issue the following MVS start command, where xx is your two-character cell identifier:

```
S xxACR,JOBNAME=xxSR01A,ENV=xxCELL.xxNODE.xxSR01A
```

where

- xxACR is the controller JCL start procedure.
- JOBNAME xxSR01A should match the server shortname.
- the ENV= value matches the symbolic link shown in Figure 13-83.

You should see this sequence of events:

1. The controller (xxSR01A) starts, but pauses for a bit.
2. The controller starts the daemon server (xxDMN).

3. When the daemon finishes initializing, the controller continues until it initializes.
4. When the controller is up, WLM automatically starts a servant region (xxSR01AS).

Look for the following in the servant SYSPRINT:

```
WSVR0001I: Server SERVANT PROCESS xxsr01a open for e-business
```

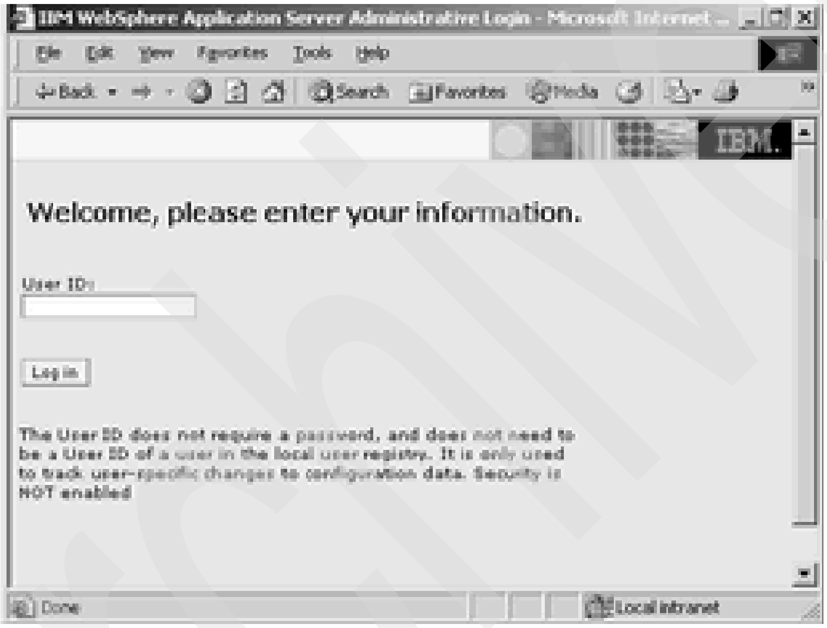
13.2.7 Access the administrative console

Point your browser at:

```
http://<your_host_name>:yy003/ibm/console
```

where yy is the two-digit port prefix you chose to use.

You should get a log in panel that looks like Figure 13-84.


IBM WebSphere Application Server Administrative Login - Microsoft Internet ...
File Edit View Favorites Tools Help
Back Search Favorites Media
IBM
Done
Local intranet

Welcome, please enter your information.

User ID:

Log in

The User ID does not require a password, and does not need to be a User ID of a user in the local user registry. It is only used to track user-specific changes to configuration data. Security is NOT enabled

Figure 13-84 Administrative console

Supply any name you want for the User ID. Global security is not yet turned on for this cell, so there is no authentication performed.

13.2.8 Enable global security

We know that having global security turned on is something most people want to do to test and evaluate the WebSphere V6 for z/OS runtime environment.

In the navigation bar on the left side of the administrative console, expand the Security folder and click Global Security.

As illustrated in Figure 13-85, do the following:

1. Select LTPA.
2. Specify any password you choose. The value you supply is not tied to any of the IDs you created earlier. This is just a string of characters used to password-protect the LTPA keys that are generated. As a means of relating them in some memorable way to your cell, you

might want to use the Admin ID password that was created back in the Security Domain section.

3. Click OK.

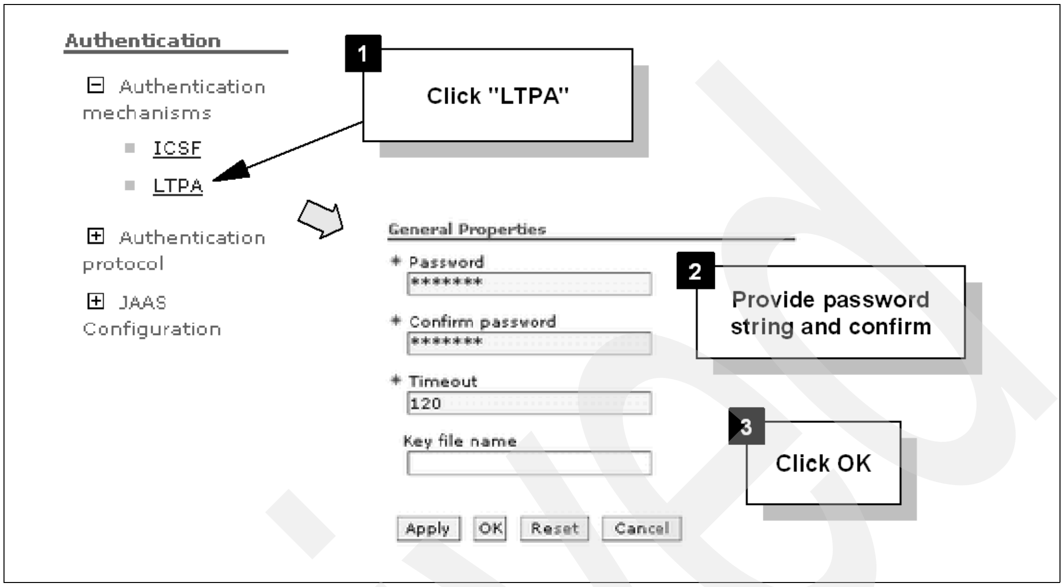
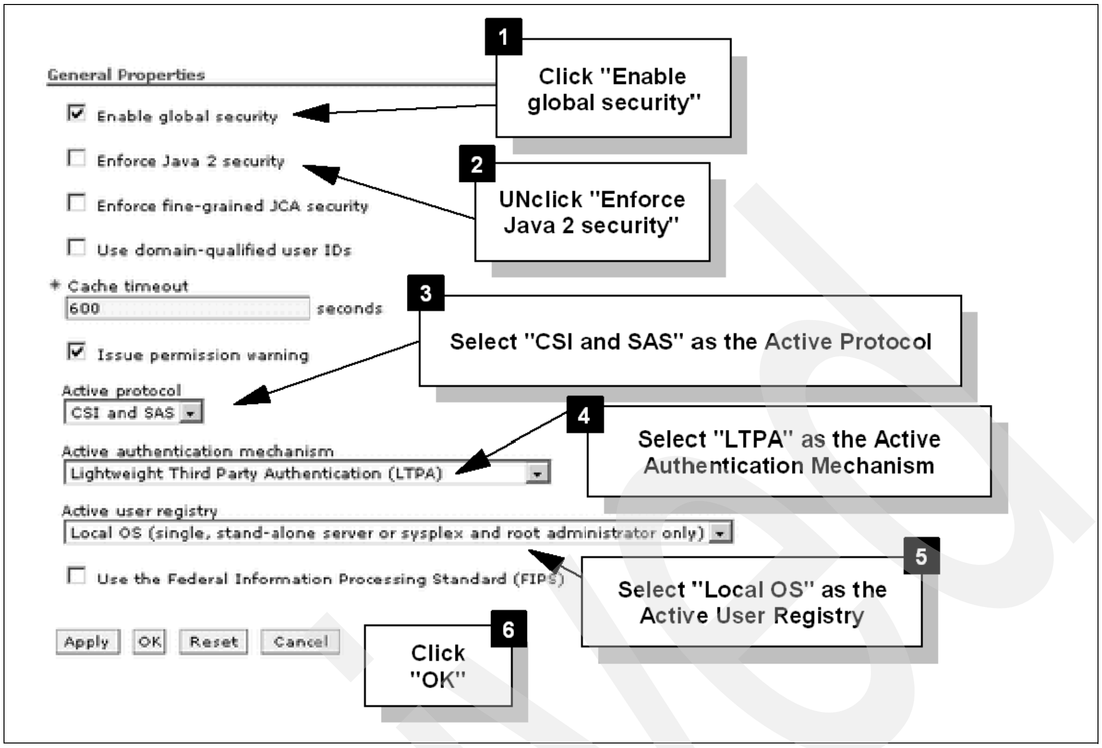


Figure 13-85 Configuring LTPA security

You should be back at the General Properties screen as shown in Figure 13-86. Do the following:

1. Select Enable global security
2. De-select Enforce Java 2 security
3. Select CSI and SAS as the active protocol.
4. Select LTPA as the active authentication mechanism.
5. Select Local OS as the active user registry.
6. Click OK.



General Properties

☑ Enable global security

□ Enforce Java 2 security

□ Enforce fine-grained JCA security

□ Use domain-qualified user IDs

* Cache timeout

600 seconds

☑ Issue permission warning

Active protocol

CSI and SAS

Active authentication mechanism

Lightweight Third Party Authentication (LTPA)

Active user registry

Local OS (single, stand-alone server or sysplex and root administrator only)

□ Use the Federal Information Processing Standard (FIPS)

1 Click "Enable global security"

2 UNclick "Enforce Java 2 security"

3 Select "CSI and SAS" as the Active Protocol

4 Select "LTPA" as the Active Authentication Mechanism

5 Select "Local OS" as the Active User Registry

6 Click "OK"

Figure 13-86 Additional security parameters

You will see a message indicating that changes have been made and need to be saved. Save the configuration by selecting Save twice.

That should take you back to the General Properties screen.

Log out of the administrative console.

Stop and restart server

The changes made to enable global security require a restart of the server. Your configuration has three address spaces: daemon, controller, and servant. At a minimum you need to stop and restart the controller and servant.

Stopping the daemon ensures the controller and the servant come down as well.

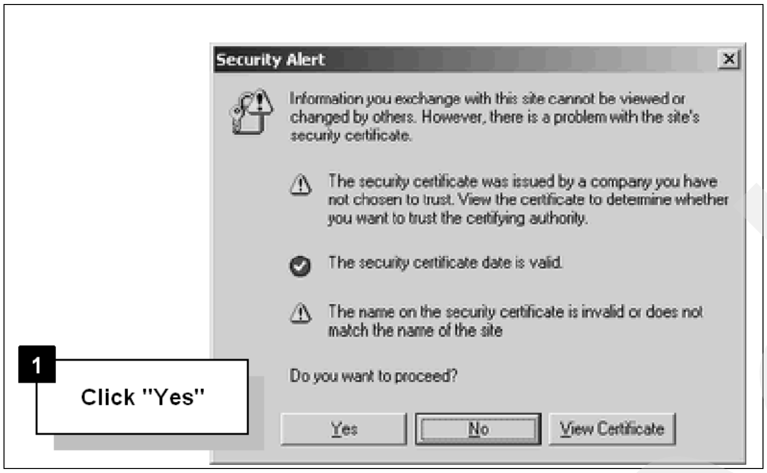
Issue an MVS STOP command against your daemon job.

Watch as all three address spaces come down.

Now issue the MVS start command as described in 13.2.6, “Start the server” on page 298.

Log on using Admin ID and password

When the server finishes initializing, point your browser at the HTTP port. You will get a pop-up window that looks like Figure 13-87.



Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

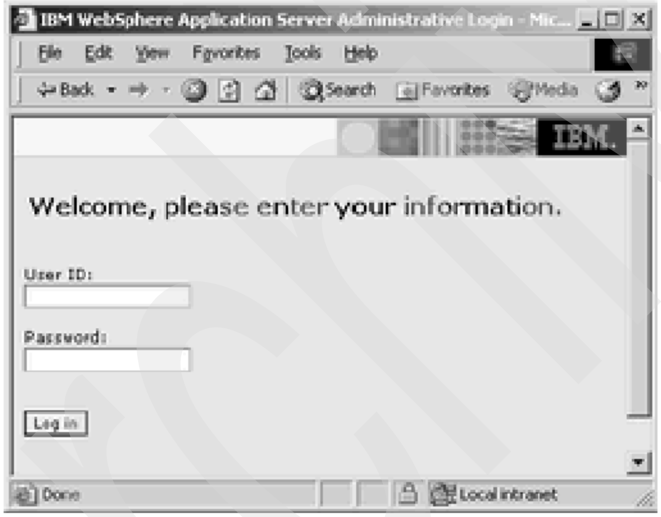
The security certificate date is valid.

The name on the security certificate is invalid or does not match the name of the site

Do you want to proceed?

Figure 13-87 Security alert when logging into secured administrative console

Log on using the Admin ID and password you set during the specification of the Security Domain.



IBM WebSphere Application Server Administrative Login - Mic...

Welcome, please enter your information.

User ID:

Password:

Figure 13-88 Secured administrative console

14

V6.1 updates for distributed platforms

This chapter discusses the changes that affect migration that Version 6.1 has introduced compared to Version 6.0 on the distributed operating platforms.

14.1 Summary of significant changes in V6.1 compared to V6.0

WebSphere Application Server V6.1 introduces many functional enhancements beyond the features of WebSphere Application Server V6.0. However, only a few of these enhancements affect the process of migration to V6.1. The enhancements you should know about if you are contemplating a migration to V6.1 are:

- Automatic migration utilities no longer support WebSphere V4 as an originating system version for the distributed platforms.
- Installation changes

  Both the graphical and command line methods of installation have minor changes you should be aware of. See 14.3, “Installation” on page 305 for more information.
- Administrative console changes

  The graphical administrative console has undergone minor changes in order to make it easier and more intuitive to use. See 14.4, “Administrative console” on page 318 for more information.
- Changes in mixed version cell restrictions

  Some of the restrictions on what can be manipulated in a mixed version cell have been lifted. See 14.5, “Restrictions on mixed version cells” on page 321 for more information.
- Administrator functional deprecations

  See 14.6, “System administration function deprecations and removals” on page 322 for more information.

14.2 Automatic migration utilities

With the addition of a new release, the automatic migration utilities now drop version 4.0 from the list of supported releases from which you can you can migrate. Table 14-1 shows the supported releases.

Table 14-1 Supported WebSphere releases for migration starting points

Version 5.0	Version 5.1	Version 6.0
WebSphere Application Server	WebSphere Application Server	WebSphere Application Server
WebSphere Application Server Express	WebSphere Application Server Express	WebSphere Application Server Express
WebSphere Application Server Network Deployment	WebSphere Application Server Network Deployment	WebSphere Application Server Network Deployment
WebSphere Application Server Enterprise	WebSphere Business Integration Server Foundation	WebSphere Business Integration Server Foundation
WebSphere Application Server for z/OS	WebSphere Application Server for z/OS	WebSphere Application Server for z/OS

14.3 Installation

The basics of installation are similar to those of V6.0, which are discussed in 8.2, “Installation” on page 100. There are differences that you encounter when attempting an installation using the graphical interface.

The graphical installation wizard for WebSphere Application Server Network Deployment gives you the choice to create a default profile during the first installation step. You are presented with the choice of profile type as shown in Figure 14-7 on page 309. If you select None, then no profile is created, just as you are used to for V6.0. If you select one of the other profile types, a default profile is create for you. These profile attributes are chosen for you without any choice at installation time:

- Profile name
- Cell name
- Node name
- Server name
- Port assignments
- Windows service name and authentication attributes

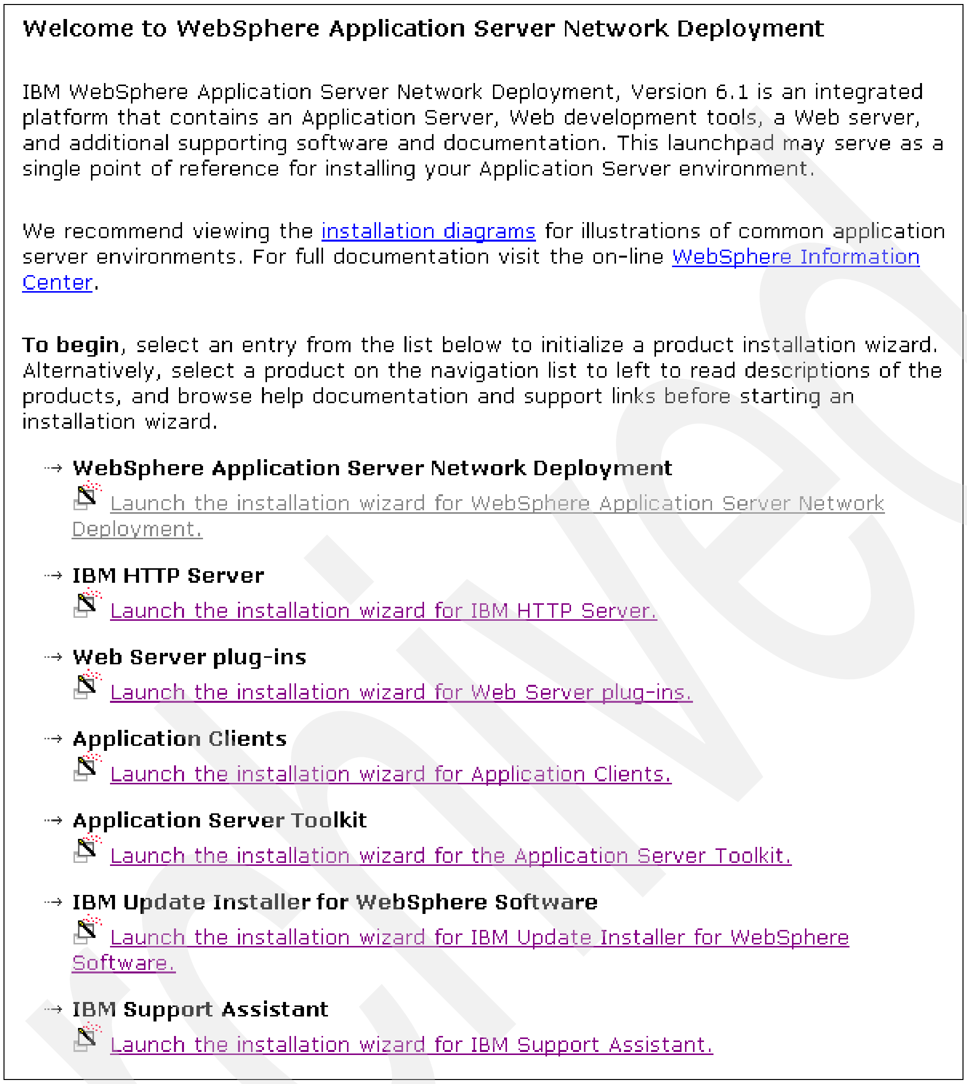
If any of the profile attributes chosen by the installer are not acceptable, then you should create your new profile after installation using the profile management tool. You then have full control over all aspects of profile creation.

14.3.1 Installing product binaries

If you are installing WebSphere Application Server or WebSphere Application Server Express, then the installation wizard works as it does for V6.0. That is, the installation wizard creates a default profile appropriate for an application server. All the attributes are chosen by default. Of course, you have the option to create additional profiles with the profile management tool.

The following is a brief summary of the primary screens you encounter.

Figure 14-1 shows the launchpad screen, which is accessed from the launchpad command found at the root of the installation image.



Welcome to WebSphere Application Server Network Deployment

IBM WebSphere Application Server Network Deployment, Version 6.1 is an integrated platform that contains an Application Server, Web development tools, a Web server, and additional supporting software and documentation. This launchpad may serve as a single point of reference for installing your Application Server environment.

We recommend viewing the installation diagrams for illustrations of common application server environments. For full documentation visit the on-line WebSphere Information Center.

To begin, select an entry from the list below to initialize a product installation wizard. Alternatively, select a product on the navigation list to left to read descriptions of the products, and browse help documentation and support links before starting an installation wizard.

→ **WebSphere Application Server Network Deployment**
Launch the installation wizard for WebSphere Application Server Network Deployment.

→ **IBM HTTP Server**
Launch the installation wizard for IBM HTTP Server.

→ **Web Server plug-ins**
Launch the installation wizard for Web Server plug-ins.

→ **Application Clients**
Launch the installation wizard for Application Clients.

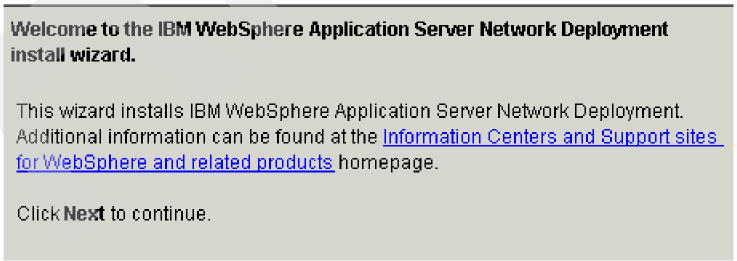
→ **Application Server Toolkit**
Launch the installation wizard for the Application Server Toolkit.

→ **IBM Update Installer for WebSphere Software**
Launch the installation wizard for IBM Update Installer for WebSphere Software.

→ **IBM Support Assistant**
Launch the installation wizard for IBM Support Assistant.

Figure 14-1 Launchpad

Figure 14-2 shows the initial screen after launching from the launchpad.



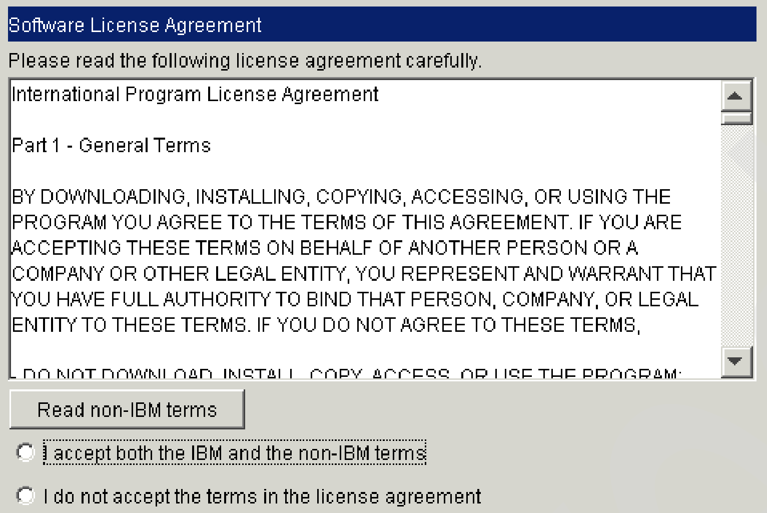
Welcome to the IBM WebSphere Application Server Network Deployment install wizard.

This wizard installs IBM WebSphere Application Server Network Deployment. Additional information can be found at the Information Centers and Support sites for WebSphere and related products homepage.

Click **Next** to continue.

Figure 14-2 Initial install screen

Figure 14-3 shows the license agreement, which you must accept.



Software License Agreement

Please read the following license agreement carefully.

International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

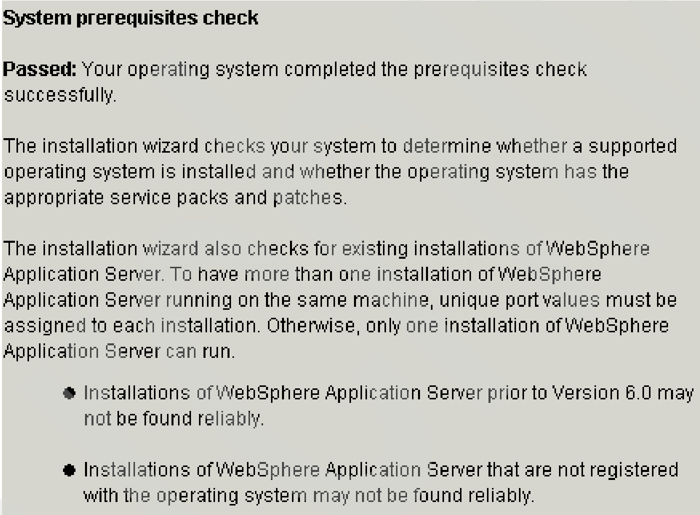
- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM;

○ I accept both the IBM and the non-IBM terms

○ I do not accept the terms in the license agreement

Figure 14-3 Installation license agreement

Figure 14-4 shows that the prerequisite check is passed.



System prerequisites check

Passed: Your operating system completed the prerequisites check successfully.

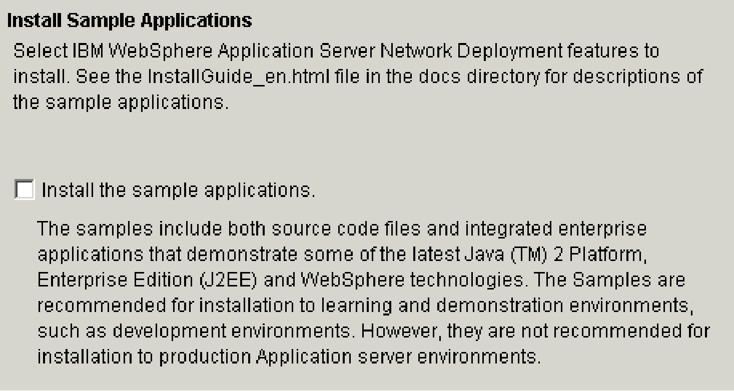
The installation wizard checks your system to determine whether a supported operating system is installed and whether the operating system has the appropriate service packs and patches.

The installation wizard also checks for existing installations of WebSphere Application Server. To have more than one installation of WebSphere Application Server running on the same machine, unique port values must be assigned to each installation. Otherwise, only one installation of WebSphere Application Server can run.

- Installations of WebSphere Application Server prior to Version 6.0 may not be found reliably.
- Installations of WebSphere Application Server that are not registered with the operating system may not be found reliably.

Figure 14-4 Operating system prerequisite check

Figure 14-5 shows the choice to select or omit the sample programs.



Install Sample Applications

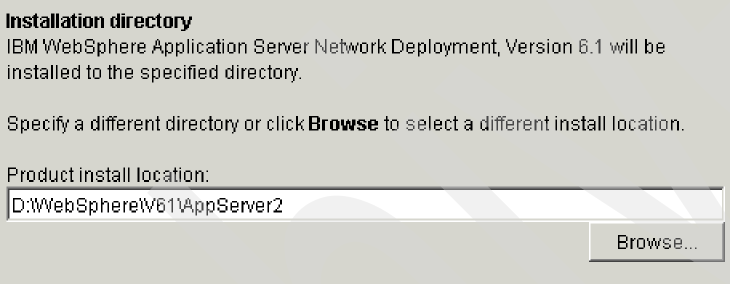
Select IBM WebSphere Application Server Network Deployment features to install. See the InstallGuide_en.html file in the docs directory for descriptions of the sample applications.

□ Install the sample applications.

The samples include both source code files and integrated enterprise applications that demonstrate some of the latest Java (TM) 2 Platform, Enterprise Edition (J2EE) and WebSphere technologies. The Samples are recommended for installation to learning and demonstration environments, such as development environments. However, they are not recommended for installation to production Application server environments.

Figure 14-5 Optional sample program install

Figure 14-6 shows the optional choice of install location.



Installation directory

IBM WebSphere Application Server Network Deployment, Version 6.1 will be installed to the specified directory.

Specify a different directory or click **Browse** to select a different install location.

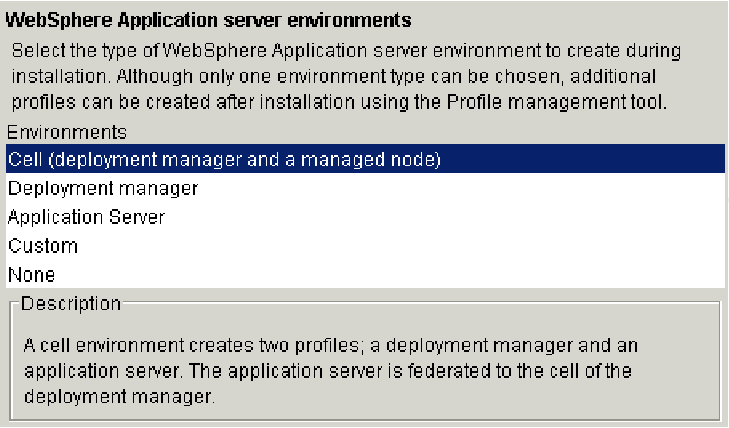
Product install location:

D:\WebSphere\V61\AppServer2

Figure 14-6 Install location

Figure 14-7 gives you the choice to select a profile type:

- Cell actually creates two profiles, a profile for the deployment manager and a profile for a federated node. If you choose this option, you have a minimum cell with a single managed node.
- Deployment manager creates a single profile for a deployment manager.
- Application server creates a single profile for a standalone server.
- Custom creates a single profile for a federated node.
- None creates no profiles to accompany the installation. You would chose none if you want more control over the profile creation.



WebSphere Application server environments

Select the type of WebSphere Application server environment to create during installation. Although only one environment type can be chosen, additional profiles can be created after installation using the Profile management tool.

Environments

- Cell (deployment manager and a managed node)
- Deployment manager
- Application Server
- Custom
- None

Description

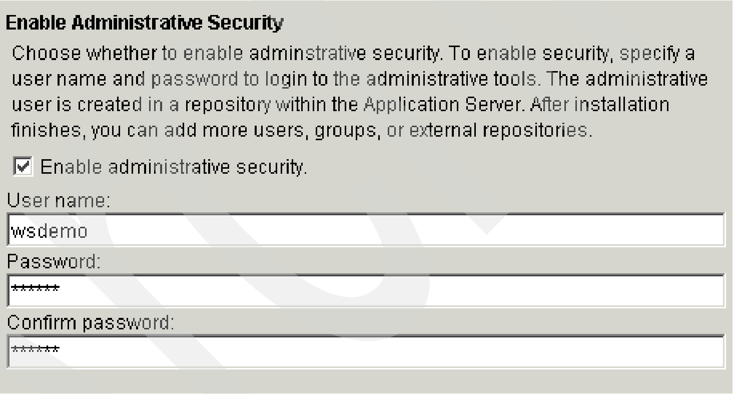
A cell environment creates two profiles; a deployment manager and an application server. The application server is federated to the cell of the deployment manager.

Figure 14-7 Selecting profile type

Figure 14-8 shows the optional selection of administrative security. This screen is new to V6.1 and enables security by default. You must specify an administrator ID and password. Note that this user and password is not registered in the operating system user registry, but is instead stored in a file in the profile. Thus, you can choose a name and password you like and not worry about a conflict with an operating system ID.

If you need to specify security using the operating system registry or LDAP registry, you would need to first start the server and configure this type of security using the administrative console.

The security screen is the last screen you see before the installation starts.



Enable Administrative Security

Choose whether to enable adminstrative security. To enable security, specify a user name and password to login to the administrative tools. The administrative user is created in a repository within the Application Server. After installation finishes, you can add more users, groups, or external repositories.

☑ Enable administrative security.

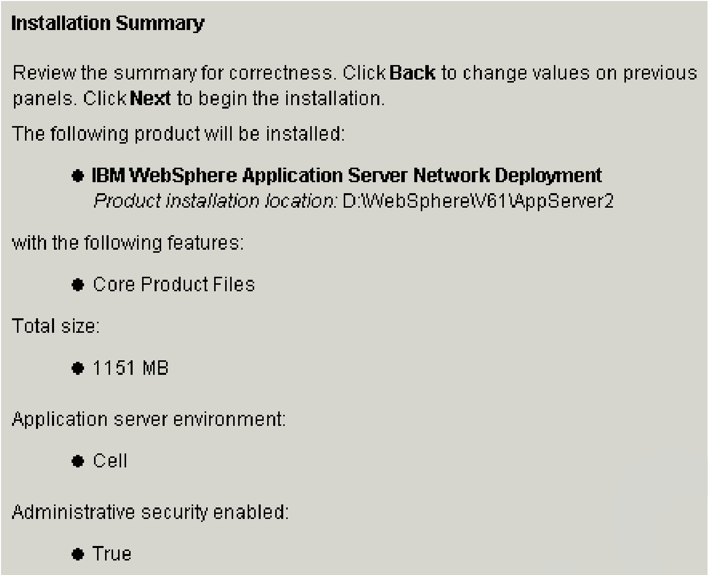
User name: wsdemo

Password: ******

Confirm password: ******

Figure 14-8 Enabling security

The installation proceeds for a few minutes, and you then see the closing screen whereby you exit the install, as shown in Figure 14-9.



Installation Summary

Review the summary for correctness. Click **Back** to change values on previous panels. Click **Next** to begin the installation.

The following product will be installed:

- **IBM WebSphere Application Server Network Deployment**
  Product installation location: D:\WebSphere\V61\AppServer2

with the following features:

- Core Product Files

Total size:

- 1151 MB

Application server environment:

- Cell

Administrative security enabled:

- True

Figure 14-9 Finishing installation

Upon finishing the installation, you have default profiles created, assuming you selected any profile type except none. Note the differences that streamline the installation:

- Network deployment product environments can have an operating environment immediately after installation. In V6.0, you have to run the profile creation tool directly after running the core installer. This tighter integration is not much different from V6, though. The V6.0 network deployment installer has a check box that launches the profile creation tool automatically.
- WebSphere Application Server Network Deployment, WebSphere Application Server, and WebSphere Application Server Express all install the same way. That is, all three product variations create a default profile. The default profile is not named default, as is the case with V6.0. The default will typically be called AppSrv01, in the case of an application server profile.
- Several of the default profile parameters have been chosen for you such that you are not given the option of specifying them. These parameters are:
  - Profile name
  - Node name
  - Cell name
  - Host address
  - Port numbers

If you need the option to specify these parameters, you must do so by running the profile management tool. See the next section for details.

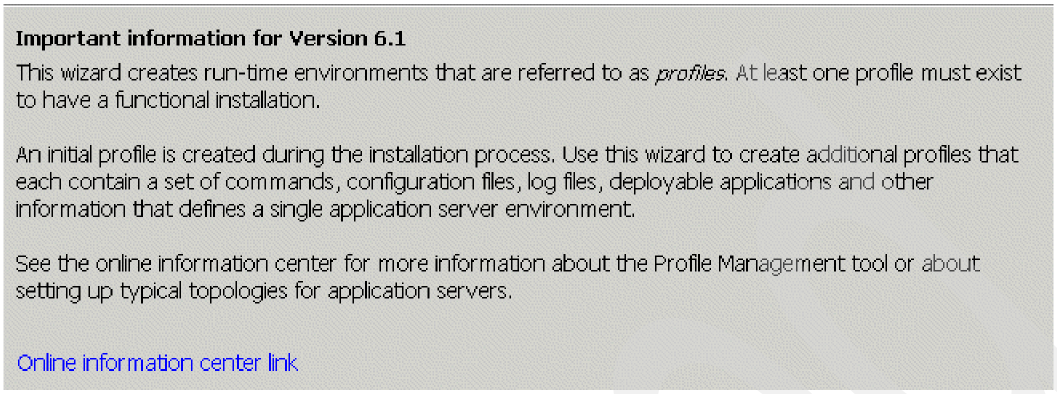
14.3.2 Creating profiles

The profile creation tool has been renamed to *profile management tool*, (PMT). The file path is also changed to:

- Windows - <WAS_HOME>\bin\ProfileManagement\pmt.bat
- UNIX - <WAS_HOME>/bin/ProfileManagement/pmt.sh

PMT does the same tasks as does the profile creation tool. The user interface screens are somewhat different in appearance and order.

Figure 14-10 shows the initial screen after executing the **pmt** command.



Important information for Version 6.1

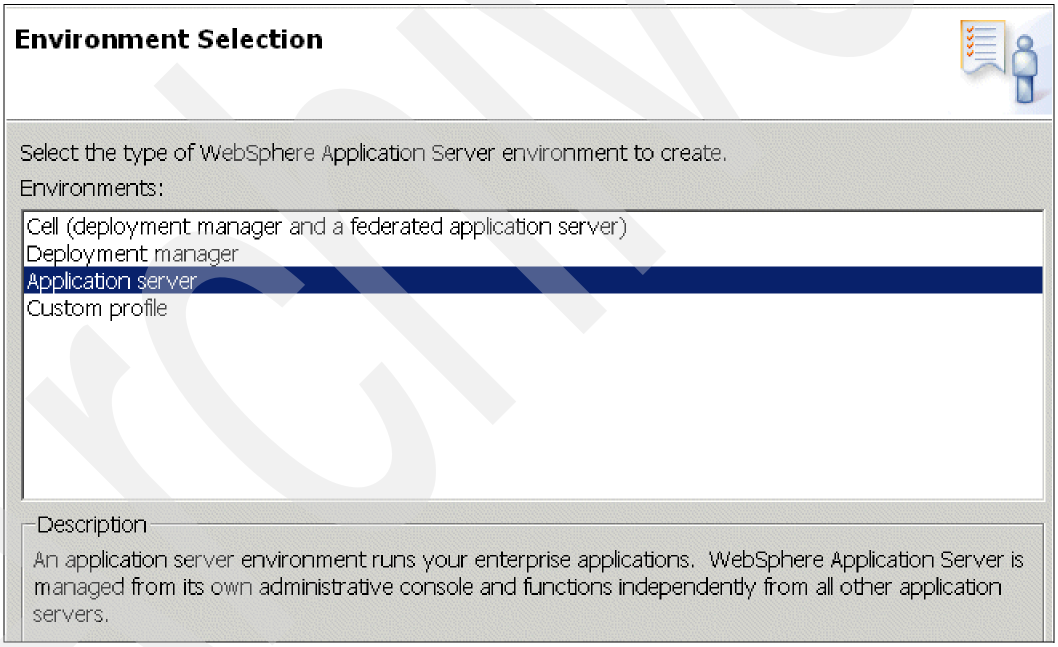
This wizard creates run-time environments that are referred to as *profiles*. At least one profile must exist to have a functional installation.

An initial profile is created during the installation process. Use this wizard to create additional profiles that each contain a set of commands, configuration files, log files, deployable applications and other information that defines a single application server environment.

See the online information center for more information about the Profile Management tool or about setting up typical topologies for application servers.

Figure 14-10 Initial profile management screen

Figure 14-11 shows the selection of the profile type. The profile types were described in the previous section.



Environment Selection

Select the type of WebSphere Application Server environment to create.

Environments:

- Cell (deployment manager and a federated application server)
- Deployment manager
- Application server
- Custom profile

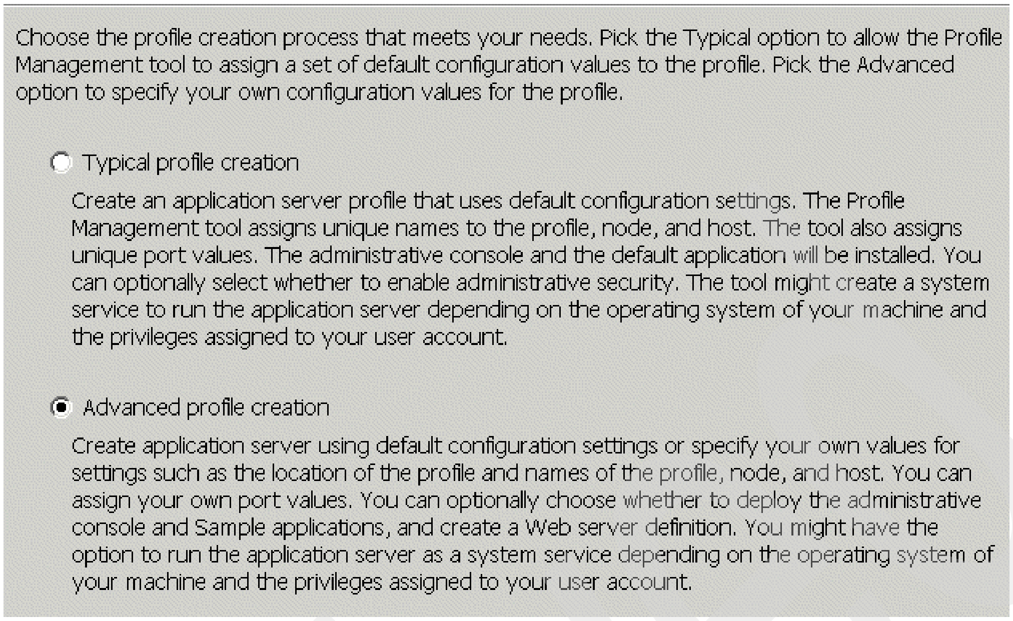
Description

An application server environment runs your enterprise applications. WebSphere Application Server is managed from its own administrative console and functions independently from all other application servers.

Figure 14-11 Profile type selection

Figure 14-12 shows the selection of either typical or advanced parameter selection.

- *Typical* advances you through an abbreviated parameter selection sequence, similar to the installation screens, beginning with Figure 14-5 on page 308.
- *Advanced* allows you to specify all the pertinent parameters. The remainder of the screens show the advanced parameters selections.



Choose the profile creation process that meets your needs. Pick the Typical option to allow the Profile Management tool to assign a set of default configuration values to the profile. Pick the Advanced option to specify your own configuration values for the profile.

○ Typical profile creation

Create an application server profile that uses default configuration settings. The Profile Management tool assigns unique names to the profile, node, and host. The tool also assigns unique port values. The administrative console and the default application will be installed. You can optionally select whether to enable administrative security. The tool might create a system service to run the application server depending on the operating system of your machine and the privileges assigned to your user account.

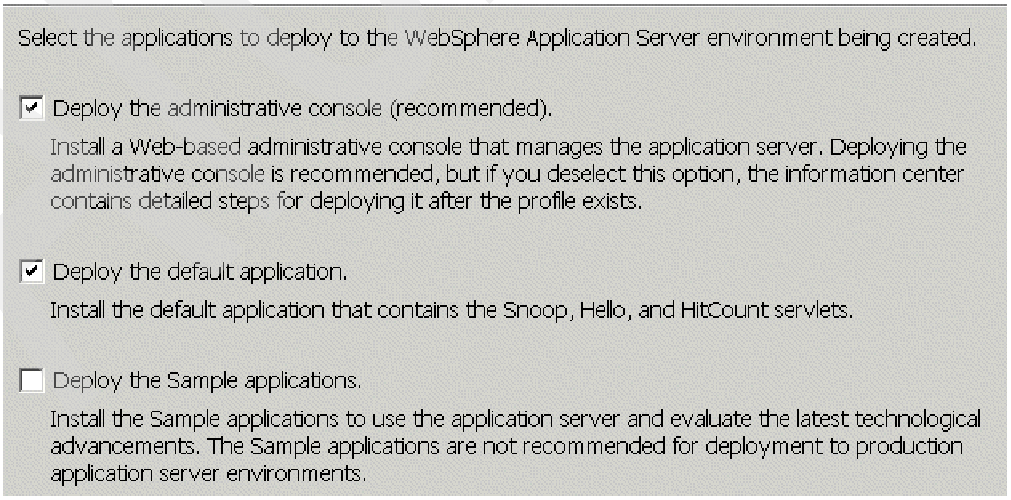
● Advanced profile creation

Create application server using default configuration settings or specify your own values for settings such as the location of the profile and names of the profile, node, and host. You can assign your own port values. You can optionally choose whether to deploy the administrative console and Sample applications, and create a Web server definition. You might have the option to run the application server as a system service depending on the operating system of your machine and the privileges assigned to your user account.

Figure 14-12 Choosing typical or advanced profile parameter specification

Figure 14-13 shows the screen where you select which standard application sets to install.

- The administrative console should be deployed if you are choosing a stand-alone application server or a deployment manager. For other profile types, do not deploy the administrative console.
- The default application is used as a debugging aid. It should be deployed if you are choosing a stand-alone application server. The default application could be installed on other profile types, but it does not make much sense to do so.
- The sample applications are optional for a stand-alone application server. The sample applications are useful in an application development environment. You can always choose to install the sample applications later using the application installation wizard in the administrative console.



Select the applications to deploy to the WebSphere Application Server environment being created.

☑ Deploy the administrative console (recommended).

Install a Web-based administrative console that manages the application server. Deploying the administrative console is recommended, but if you deselect this option, the information center contains detailed steps for deploying it after the profile exists.

☑ Deploy the default application.

Install the default application that contains the Snoop, Hello, and HitCount servlets.

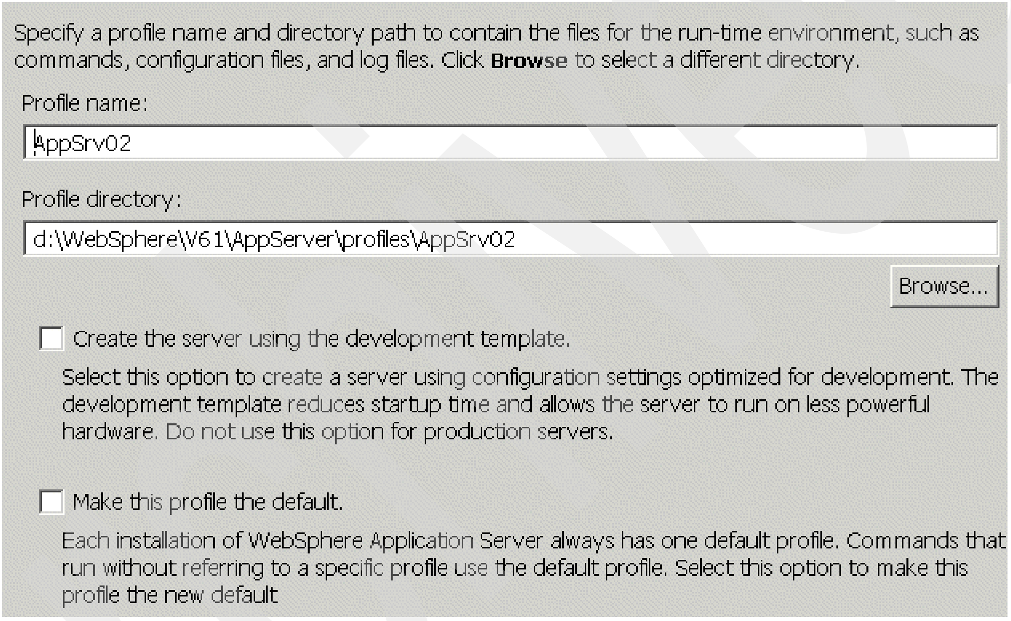
□ Deploy the Sample applications.

Install the Sample applications to use the application server and evaluate the latest technological advancements. The Sample applications are not recommended for deployment to production application server environments.

Figure 14-13 Selecting which applications to deploy

Figure 14-14 shows how to specify the profile name and profile location. You also see these optional buttons:

- You can enable the server to start using the development template. The development template adds two JVM options to the startup sequence to disable bytecode verification and to speed up JIT compiler optimization. While the JVM startup speed is enhanced, the subsequent runtime performance is degraded to some small degree. Choosing whether to use this option is a matter of how often your environment needs to restart application servers. In development testing environments, servers are sometimes restarted frequently to perform tests. If development testing matches your environment, you might try using the development template and then evaluate whether your tests run faster. If you have a production environment where your application server stays up for long periods between restarts, you should not use the development template option. Your performance will likely be degraded if you do.
- You can select this profile to be the default, which allows you to use this profile without specifying a profile name parameter on commands that require a profile name.



Specify a profile name and directory path to contain the files for the run-time environment, such as commands, configuration files, and log files. Click **Browse** to select a different directory.

Profile name:

AppSrv02

Profile directory:

d:\WebSphere\V61\AppServer\profiles\AppSrv02

□ Create the server using the development template.

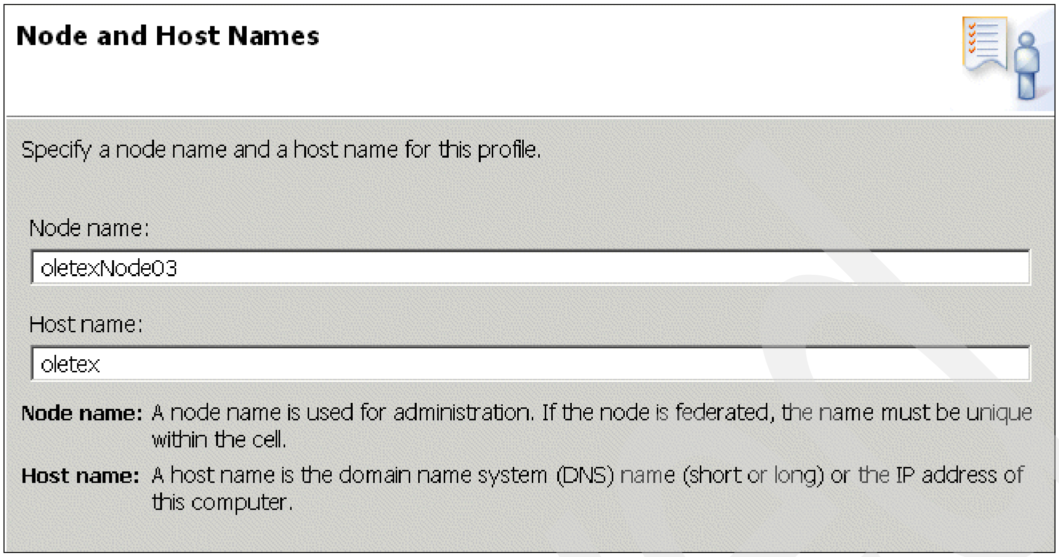
Select this option to create a server using configuration settings optimized for development. The development template reduces startup time and allows the server to run on less powerful hardware. Do not use this option for production servers.

□ Make this profile the default.

Each installation of WebSphere Application Server always has one default profile. Commands that run without referring to a specific profile use the default profile. Select this option to make this profile the new default

Figure 14-14 Specifying profile name and location

Figure 14-15 shows where you specify node and host name.



Node and Host Names

Specify a node name and a host name for this profile.

Node name:
oletexNode03

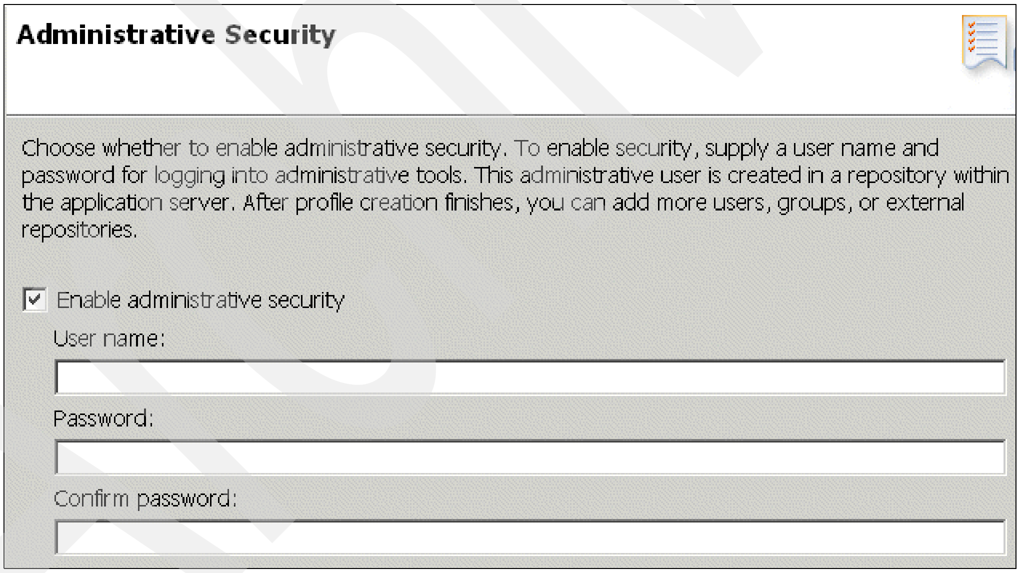
Host name:
oletex

Node name: A node name is used for administration. If the node is federated, the name must be unique within the cell.

Host name: A host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Figure 14-15 Specifying node and host name

Figure 14-16 shows how administrative security is configured. These parameters are identical to those when performing an installation. See Figure 14-8 on page 309 for details of the meanings of these parameters.



Administrative Security

Choose whether to enable administrative security. To enable security, supply a user name and password for logging into administrative tools. This administrative user is created in a repository within the application server. After profile creation finishes, you can add more users, groups, or external repositories.

☑ Enable administrative security

User name:

Password:

Confirm password:

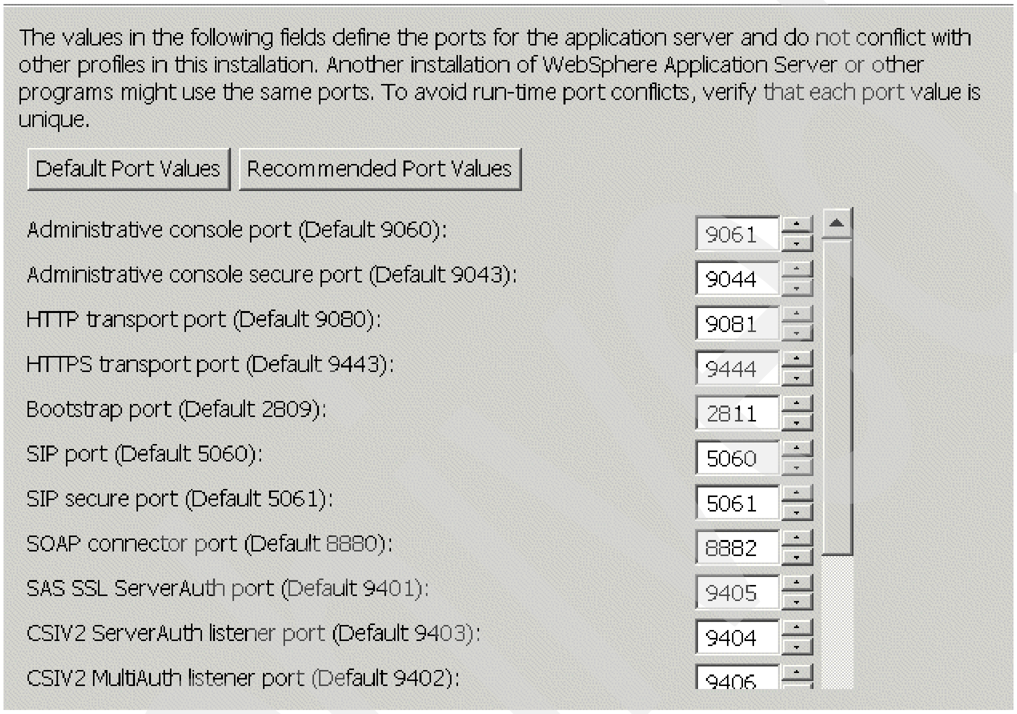
Figure 14-16 Enabling administrative security

Figure 14-17 shows how you can change the port assignments. When you first enter the screen for this profile, the profile management tool assigns ports based on assignments that have already been made for previous profiles. The profile management tool assigns ports such that previously assigned ports do not conflict. You have three choices on how to change the port assignments:

- You can key a number into each port field.
- You can click the Default Port Values button, which assigns each port to its base default value. The base default value is what is initially used when there are no other existing

profiles to check. If you use this button, it is up to you to check whether the resultant port assignments can conflict with other applications running concurrently.

- You can click the Recommended Port Values button, which assigns all ports to the best known value that does not conflict with other existing profiles. Note that Figure 14-17 shows that the Recommended Port Values button is disabled. When first entering the screen, all the port values are set to the recommended values. Changing a single value then enables the Recommended Port Values button.



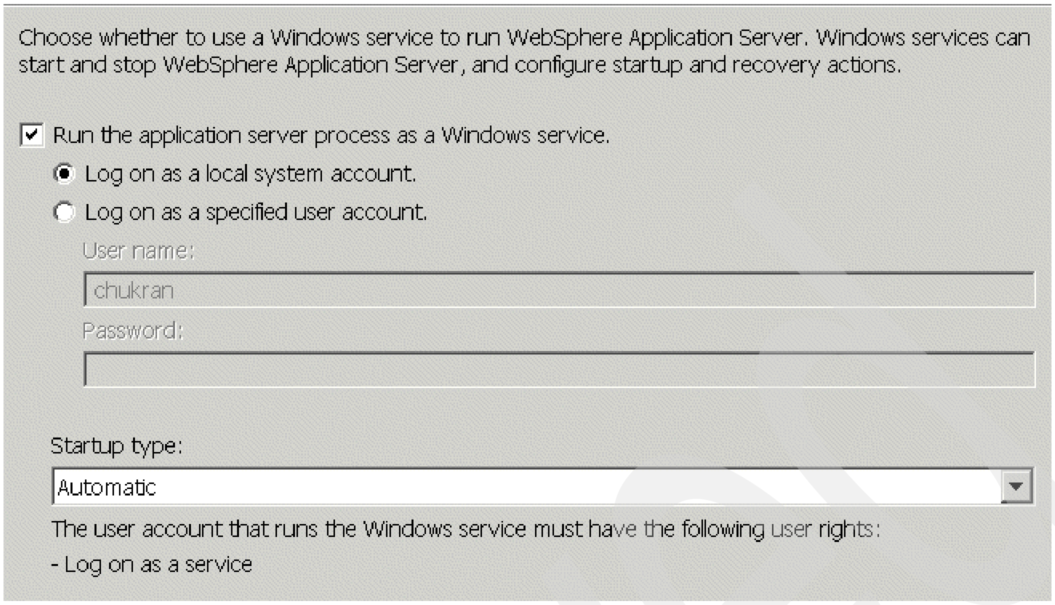
The values in the following fields define the ports for the application server and do not conflict with other profiles in this installation. Another installation of WebSphere Application Server or other programs might use the same ports. To avoid run-time port conflicts, verify that each port value is unique.

Administrative console port (Default 9060):	9061
Administrative console secure port (Default 9043):	9044
HTTP transport port (Default 9080):	9081
HTTPS transport port (Default 9443):	9444
Bootstrap port (Default 2809):	2811
SIP port (Default 5060):	5060
SIP secure port (Default 5061):	5061
SOAP connector port (Default 8880):	8882
SAS SSL ServerAuth port (Default 9401):	9405
CSIV2 ServerAuth listener port (Default 9403):	9404
CSIV2 MultiAuth listener port (Default 9402):	9406

Figure 14-17 Specifying port assignments

Figure 14-18 shows the Windows Service Definition screen, which is only shown for the Windows operating system. The service definition is checked by default. You can have the service run under the system ID, whereby no password is required. Or you can specify a particular existing user and password with which the service runs. That user must have Windows access rights to log on as a service. If you do not want to use Windows service definitions, simply uncheck the box to disable the definition.

Also note that you have no control over the service name. The profile management tool assigns a service name by combining the WebSphere product name and the profile name. Thus you can create multiple profiles that use a Windows service definition, and they will all have unique service names.



Choose whether to use a Windows service to run WebSphere Application Server. Windows services can start and stop WebSphere Application Server, and configure startup and recovery actions.

☑ Run the application server process as a Windows service.

● Log on as a local system account.

○ Log on as a specified user account.

User name: chukran

Password:

Startup type: Automatic

The user account that runs the Windows service must have the following user rights:

- Log on as a service

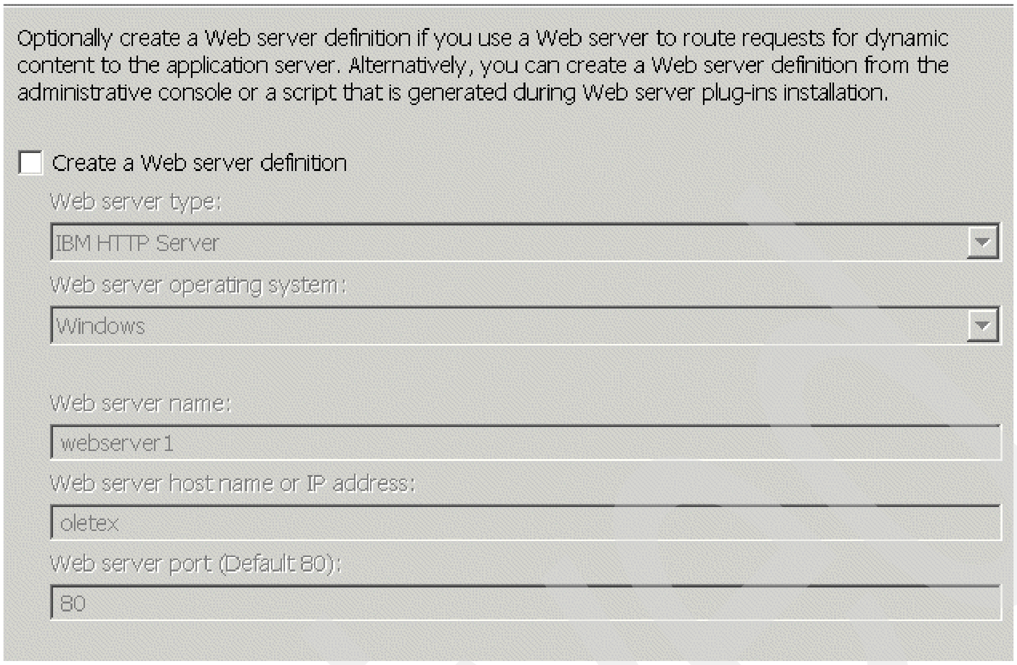
Figure 14-18 Windows service definition

Figure 14-19 shows the screen to create a Web server definition. The ability to create a Web server definition at profile creation time is a new feature for V6.1. You previously had to create a Web server definition via the administrative console. You can still create a Web server definition via the administrative console as you may have for V6.0. See 12.3.6, “Migrating Web server and WebSphere plug-ins” on page 201 for more information on creating Web servers.

If you want to create a Web server definition, you need to supply these parameters:

- The Web server type, which is selected from a pull-down menu.
- The operating system on which the Web server runs, which is selected from a pull-down menu.
- A Web server name of your choice.
- The host name or IP address where the server runs.
- The port number for the Web server.

If you don’t need a Web server definition, simply uncheck the Create box and continue.



Optionally create a Web server definition if you use a Web server to route requests for dynamic content to the application server. Alternatively, you can create a Web server definition from the administrative console or a script that is generated during Web server plug-ins installation.

□ Create a Web server definition

Web server type: IBM HTTP Server

Web server operating system: Windows

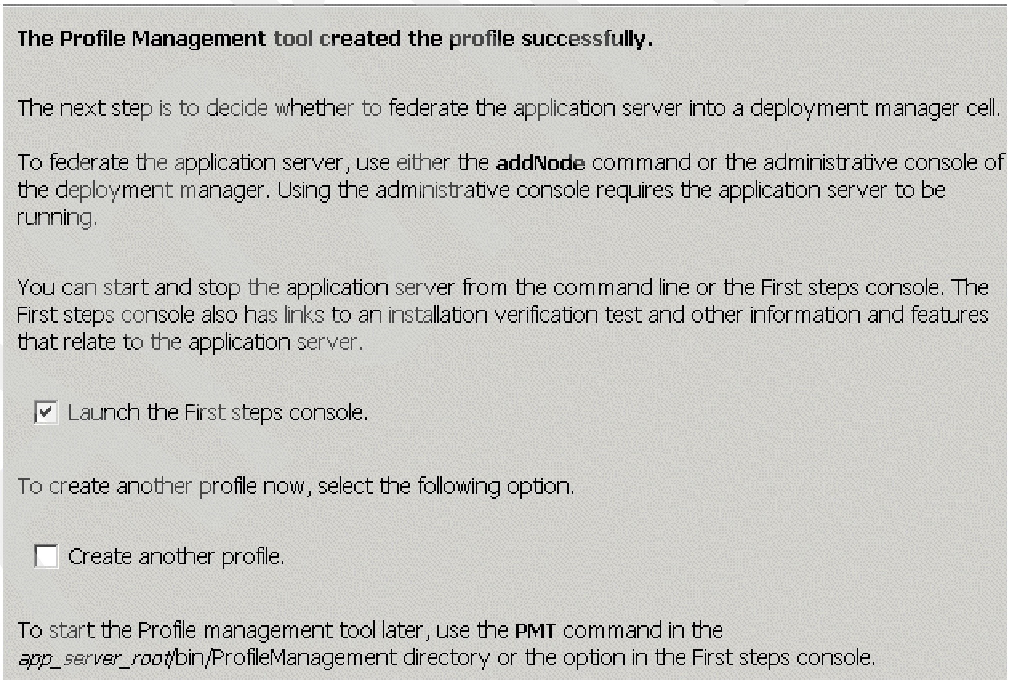
Web server name: webserver1

Web server host name or IP address: oletex

Web server port (Default 80): 80

Figure 14-19 Web server definition

Figure 14-20 shows the completion screen after the profile has been successfully created. You have the option to launch the First Steps console, which allows you to launch the server defined by the profile you just created.



The Profile Management tool created the profile successfully.

The next step is to decide whether to federate the application server into a deployment manager cell.

To federate the application server, use either the **addNode** command or the administrative console of the deployment manager. Using the administrative console requires the application server to be running.

You can start and stop the application server from the command line or the First steps console. The First steps console also has links to an installation verification test and other information and features that relate to the application server.

☑ Launch the First steps console.

To create another profile now, select the following option.

□ Create another profile.

To start the Profile management tool later, use the **PMT** command in the *app_server_root*/bin/ProfileManagement directory or the option in the First steps console.

Figure 14-20 Profile completion

Command line driven installation also has changes:

- The wasprofile command has been changed to <WAS_HOME>\bin\manageprofiles.

- The V6.0 response files for silent installation are not compatible with V6.1. You must change any V6.0 response files to the new syntax. For more information on the new syntax, see the article titled *Customizing the response file* at this address:

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

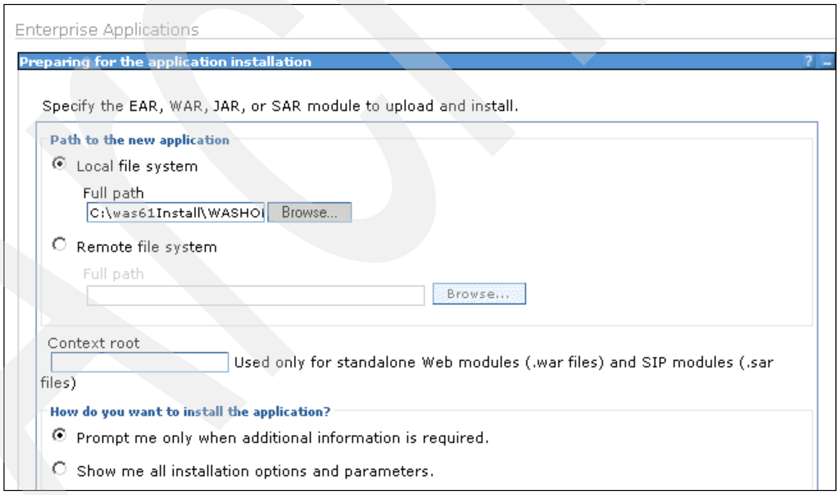
14.4 Administrative console

For V6.1, the administrative console has had a minor face-lift to improve usability. Most of the changes are additions to the existing user interface, but a few are changes to the default navigation that cause you to operate the console differently in order to do basic tasks. We describe only those changes that change the basic operation of the administrative console here. To view all the new administrative console changes, consult the IBM Education Assistant at this address:

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/Administration.html

14.4.1 Enterprise application install wizard

The enterprise application install wizard shows you a reduced number of steps when you attempt to install an application. This reduction in steps may be unexpected based on your experience with earlier versions. Figure 14-21 shows the initial screen for the application installation wizard. The radio button labeled “Prompt me only when additional information is required” is checked by default. This means that the installation wizard first examines the EAR file you are installing, and omits any steps that already have data specified in the EAR file. Figure 14-24 shows the subsequent screen using the default behavior. Note that only 3 steps appear in the step list.


Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

Full path: C:\was61Install\WASHO

○ Remote file system

Full path:

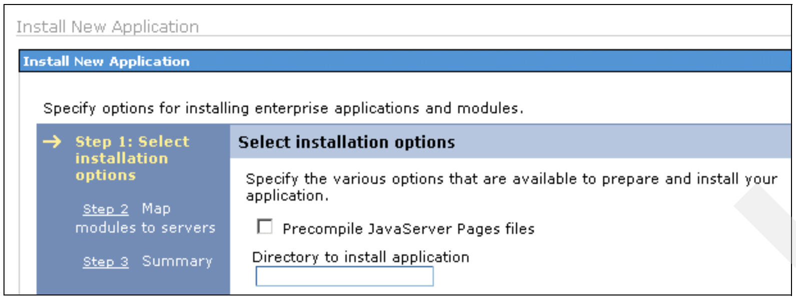
Context root: Used only for standalone Web modules (.war files) and SIP modules (.sar files)

How do you want to install the application?

● Prompt me only when additional information is required.

○ Show me all installation options and parameters.

Figure 14-21 Application installation initial default



Install New Application

Install New Application

Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options

Step 2 Map modules to servers

Step 3 Summary

Select installation options

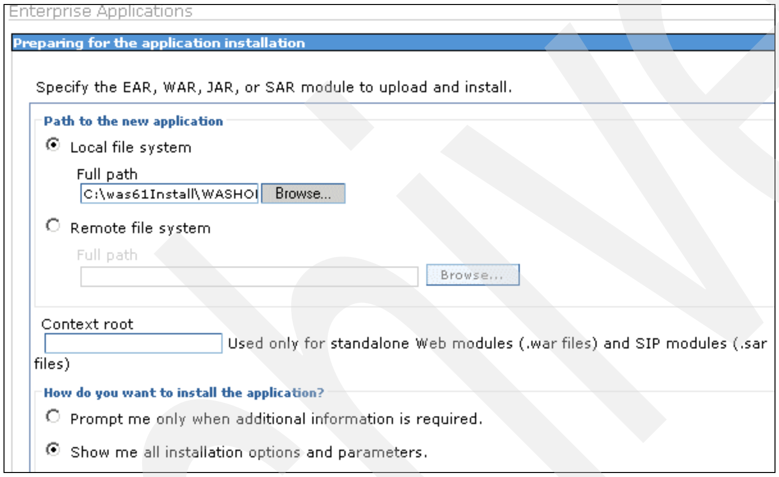
Specify the various options that are available to prepare and install your application.

□ Precompile JavaServer Pages files

Directory to install application

Figure 14-22 Application installation using reduced steps

If you want the V6.0 behavior where all the pertinent steps are shown so you can override information that is contained in the EAR file, check the box labeled “Show me all installation options and parameters,” as shown in Figure 14-23. This non-default option results in the screen shown in Figure 14-24. Note that there are now seven steps instead of three.



Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

Full path

C:\was61Install\WASHOI

○ Remote file system

Full path

Context root

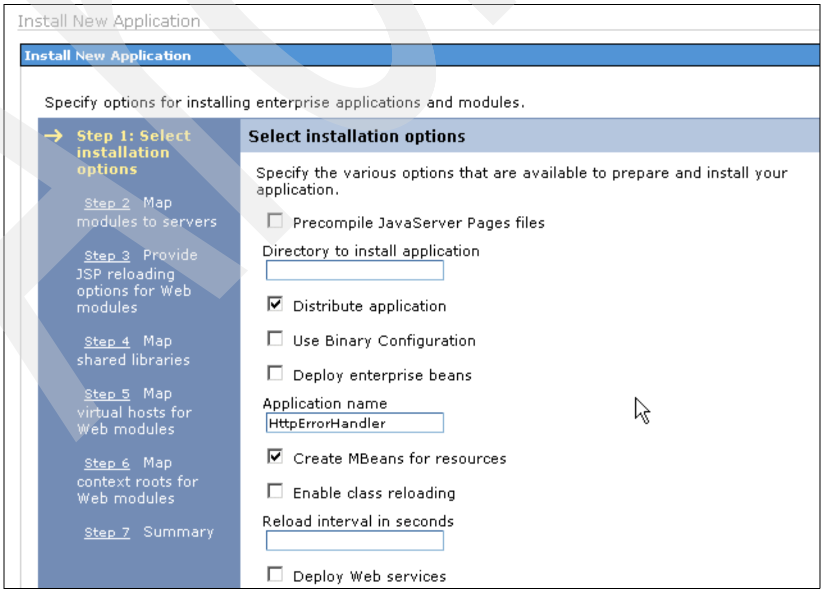
Used only for standalone Web modules (.war files) and SIP modules (.sar files)

How do you want to install the application?

○ Prompt me only when additional information is required.

● Show me all installation options and parameters.

Figure 14-23 Installation initial screen non-default behavior



Install New Application

Install New Application

Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options

Step 2 Map modules to servers

Step 3 Provide JSP reloading options for Web modules

Step 4 Map shared libraries

Step 5 Map virtual hosts for Web modules

Step 6 Map context roots for Web modules

Step 7 Summary

Select installation options

Specify the various options that are available to prepare and install your application.

□ Precompile JavaServer Pages files

Directory to install application

☑ Distribute application

□ Use Binary Configuration

□ Deploy enterprise beans

Application name

HttpErrorHandler

☑ Create MBeans for resources

□ Enable class reloading

Reload interval in seconds

□ Deploy Web services

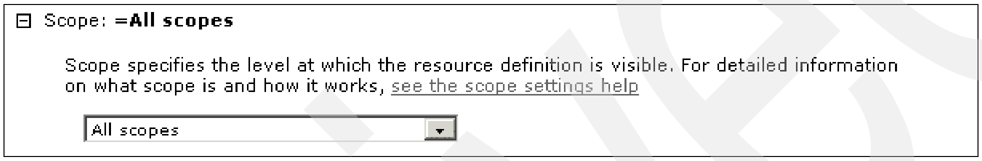
Figure 14-24 Application installation with full number of steps

14.4.2 Resource scope selection

Resource scope selection in the administrative console has been improved over that in V5 and V6.0. To review resource scope, remember that a resource can be defined to apply to three different granularity levels:

- Cell scope, in which the resource applies to every node and every server
- Node scope, in which the resource applies to every server in a single specific node
- Server scope, in which the resource applies to a single specific server in a single specific node

The prior versions are clumsy in how you select the scope. The prior versions also provide a default scope which may often allow you to forget to change the scope of the resource. V6.1 has changed both of these aspects of scope selection. Figure 14-25 show an example of creating a new resource. Note that the default scope is “All scopes.” All scopes is not a valid scope setting, which means you must select a valid scope from the pull-down list.



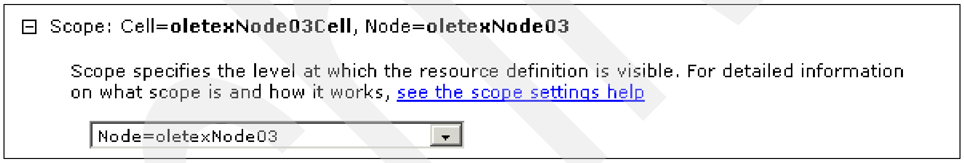
⊟ Scope: **=All scopes**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

All scopes ▾

Figure 14-25 Initial default scope selection

Figure 14-26 shows an example of how node scope is selected with a single menu selection. Upon selecting a scope, you can then proceed with creating the resource.



⊟ Scope: Cell=**oletexNode03Cell**, Node=**oletexNode03**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

Node=oletexNode03 ▾

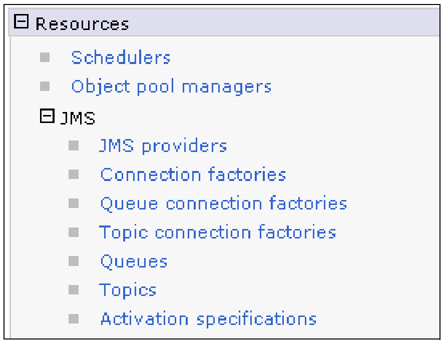
Figure 14-26 Selecting node scope

14.4.3 JMS resource navigation

The JMS resources have been refactored in V6.1 to allow a more logical organization. The operation is the same as you may be used to, but with V6.1, you now have the choice to access JMS resources directly. The advantage of accessing the JMS provider directly is that you do not have to know the owning JMS provider, thus speeding up browsing for JMS resources.

Figure 8-23 on page 114 shows the JMS resource organization for previous versions. Figure 14-27 shows the refactored JMS resources. Note that connection factories are now shown at the top level, enabling you to view the resource without knowing the owning provider. Of course, you may still access the same resources by first selecting the JMS provider, and then selecting the resource under the provider. For more information, see the article titled *Installing and configuring a JMS provider* at this address:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tmj_inst0.html`



⊟ Resources
- Schedulers
- Object pool managers

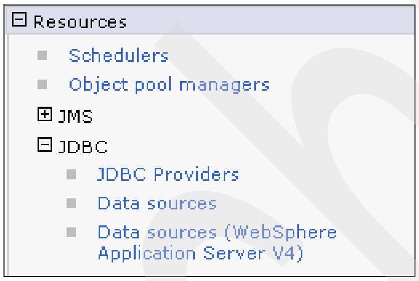
⊟ JMS
- JMS providers
- Connection factories
- Queue connection factories
- Topic connection factories
- Queues
- Topics
- Activation specifications

Figure 14-27 Refactored JMS resources

14.4.4 JDBC data source creation

JDBC resources have been refactored to allow you to access the data sources directly. Figure 14-28 shows how the JDBC resources have been refactored. Of course, you may still access the data sources via the appropriate JDBC provider as you may be used to. For more information see the article titled *Configuring a JDBC provider using the administrative console* at this address:

```
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.web
sphere.nd.doc/info/ae/ae/tdat_ccrtprov.html
```



⊟ Resources
- Schedulers
- Object pool managers

⊞ JMS

⊟ JDBC
- JDBC Providers
- Data sources
- Data sources (WebSphere Application Server V4)

Figure 14-28 Refactored JDBC resources

14.5 Restrictions on mixed version cells

Table 14-2 shows a summary of the remaining restrictions on mixed version cells. Refer to Table 10-2 on page 159 for the restrictions in place for V6.0. One additional restriction has been lifted, namely that you can now federate a V6.0.2 node to a V6.1 cell. It is still a restriction that you cannot federate a V5.x node to a V6.1 cell.

Table 14-2 Mixed version restrictions when adding resources

	New resource to add	Supported in V6.1
Adding new nodes	Federate V6.0.2 node	Yes
	Federate V5.x node	No
Adding new servers	Add server to V6.0.2 node	Yes
	Add server to V5.x node	Yes

	New resource to add	Supported in V6.1
Adding new cluster members	Add V5.x server to V5.x-only cluster	Yes
	Add V5.x server to V6.x-only cluster	No
	Add V6.x server to V5.x-only cluster	No
	Add V5.x server to mixed cluster	Yes
	Add V6.x server to mixed cluster	Yes

> **Note:** If you are migrating WebSphere Application Server for z/OS, then there is a special restriction that applies only to the z/OS platform. See 15.3.4, “Mixed-version cell restriction for z/OS” on page 330 for more details on how that restriction may affect you.

14.6 System administration function deprecations and removals

There are several features and commands that either have changed names or locations, or have been declared obsolete. You should be aware of the details of these changes that may affect your migration to V6, or may affect migrations to future versions.

14.6.1 Commands

Table 14-3 lists commands that have been deprecated and the replacements you should use

Table 14-3 Commands deprecated for V6.1

Command	Replacement
wasprofile	Use manageprofiles instead
pctWindows,	Use pmt command instead
wcslogbr	Use log analyzer view in application server toolkit

Profile creation commands

The facilities that perform profile creation have changed names, but the basic functionality is the same as in V6.0. V6.0 has the **`wasprofile`** command to manipulate profiles, while V6.1 has both **`wasprofile`** and **`manageprofiles`** commands. Both wasprofile and manageprofiles take the same options and are essentially aliases of one another. Wasprofile is deprecated, which means that the command will eventually disappear in a future release. You should be using manageprofiles in order to prepare for the eventual deletion of wasprofile.

The profile creation graphical tools have changed names in V6.1. V6.0 stores the profile creation tool in the directory bin/ProfileCreator. The command launch name varies with the operating platform. For example, Windows provides the command **`pctWindows`**. The other platforms have the name prefixed accordingly. V6.1 names this command **`pmt`** and uses this name consistently across all the operating platforms. Pmt is stored in bin\ProfileManagement.

See 14.3.2, “Creating profiles” on page 310 for more details on use of the profile management tool.

Log analyzer

The application server log analyzer is a stand-alone tool in V6.0 and is launched by the **wcslogbr** command. This command has been removed in V6.1. The equivalent is now packaged in the Application Server Toolkit. For more information see the Infocenter article “Analyzing log records” at this address:

```
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/org.eclipse
.tptp.platform.doc.user/tasks/teanlog.htm
```

14.6.2 Application deployment features

Table 14-4 Application deployment features deprecated in V6.1

Feature	Action
Cloudscape V5.1 database	Use Cloudscape V10 database
V5 JMS resources	Use V5 JMS resources
DB2 Legacy JDBC driver	Use DB2 Universal JDBC driver
Simple WebSphere Authentication Method	Use Lightweight Third Party Authentication

Cloudscape database

The Cloudscape V5.1 database is no longer provided and has been replaced by the Cloudscape 10.1 database. The Cloudscape component is now based on the Apache Derby project, and consequently some of the terminology has changed. Any references in the documentation to the name Derby can be considered to be equivalent to the name Cloudscape. For more information about Cloudscape 10.1 and how it differs from Cloudscape 5.1 see the article titled *About Cloudscape 10.1.x* at this address:

```
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.web
sphere.nd.doc/info/ae/ae/rdat_Cloudscape101x.html
```

If you have a Cloudscape 5.1 database instance, you must first migrate the database to Cloudscape 10.1. Cloudscape migration can be accomplished in one of two ways:

- Run the automatic migration utilities which perform Cloudscape migration as one of the subtasks. See 8.6.2, “Configuration migration” on page 125 for more information on how to run the automatic migration utilities.
- Manually run the Cloudscape database migration tool. This tool takes a Cloudscape 5.1 database, copies it, and then converts the copy to Cloudscape 10.1 format. To learn more about manually migrating your Cloudscape 5.1 databases, see the article titled *Migrating Cloudscape databases* at this address:

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

The Cloudscape data store helpers have been superseded by the equivalent Derby datastore helpers. The Cloudscape data store helpers still exist, but are deprecated. Table 14-5 summarizes the equivalencies.

Table 14-5 Cloudscape data store helpers

Cloudscape data store helpers	Derby data store helpers
com.ibm.websphere.rsadapter.CloudscapeDataStoreHelper	com.ibm.websphere.rsadapter.DerbyDataStoreHelper
com.ibm.websphere.rsadapter.CloudscapeNetworkServerDataStoreHelper	com.ibm.websphere.rsadapter.DerbyNetworkServerDataStoreHelper

JMS resources

V5 default messaging resources are deprecated in V6.1. This means that any applications that use V5 default messaging resources will stop working when this type of resource is no longer supported. You should change your applications to use V6 messaging resources.

V5 default messaging resources are discussed on page 197.

DB2 JDBC driver

The DB2 CLI-based legacy Type 2 JDBC driver has been deprecated. You should use the DB2 Universal JDBC driver instead. Table 14-6 shows you the association of the driver type with its file name so you can identify whether you are using these drivers.

Table 14-6 DB2 JDBC driver file identification

Driver type	File
CLI Type 2 JDBC driver	db2java.zip
Universal Type 4 JDBC driver	dj2jcc.jar

SWAM authentication

The Simple WebSphere Authentication Method (SWAM) is a easy to configure means to perform authentication on stand-alone application servers. This method is deprecated in V6.1 and is being replaced by Lightweight Third Party Authentication (LTPA). Learn more about how to configure LTPA in the article titled *Lightweight Third Party Authentication* at this address:

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

14.6.3 Scripting language compatibility

The JACL scripting language is deprecated starting with V6.1. This means that the JACL scripting language is still supported in V6.1, but the language is no longer strategic. JACL will eventually be dropped in a future release. The Jython language, which is a Java implementation of Python, is the strategic scripting language. You should begin to re-implement your wsadmin scripts in the Jython language in preparation for discontinuance of support for JACL. To begin to migrate your scripts to Jython, you should get an overview of the scripting tools available in the WebSphere Application Server Toolkit. See the article titled *Developing Automation Scripts* at this address:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.ws.ast.jythontools.doc/topics/cautoscript.html

You can also investigate converting your JACL scripts to Jython using the JACL-to-Jython script conversion assistant available at this address:

http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012144

15

V6.1 updates for z/OS platforms

This chapter discusses the changes that affect migration that Version 6.1 has introduced compared to Version 6.0 on the z/OS operating platforms.

15.1 Summary of significant changes in V6.1 compared to V6.0

WebSphere Application Server V6.1 introduces many functional enhancements beyond the features of WebSphere Application Server V6.0. However, only a few of these enhancements affect the process of migration to V6.1. The enhancements you should know about if you are contemplating a migration to V6.1 are:

- Operating system prerequisites

  The z/OS operating system minimum levels have changed. See the next section for more information.

- Automatic migration utilities

  V6.1 introduces these minor changes to how automatic migration is done:

  - You may need to configure additional SAF profiles.
  - You may need to hand-tune the daemon JCL with regard to STEPLIB statements.
  - You are now presented with choices to how applications are migrated.
  - You are given a choice to convert your wsadmin scripts to accommodate minor compatibly changes.

  See 15.3, “Automatic migration utilities” on page 327 for more details.

- Installation changes

  V6.1 introduces minor changes to the ISPF server customization sequences that you should be aware of. In addition the type of ISPF server customization that you may be familiar with in previous version, V6.1 introduces a graphical tool with which you can perform customization on your Intel architecture workstation. This tool allows you to create the customization data sets on your Windows or Linux workstation and then upload the data sets to your z/OS system where the customization jobs are run. This new method is entirely optional. If you are more comfortable using the ISPF customization method, you can still use that approach.

  See 15.4, “Installation” on page 332 for more information.

- Administrative console changes

  The graphical administrative console has undergone minor changes in order to make it easier and more intuitive to use. These changes are identical to those changes implemented on the distributed platforms. See Section 14.4, “Administrative console” on page 318 for more information.

- Administrator functional deprecations and removals

  Most of the changes are identical to those changes implemented on the distributed platforms. See Section 14.6, “System administration function deprecations and removals” on page 322 for more information.

  There are a few removals and deprecations that are specific to only the z/OS platform. See 15.5, “Removals and deprecations specific to the z/OS platform” on page 333 for details.

15.2 Operating system prerequisites

WebSphere Application Server V6.1 has a new operating system requirement of z/OS 1.6 or higher. See “Prerequisites for WebSphere Application Server V6.0” on page 336 for more information.

15.3 Automatic migration utilities

The automatic migration utilities that are discussed in Section 13.1, “Migrating Network Deployment cell from V 5.1 to V6.0” on page 242 work in the same basic way in V6.1 as they do in V6.0. You begin by running ISPF screens that create migration jobs. Some of the ISPF screens have new options that enhance the effectiveness of the migration jobs. In some cases, you may have to do additional manual steps. You run the created jobs that the ISPF screens have created. In some cases, some of the job names have changed.

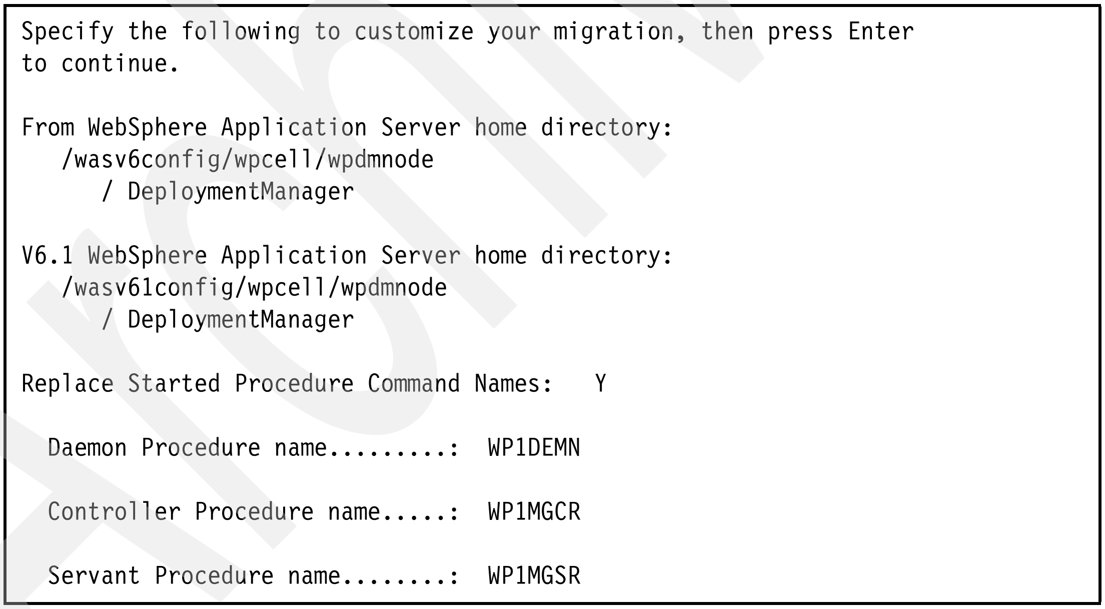
You also follow the general order of node migration described in Chapter 10., “Migration tasks” on page 151. If you are about to migrate to V6.1, look over the following subsections to understand the minor changes to the migration details. Then follow through the migration example in 13.1, “Migrating Network Deployment cell from V 5.1 to V6.0” on page 242.

If you want a complete walkthrough of a migration example explicitly for V6.1, consult the whitepaper *Migrating to WebSphere Application Server for z/OS Version 6.1* at this address:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100771

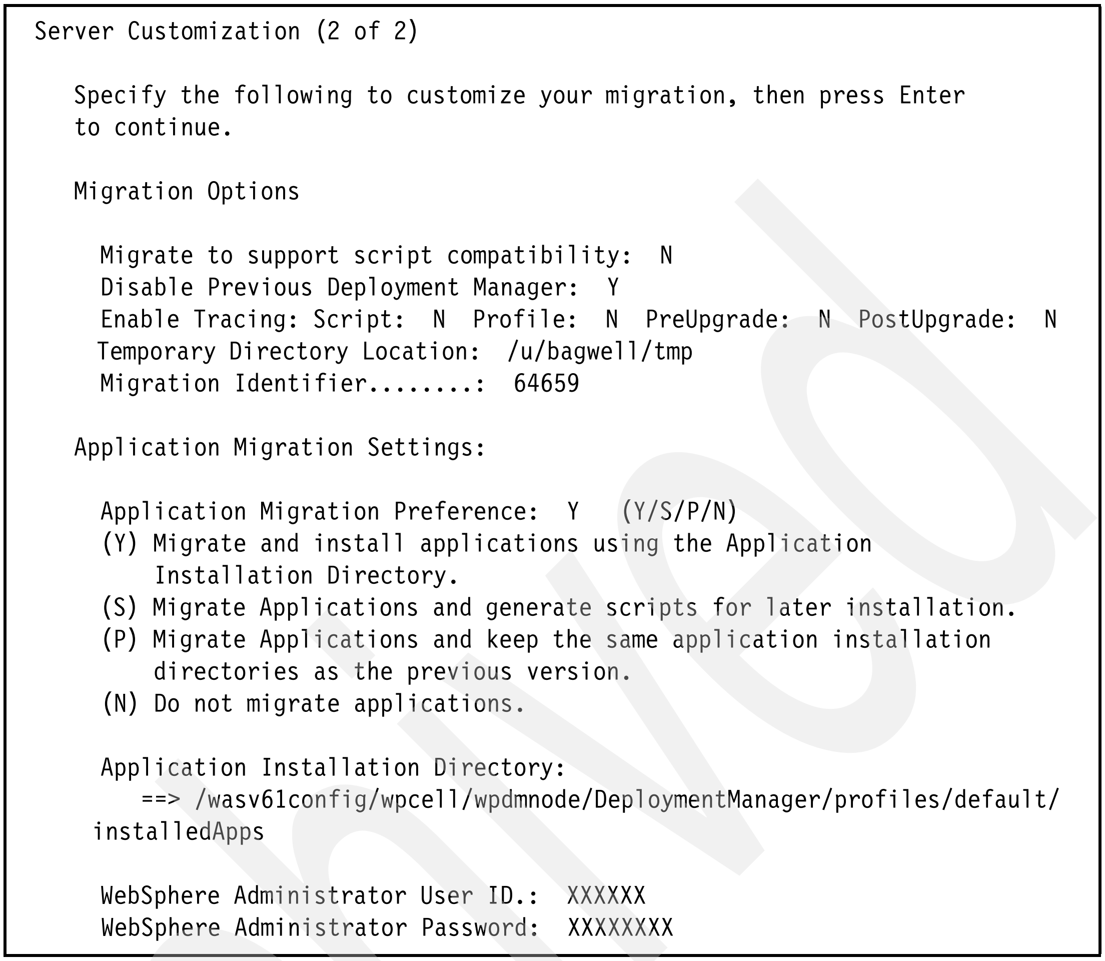
15.3.1 Modified ISPF migration screens

Some minor changes from V6.0 have been made to the server customization screens. Figure 15-1 and Figure 15-2 show two sequential screens that have additional fields to fill in, and that one field has been eliminated.



```
Specify the following to customize your migration, then press Enter
to continue.

From WebSphere Application Server home directory:
   /wasv6config/wpcell/wpdmnode
      / DeploymentManager

V6.1 WebSphere Application Server home directory:
   /wasv61config/wpcell/wpdmnode
      / DeploymentManager

Replace Started Procedure Command Names:   Y

  Daemon Procedure name.........:  WP1DEMN

  Controller Procedure name.....:  WP1MGCR

  Servant Procedure name........:  WP1MGSR
```

Figure 15-1 Server customization (Screen 1)



```
Server Customization (2 of 2)

   Specify the following to customize your migration, then press Enter
   to continue.

   Migration Options

     Migrate to support script compatibility:  N
     Disable Previous Deployment Manager:  Y
     Enable Tracing: Script:  N  Profile:  N  PreUpgrade:  N  PostUpgrade:  N
     Temporary Directory Location:  /u/bagwell/tmp
     Migration Identifier........:  64659

   Application Migration Settings:

     Application Migration Preference:  Y   (Y/S/P/N)
     (Y) Migrate and install applications using the Application
         Installation Directory.
     (S) Migrate Applications and generate scripts for later installation.
     (P) Migrate Applications and keep the same application installation
         directories as the previous version.
     (N) Do not migrate applications.

     Application Installation Directory:
        ==> /wasv61config/wpcell/wpdmnode/DeploymentManager/profiles/default/
     installedApps

     WebSphere Administrator User ID.:  XXXXXX
     WebSphere Administrator Password:  XXXXXXXX
```

Figure 15-2 Server customization (Screen 2)

Note that the field High Availability Manager Host, shown in Figure 13-12 on page 250 and Figure 13-38 on page 264, has been eliminated. The ISPF customization figures this out for itself from the host for the deployment manager.

The new fields you need to specify are:

- Replace Started Procedure Command Names

  WebSphere Application Server stores the START command to be used for each server in its XML configuration files, including the JCL start procedure name to use.

  This field asks whether you want the JCL start procedure names you provide on this screen to be updated in the WebSphere configuration itself. This defaults to **Y** (Yes), which means whatever JCL procedure names you supply will be used to update the configuration XML as well. You would specify **No** only if you were absolutely certain the JCL names you supply here are the same as what was used before. If in doubt, choose Yes and allow the migration utility to update.

  If you make the wrong choice, one of the following might happen:

  – A JCL error occurs if the old JCL can't be found.

  – The server is started up using the old JCL, with the possibility of wrong modules being pulled in by STEPLIB statements.

New JCL is copied into PROCLIB when you run the customized jobs. These separate events happen:

  - JCL is copied into PROCLIB by the customized batch job.
  - The update of the JCL names in the configuration is done only if you choose Yes.

- Migrate to support script compatibility

  You now have the option for the migration utilities to perform a script compatibility translation for you by specifying Yes for this option. If you choose `Y` (Yes), this option now runs the `convertScriptCompatibility` command under the covers. See Chapter 11, “Script compatibility” on page 167 and 8.6.4, “Script compatibility” on page 135 for more information on what this option does for you. In the V6.0 migration, this conversion is not done for you and you consequently have to run `convertScriptCompatibility` manually.

- Disable Previous Deployment Manager

  V6.0 migration always disables the old deployment manager. In V6.1, you have the option of not doing this by specifying no. If you choose `Y` (Yes), the deployment manager is stopped and its serverindex.xml file is renamed so that the server cannot be restarted.

- Application migration preference

  In V6.1, you now have an option to migrate your applications. Your choices are:

  - `Y` migrates all applications by reinstalling them to the default location in the new installation.
  - `S` does not copy any applications, but instead creates scripts that you can run later to install those applications.
  - `P` assumes the applications are installed in a directory outside of the installation directory. The applications are not copied, but are left where they are. The new installation now has the applications installed by virtue of pointing to their current location.
  - `N` causes the applications to be lost because they are not installed in the new installation.

- Temporary directory location

  The default temporary directory is /tmp/migrate. You can change this default if you need more space for temporary files by specifying a different directory in a larger file system.

15.3.2 New SAF profiles

V6.1 introduces new security functions that require you to manually configure SAF profiles. You can perform this SAF profile creation at any time before you run the migration jobs. These are the new profiles you need to configure:

- STARTED profiles

  If you changed the procedure names for the daemon and controller in Figure 15-1 on page 327, then you must define a new STARTED profile for each.

- TRUSTEDAPPS facility profile

  If you have the node property Enabled Trusted Applications set, then you must define the facility class profile TRUSTEDAPPS.

- EJBROLE profiles

  Two additional administrator roles have been added in V6.1, which require two additional EJBROLE profiles.

Examples of commands to define the required SAF profiles are in the article titled *System Authorization Facility classes and profiles* at this address:

```
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.web
sphere.zseries.doc/info/zseries/ae/csec_safclasses.html
```

15.3.3 Modifying daemon JCL

WebSphere Application Server for z/OS V6.1 requires that the daemon process be at the highest level of code of any of the servers that it manages on the same LPAR. It will be at the Version 6.1 level when the deployment manager is started. If the deployment manager manages nodes on the same LPAR in a mixed-cell environment, the daemon STARTED JCL procedure for both the deployment manager and the down-level nodes must have both the Version 6.1 libraries and those of the highest level of the down-level nodes in STEPLIB.

Here we use an example of migrating from V5.1 to V6.1. Figure 15-3 shows an example of how you would change the JCL for the daemon to include STEPLIB statements for both the V6.1 and V5.1 libraries.

```
//STEPLIB  DD DSN=WAS610.SBBOLD2,DISP=SHR
//         DD DSN=WAS610.SBBOLOAD,DISP=SHR
//         DD DSN=WAS610.SBBOLPA,DISP=SHR
//         DD DSN=WAS510.SBBOLD2,DISP=SHR
//         DD DSN=WAS510.SBBOLPA,DISP=SHR
```

Figure 15-3 Example daemon JCL after migration

Deciding which STEPLIB statements must be modified is determined by these rules:

- The daemon process must be running at the highest level of any node, which effectively means that it must be running at V6.1 code level to match whichever node you have migrated.
- The daemon process must have access to the libraries of the highest non-V6.1 code level. In our example, this happens to be V5.1.

After you have changed the JCL, you would restart the daemon process to pick up the V6.1 libraries.

15.3.4 Mixed-version cell restriction for z/OS

If you are migrating a cell at V5.0.2, you must migrate all the nodes in the cell on the same LPAR at the same time. A V6.1 daemon cannot support any nodes at the V5.0.2 level. This means that if you are migrating a V5.0.2 cell, the deployment manager and all of the nodes in the same LPAR must be migrated at the same time. Those nodes cannot participate in a mixed-version cell.

Figure 15-4 illustrates the forbidden configuration. SYSA is an LPAR with a cell consisting of a deployment manager, a daemon, and a node. Before the migration, the entire cell is at the V5.0.2 level. For illustration purposes, you could migrate the deployment manager to V6.1 and this would work. However, you could never restart the node, because the daemon, which is also at the V6.1 level as a result of deployment manager migration, cannot support a node at V5.0.2. Thus the configuration in Figure 15-4 is invalid.

If you have the situation represented by Figure 15-4, then you solve your problem by migrating the server node right after you migrate the deployment manager. You cannot run the node at V5.0.2, but you can certainly migrate it to V6.1 and then it will run.

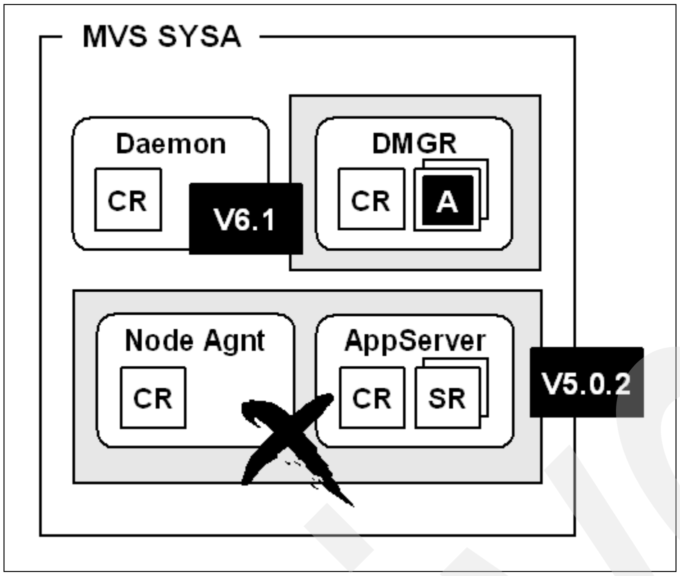


Figure 15-4 Migrating nodes in the same LPAR

If you have other nodes of the same cell that are in a different LPAR from the deployment manager, then these nodes do not have to be migrated at the same time as deployment manager. The nodes in the different LPAR are able to participate as mixed-version nodes in the cell.

Figure 15-5 illustrates the case of a V5.0.2 cell spanning two LPARs. SYSA is one LPAR with the deployment manager and a node, just like the previous example. SYSB is the second LPAR with a single node. SYSA must have all its nodes migrated to V6.1 at the same time. However, once migrated, the node in SYSB can continue to run. And that node can be restarted and can continue to run at V5.0.2 indefinitely. The difference is that SYSB has its own daemon, which remains at V5.0.2

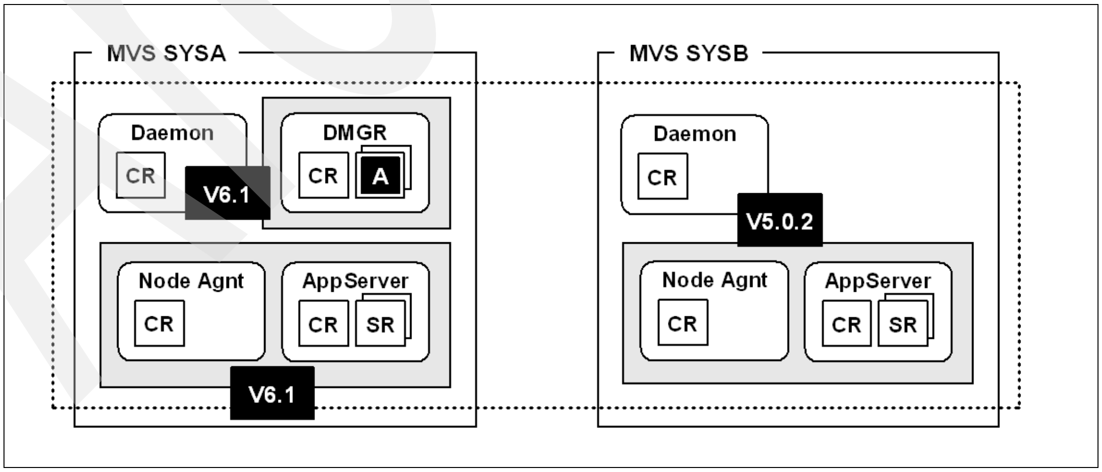


Figure 15-5 Migrating nodes which span LPARs

However, migrating a V5.1 cell does not have this same restriction. You can freely have a V5.1 mixed-version nodes participate in the V6.1 cell without regard to LPAR topology.

15.4 Installation

The installation and customization process for V6.1 is very similar to that for V6.0. There are minor changes to the ISPF customization screens in the areas of security. A new feature for V6.1 is the ability to replace the ISPF customization with a customization program that runs on Windows or Linux.

15.4.1 ISPF customization

There are several minor changes to the ISPF screens that perform installation customization, but the most significant is the customization of the initial security settings, alternatively called out-of-the-box security. This initial security setting defines an administrative user who is permitted access to the administrative console.

You have the choice to:

- Configure the z/OS operating system to use an SAF-compliant security system for defining the administrative user.

  The z/OS security product option uses the z/OS system's SAF-compliant security system, such as IBM RACF, to define WebSphere Application Server administrative users. Note that this option does not replace security domain configuration. You must always perform this configuration regardless of your choices for initial security. See 13.2.2, "ISPF customization dialogs" on page 278, and Figure 13-54 on page 279 for configuration of the security domain.

  An administrator user ID and an unauthorized user ID are created in the security database. For this security option, you must decide whether to set a security domain name, and choose an administrator user ID and an unauthenticated (guest) user ID.

  The Security domain identifier is optional and is used to distinguish between APPL or EJBROLE profiles based on security domain name. You specify an alphanumeric security domain name of one to eight characters. Internally, this sets `SecurityDomainType` to the string `cellQualified`. All servers in the cell prepend the security domain name you specify to the application-specific J2EE role name to create the SAF EJBROLE profile for checking. The security domain name is not used, however, if role checking is performed using WebSphere Application Server for z/OS bindings. The security domain name is also used as the APPL profile name, and is inserted into the profile name used for CBIND checks. The RACF jobs that the Customization Dialog generates create and authorize the appropriate RACF profiles for the created nodes and servers. If you do not want to use a security domain identifier, leave this field blank.

  The WebSphere Application Server administrator User ID is the initial WebSphere Application Server administrator. It must have the WebSphere Application Server configuration group as its default UNIX System Services group. The UNIX System Services UID number for the administrator user ID is specified here, and must be a unique numeric value between 1 and 2,147,483,647.

  The WebSphere Application Server unauthorized user ID is associated with unauthenticated client requests. It is sometimes referred to as the "guest" user ID. It should be given the RESTRICTED attribute in RACF, to prevent it from inheriting UACC-based access privileges. The UNIX System Services UID number for the user ID is specified here and is associated with unauthenticated client requests.

- Configure WebSphere Application Server to define the administrative user.

  When this option is chosen during customization, a file-based user registry is created in the configuration file system.

An administrator user ID (and an optional samples user ID and group) are added to the file-based user registry.

The administrator user ID is added to the list of authorized console users.

Self-signed digital certificates for servers are created in the configuration file system automatically by WebSphere Application Server.

- Bypass initial security and have administrative security disabled.

For more information about out-of-the-box security, see the article titled *Initial security configuration* at this address:

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

15.4.2 Graphical customization on your workstation

V6.1 introduces a new method of creating the server customization jobs. The Profile Management Tool (PMT) for the distributed platforms, discussed in 14.3.2, “Creating profiles” on page 310, has a sibling that runs on either Windows or Linux workstations. The PMT for WebSphere Application Server for z/OS is part of the Application Server Toolkit that ships with the product. You first install the Application Server Tookit and then open the PMT for z/OS perspective. From there, you sequence through a wizard that prompts you for the appropriate customization fields. You must have the same level of knowledge of customization that you have to perform ISPF customization. The advantage is that you can perform the customization that precedes running the jobs on your workstation.

We recommend you eventually learn this customization method. IBM has declared that the ISPF means of server customization is deprecated and eventually will be removed.

For information on how to install and run PMT, see the article titled *Using the Profile Management tool* at this address:

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

Also see the show-me demonstration of PMT customization in the IBM Education Assistant at this address:

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1z/InstallationAndMigration.html

15.5 Removals and deprecations specific to the z/OS platform

There are two features that have been removed or deprecated that are only features in the WebSphere Application Server for z/OS product.

15.5.1 DB2 for z/OS Local JDBC Provider

WebSphere Application Server for z/OS V6.1 does not support integration with the legacy DB2 JDBC Driver. Prior versions of WebSphere Application Server for z/OS use the DB2 for zOS Local JDBC Provider (RRS). You should migrate to using the DB2 universal JDBC driver. For detailed instructions on how to migrate to the universal driver, consult the whitepaper titled *JDBC Migration White Paper and Utility for DB2 on z/OS* at this address:

http://www-1.ibm.com/support/docview.wss?uid=swg27007826

15.5.2 z/OS System SSL

System SSL configurations for all server types except the daemon process are deprecated. You should convert your scripts to use JSSE instead. Beginning in V6.1, the default customization for servers uses JSSE for all process types except daemon. Since the daemon is native code rather than Java code, it continues to use System SSL. For more information see the article titled *SSL configurations collection* at this address:

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

A

Prerequisite software

Before you can upgrade IBM Rational Developer or WebSphere Application Server to V6, you need to do additional planning regarding other software packages that may need to be updated. This appendix is a summary of the various versions of software that are supported.

Prerequisites for IBM Rational Developer V6.0

IBM Rational Developer has dependencies on the host operating system, but is otherwise independent. The hardware requirements vary, but plan on at least

Consult this document for the most recent information about prerequisites for IBM Rational Developer:

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

Operating systems

The following list shows which operating systems are supported for the Rational Developer product line. This llist also applies for WebSphere Application Server Toolkit.

- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Standard Server
- Microsoft Windows XP Professional
- Microsoft Windows 2003 Standard Server
- Microsoft Windows 2003 Enterprise Server
- Red Hat Enterprise Linux WS 3.0 IA32 architecture
- SUSE Linux Enterprise Server 9.0 IA32 architecture

Prerequisites for WebSphere Application Server V6.0

Hardware requirements vary, but as a rough rule of thumb, plan on having 1GB of memory, 2GB of free disk space, and a processor that was purchased within the past 2 to 3 years so that you have enough processor power. You could certainly use a system that is older, but the processor power is what you need to investigate. Consult this document for detailed information about prerequisites for WebSphere Application Server Version 6:

http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27007251

Operating systems

Table A-1 shows which operating systems are supported. The operating systems which are marked X in the column title *Supported for development only* are intended by the operating system vendor to be a workstation operating system rather than a server operating system. Operating systems intended for workstations may not have the capability to perform to the same capacity and robustness as would an operating system for a server. Therefore, operating systems intended for workstations are not supported for production server environments. Such a configuration is supported for development use for development testing and prototyping.

Table A-1 Supported operating system versions V6.0

Operating System	Supported for development only	Supported for production and development
AIX 5L™ 5.1 with maintenance 5100-05		X
AIX 5L 5.2 with maintenance 5200-03		X

Operating System	Supported for development only	Supported for production and development
AIX 5L 5.3		X
HP-UX 11i with quality pack June 2004		X
Red Hat Enterprise Linux AS 3.0 Update 2 or 3		X
Red Hat Enterprise Linux 3.0 WS/ES Update 2 or 3	X	
SuSE Linux Enterprise Server 8 SP3		X
SuSE Linux Enterprise Server 9		X
Solaris 8 / Recommended Patch Cluster of June 2004		X
Solaris 9 / Recommended Patch Cluster of June 2004		X
Windows 2000 Advanced Server SP4		X
Windows 2000 Server SP4		X
Windows 2000 Professional SP4	X	
Windows Server® 2003, Datacenter		X
Windows Server 2003, Enterprise		X
Windows Server 2003, Standard		X
Windows XP Professional SP1a	X	
z/OS 1.4		X

Database servers

The supported database servers, necessary to perform Enterprise Java Bean persistence, are the following:

- Cloudscape 5.1.60.17
- IBM DB2 for z.OS v7 or v8
- IBM DB2 for iSeries 5.2 or 5.3
- IBM DB2 Connect™ 8.1 FP7a or 8.2
- IBM DB2 Enterprise Server Edition 8.1 FP7a or 8.2
- IBM DB2 Express 8.1 FP7a or 8.2
- IBM DB2 Workgroup Server Edition 8.1 FP7a or 8.2
- IBM DB2 Information Integrator 8.1 FP7a or 8.2
- IBM Informix® Dynamic Server 9.3 or 9.4
- Oracle 8i Standard or Enterprise Release 3 8.1.7.4
- Oracle 9i Standard or Enterprise Release 2 9.2.0.4
- Oracle 10g Standard or Enterprise Release 1 10.1.0.3
- Microsoft SQL Server Enterprise 2000 SP 3a
- Sybase Adaptive Server Enterprise 12.0.0.8, 12.5.1 or 12.5.2

JDBC driver managers

Table A-2 shows the supported JDBC drivers.

Table A-2 Supported JDBC drivers

JDBC driver	Which database connects	JDBC driver type
DataDirect Sequelink 5.4	Microsoft SQL Server	3
DataDirect Connect 3.4	Microsoft SQL Server	4
IBM WebSphere embedded Sequelink 5.4	Microsoft SQL Server	3
IBM WebSphere embedded Connect3.4	Microsoft SQL Server	4
Microsoft SQL Server 2000 JDBC Driver 2.2.0037	Microsoft SQL Server	4
IBM DB2 Universal JDBC 2.2. or 2.3	IBM DB2	2 or 4
IBM DB2 Legacy CLI JDBC	IBM DB2	2
IBM Informix JDBC 2.21 JC6	IBM Informix	4
Oracle 9i JDBC OCI	Oracle 8i and 9i	2
Oracle 9i JDBC Thin		4
Oracle 10g JDBC OCI	Oracle 9i and 10g	2
Oracle 10g JDBC Thin		4
Sybase jConnect 5.5 EBF 11656	Sybase Adaptive Server	4
OS/400® Java Toolbox JDBC Driver 4.3	DB2 on iSeries	4

Web servers

The following Web servers are supported:

- Apache Server 2.0.49
- IBM HTTP Server 2.0.47.1
- IBM HTTP Server 6.0
- Internet Information Services 5.0 or 6.0
- Lotus® Domino® Enterprise Server 6.0.3 or 6.5.1
- Sun Java System Web Server 6.0 SP7 and 6.1 SP1

Prerequisites for WebSphere Application Server V6.1

Hardware requirements vary, but as a rough rule of thumb, plan on having 1GB of memory, 2GB of free disk space, and a processor that was purchased within the past 2 to 3 years so that you have enough processor power. You could certainly use a system that is older, but the processor power is what you need to investigate.

Consult this document for detailed information about prerequisites for WebSphere Application Server Version 6:

http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27007651

Operating systems

Table A-1 shows which operating systems are supported. The operating systems which are marked X in the column title *Supported for development only* are intended by the operating system vendor to be a workstation operating system rather than a server operating system.

Operating systems intended for workstations may not have the capability to perform to the same capacity and robustness as would an operating system for a server. Therefore, operating systems intended for workstations are not supported for production server environments. Such a configuration is supported for development use for development testing and prototyping.

Table A-3 Supported operating system versions V6.1

Operating System	**Supported for development only**	**Supported for production and development**
AIX 5L 5.2 with maintenance 5200-07		X
AIX 5L 5.3 with maintenance 5300-04-01		X
HP-UX 11iv2 with quality bundle March 2006		X
Red Hat Enterprise Linux AS/ES 3.0 Update 5 or 6		X
Red Hat Enterprise Linux AS/ES 4.0 Update 2		X
Red Hat Enterprise Linux 3.0 WS Update 5 or 6	X	
Red Hat Enterprise Linux 4.0 WS Update 2	X	
SuSE Linux Enterprise Server 9 with SP2 or SP3		X
Solaris 9 with latest Recommended Patch Cluster		X
Solaris 10 with latest Recommended Patch Cluster		X
Windows 2000 Advanced Server SP4		X
Windows 2000 Server SP4		X
Windows 2000 Professional SP4	X	
Windows Server 2003, Datacenter with SP1		X
Windows Server 2003, Enterprise with SP1		X
Windows Server 2003, Standard with SP1		X
Windows XP Professional SP2	X	
z/OS 1.6		X

Database servers

The supported database servers necessary to perform Enterprise Java Bean persistence for V6.1 are the following:

- Cloudscape 10.1
- IBM DB2 for z.OS v7 or v8
- IBM DB2 for iSeries 5.2, 5.3, or 5.4
- IBM DB2 Connect 8.1 FP7a or 8.2
- IBM DB2 Enterprise Server Edition 8.2 FP4
- IBM DB2 Express 8.2 FP4
- IBM DB2 Workgroup Server Edition 8.2 FP4
- IBM WebSphere Information Integrator 8.2 FP4
- IBM Informix Dynamic Server 9.4 - C7W1 or 10.0 - C4

- Oracle 9i Standard or Enterprise Release 2 9.2.0.7
- Oracle 10g Standard or Enterprise Release 1 10.1.0.4
- Oracle 10g Standard or Enterprise Release 2 10.2.0.1 or 10.2.0.2
- Microsoft SQL Server Enterprise 2000 SP4
- Microsoft SQL Server Enterprise 2005
- Sybase Adaptive Server Enterprise 12.5.2 or 15.0

JDBC driver managers

Table A-2 shows the supported JDBC drivers.

Table A-4 Supported JDBC drivers for V6.1

JDBC driver	Which database connects	JDBC driver type
DataDirect Sequelink 5.5	Microsoft SQL Server	3
DataDirect Connect 3.5	Microsoft SQL Server	4
IBM WebSphere embedded Sequelink 5.5	Microsoft SQL Server	3
IBM WebSphere embedded Connect 3.5	Microsoft SQL Server	4
Microsoft SQL Server 2000 JDBC Driver 2.2.0040	Microsoft SQL Server	4
IBM DB2 Universal JDBC 2.8	IBM DB2	2 or 4
IBM DB2 Legacy CLI JDBC (deprecated)	IBM DB2	2
IBM Informix JDBC 3.0 JC3	IBM Informix	4
Oracle 9i JDBC OCI	Oracle 9i	2
Oracle 10g	Oracle 9i and 10g	2
Sybase jConnect 5.5 EBF 11656 or 6.0 EBF	Sybase Adaptive Server	4
OS/400 Java Toolbox JDBC Driver 4.8	DB2 on iSeries	4

Web servers

The following Web servers are supported for V6.1:

- Apache Server 2.0.54
- IBM HTTP Server 6.0.2 or 6.1
- Internet Information Services 5.0 or 6.0
- Lotus Domino Enterprise Server 6.5.4 or 7.0
- Sun Java System Web Server 6.0 SP9 and 6.1 SP3

B

Z/OS preparation steps

Before you can begin configuration of WASZOS, you must first satisfy the prerequisites for z/OS.

Preparing the z/OS operating system to install WebSphere

In order to run a WebSphere for z/OS Standalone Server, certain MVS system programming steps must be taken. A checklist is generated when you generated the customized jobs for the Standalone Server

The following is a summarized version of what's found in the BBOSSINS member when the Standalone Server's customized jobs are generated.

RRS enabled and operational

Insure that RRS be present prior to starting your application server.

WLM dynamic application environment validation

Insure you have WLM APAR OW54622 installed. Also insure that dynamic WLM application environments are present.

Update BLSCUSER

The instructions read:

In order to use the IPCS support provided by the product, append the contents of this member to the `BLSCUSER` member in your `IPCSPARM` or system `PARMLIB` data sets.

The instructions refer you to a member generated into the Standalone Server's CNTL data set called BBOIPCSP.

Update xxSCHED

The instructions read:

In order to set the correct program properties for the WebSphere for z/OS run-time executables, append the contents of this member to the `SCHEDxx` member in your system `PARMLIB` concatenation.

The instructions refer you to a member generated into the Standalone Server's CNTL data set called `BBOSCHED`.

APF authorization

Ensure the following data sets are APF-authorized:

- xxx.SBBOLPA
- xxx.SBBOLOAD
- xxx.SBBOLD2
- yyy.SCEERUN
- yyy.SCEERUN2

- zzz.SGSKLOAD

where:

xxx is the high-level qualifier of the SMP/E target data sets for WebSphere

yyy is the high-level qualifier for the SCEERUN and SCEERUN2 data sets

zzz is the high-level qualifier for the SGSKLOAD data set

Add these data sets to your PROGxx or IEAAPFxx PARMLIB members, as appropriate, ensuring you specify the correct VOLSER values.

SMF 120 records

If you want to collect the SMF120 records created by the run-time servers, update SMFPRMxx via the following:

- Update the SYS or SUBSYS(STC,...) statement for started tasks to include the 120 record.
- (optional) You can specify designated subtypes 1-6.

For example:

```
SUBSYS(STC,EXITS(IEFU29,IEFACTRT),INTERVAL(SMF,SYNC),TYPE(0,30,70:79,88,89,120,245))
```

For details on the SMF records, see related topics in the WebSphere for z/OS Information Center at:

http://www.ibm.com/software/webservers/appserv/zos_os390/library/

Automount the stand-alone server's configuration HFS

Update your active BPXPRMxx member to have the HFS you created for the stand-alone server auto-mounted in read/write mode at the proper configuration mount point. The BBOSSINS instruction member generated for the stand-alone server will have the information, or you can positively determine the proper values by doing the following:

- Inspect the BBOWCHFS job in the stand-alone server's CNTL target data set. That's where the HFS is allocated. You can determine the HFS data set name there.
- Inspect the BBOWCPYD member in the DATA target data set. That's called by BBOWCHFS, and that's where the mount point will be specified.

Important: It is critically important that you mount the configuration HFS at the same location as specified in the configuration of the standalone server. The value specified there creates a key “HOME” variable. Mounting the configuration HFS somewhere else will result in the server failing to start.

An example of a mount command in the BPXPRMxx member:

```
MOUNT FILESYSTEM('<your_config_data_set>')
  MOUNTPOINT('<your_config_mount_point>')
   TYPE(HFS)
   MODE(RDWR)
```

Reserve TCP ports

The BBOSSINS instruction member suggests that the TCP ports to be used by the Standalone Server be reserved. A member is supplied called BBOTCPIP that provides the instructions that can be placed in the TCP/IP profile member. However, this is not a strict technical requirement.

MAXFILEPROC value greater or equal to 2000

WebSphere for z/OS regions open a large number of files (more than 1024). Make sure your BPXPRMxx PARMLIB member(s) specify a value of MAXFILEPROC that is greater than or equal to 2000. Use the following MVS console command to see your current MAXFILEPROC setting:

```
D OMVS,OPTIONS
```

Update CFRM policy

Prior to using log streams that have been indicated as CF-resident, you must update the CFRM policy to define the structures to be used. Tailor member BBOWCFRM (found in the CNTL target data set) to define the log streams.

C

Discarding z/OS configuration

If you need to clean up your configuration of WebSphere Application Server for z/OS, the following section tells you the command to restore your system to the state before you began configuration.

Discarding configuration and cleaning up

Follow these steps to clean up your configuration to its starting state:

Unmount HFS and delete data set, delete mount point

1. Stop the Daemon server if it's still up:

   ```
   /P xxDMN
   ```

   where "xx" is your 2-character cell identifier. You need the proper authority to execute that command. Superuser or READ access to BPX.SUPERUSER.FACILITY.

2. Unmount the HFS with the following TSO command:

   ```
   UNMOUNT FILESYSTEM('OMVS.WAS.xxCELL.CONFIG.HFS')
   ```

   where "xx" is your two character cell identifier.

3. Delete the OMVS.WAS.xxCELL.CONFIG.HFS data set, where "xx" is your 2-character cell identifier.

   If you're updated BPXPRMxx to automount that HFS file system, you may consider removing that automount command from BPXPRMxx. Leave it there if you plan on simply recreating a new cell with the same HFS file system data set name and mount point.

4. Delete the /wasv6config/xxCELL mount point. Be sure nobody else is sharing the /wasv6config portion of that before you delete it.

Delete member copied by BBOWCPY1 job

1. Browse the BBOWCPY1 job that's found in your xxCELL.STAND.CNTL data set, where "xx" is your 2-character cell identifier. Note how all that job does is copy members out of either the CNTL or DATA data sets into PROCLIB, PARMLIB or SYSEXEC. It renames the members as it copies them.

   Delete the members from the various data sets shown in Table 15-1, where "xx" is your two-character cell identification number. Be careful with the CTIBBO60 member and the BBORBLOG member. They are not specific to your cell, but generic to WebSphere for z/OS V6 in general. If others are using WebSphere on this MVS image, you should leave those members and delete only those members that are specific to your cell. If you plan on recreating the cell with the same two character cell ID, then don't bother deleting these members. The BBOWCPY1 job will just overwrite the members.

Table 15-1 Data sets to clean up

Dataset	Member
PARMLIB	CTIBBO60
PROCLIB	xxACR
PROCLIB	xxACRZ
PROCLIB	xxASR
PROCLIB	xxASRZ
PROCLIB	xxCRA
PROCLIB	xxDMN
PROCLIB	xxDMNZ

Dataset	Member
PROCLIB	xxOW6SH
SYSEXEC	BBORBLOG

Clean up RACF database

Remove the following RACF objects, where xx means your two character cell identifier:

- User IDs
  - xxACRU
  - xxADMIN
  - xxADMSH
  - xxASRU
  - xxCRAU
  - xxDMNU
  - xxGUEST
- Groups
  - xxCFG1
  - xxCLGP
  - xxSR1
- STARTED - which requires a SETROPT refresh
  - xxACR.*
  - xxDMN.*
  - xxOW6SH.*
  - xxSR01AA.*
  - xxSR01AS.*
- APPL - requires a SETROPT refresh
  - xxCELL
- CBIND - requires a SETROPT refresh
  - CB.BIND.xxCELL.xxSR01
  - CB.xxCELL.xxSR01
  - CB.BIND.xxCELL.BBO*
  - CB.BIND.xxCELL.*
  - CB.xxCELL.BBO*
  - CB.xxCELL.*
- SERVER - requires a SETROPT refresh
  - CB.*.xxSR01
  - CB.*.xxSR01.*
  - CB.*.xxSR01ADJUNCT
  - CB.*.xxSR01ADJUNCT.*
- EJBROLE - requires a SETROPT refresh

- xxCELL.administrator
- xxCELL.configurator
- xxCELL.monitor
- xxCELL.operator
- xxCELL.CosNamingCreate
- xxCELL.CosNamingDelete
- xxCELL.CosNamingRead
- xxCELL.CosNamingWrite

Delete CNTL, DATA, SDCFG and SAVECFG data sets

1. Delete the following target data sets:
   - xxCELL.STAND.CNTL
   - xxCELL.STAND.DATA
   - xxCELL.SECURITY.CNTL
   - xxCELL.SECURITY.DATA
2. Delete the following saved variables data sets:
   - xxCELL.STAND.SAVECFG
   - xxCELL.SECURITY.SDCFG

   Think twice before deleting these. You may want to hold onto these and use them as input to initialize a new cell's configuration – that is, load these variables and change only a few things rather than having to retype everything.

D

Bibliography

The following references were used as sources for much of the work we have done here. You should consider consulting these references to gather more detail on the subject matter you need to explore.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 351. Note that some of the documents referenced here may be available in softcopy only.

- *Migrating to WebSphere V5.0 An End-to-End Migration Guide,* SG24-6910

  You should consult this redbook if you are considering a migration directly from V3.5 to V6.

- *Migrating Applications from WebSphere for z/OS V4 and V3.5 to V5*, SG24-7044

  You should consult this redbook if you use the WebSphere Application Server on the z/OS operating system and intend to migrate.

- *WebSphere Application Server V6 Technical Overview*, REDP-3918
- *WebSphere Application Server V6 System Management and Configuration Handbook*, SG24-6451

  This redbook has considerably more detailed information about the system administration of V6. It also covers new V6 functionality which takes you beyond the realm of migration of existing applications.

- *WebSphere Application Server V6 Scalability and Performance Handbook*, SG24-6392
- *WebSphere Application Server V6: Web Services Development and Deployment*, SG24-6461
- *WebSphere Application Server V6 Security Handbook*, SG24-6316
- *WebSphere Application Server V6: High Availability Solutions*, REDP-3971

WebSphere Application Server Version 6 Information Center

A great deal of the technical information in this redbook is based on information found in the online documentation guide WebSphere Application Server Version 6 Information Center. We also refer to this guide as the InfoCenter. You can find the guide at this address:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp

We cite URLs in the book text that take you directly to the specific article. We hope that the URL will remain accurate and not change. In the event that the URL changes, you can search for the article by pasting the article title in the Search bar in the upper left part of the Web browser window.

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM developerWorks® - Technical resources for the WebSphere software platform

  The top level address for searching articles is:

  http://www.ibm.com/developerworks/websphere

  These are articles on the IBM developerWorks site that we cite in the text and we otherwise recommend:

  - Meet the Experts: Wayne Beaton on WebSphere Application Server migration

    http://www.ibm.com/developerworks/websphere/library/techarticles/0403_beaton/beaton.html

  - The Ideal WebSphere Development Environment

    http://www.ibm.com/developerworks/websphere/techjournal/0312_beaton/beaton.html

  - The top 10 (more or less) J2EE best practices

    http://www.ibm.com/developerworks/websphere/techjournal/0405_brown/0405_brown.html

  - Get the message: Messaging in J2EE 1.4

    http://www-106.ibm.com/developerworks/java/library/j-getmess/

  - Plants by WebSphere Sample

    http://www.ibm.com/developerworks/websphere/library/samples/WASV501/plantsby.html

- Sun J2EE and J2SE documentation

  - J2EE 1.4 compatibility issues

    http://docs.sun.com/source/819-0077/J2EE.html

  - J2SE 1.4 compatibility issues

    http://java.sun.com/j2se/1.4/compatibility.html

  - J2EE 1.4 specification

    http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf

  - J2EE Deprecated APIs

    http://java.sun.com/j2ee/1.4/docs/api/deprecated-list.html

  - JSEE Deprecated APIs

    http://java.sun.com/j2se/1.4.2/docs/api/deprecated-list.html

  - JDBC 3.0 specification

    http://java.sun.com/products/jdbc/download.html

  - J2EE Servlet 2.4 specification

    http://java.sun.com/j2ee/1.4/download.html#platformspec

- IBM WebSphere Support and downloads

  http://www.ibm.com/software/websphere/support/

  - Supported Hardware and Software for WebSphere Application Server Version 6

    http://www.ibm.com/software/webservers/appserv/doc/v60/prereqs/prereq60.html

  - WebSphere MQ Family support

    http://www.ibm.com/software/integration/mqfamily/support/summary/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Index

WebSphere Application Server V6 Migration Guide

(0.5" spine)
0.475"<->0.873"
250 <-> 459 pages

WebSphere Application Server V6 Migration Guide

Updated for the z/OS platform

Updated for V6.0.2 and V6.1

Practical migration examples

This IBM Redbook will assist you in the migration of your WebSphere Application Server installation. The end-to-end migration path includes the migration of the development environment, the test/integration environment, and the production environment using software engineering methodologies. This guide presents the best practices in migration strategy and planning, migration tools, and practical migration examples.

The following Version 6 products are covered:

- IBM Rational Application Developer for WebSphere Software
- IBM Rational Web Developer for WebSphere Software
- WebSphere Application Server
- WebSphere Application Server Express
- WebSphere Application Server Network Deployment
- WebSphere Application Server for z/OS

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6369-01 ISBN 0738492450