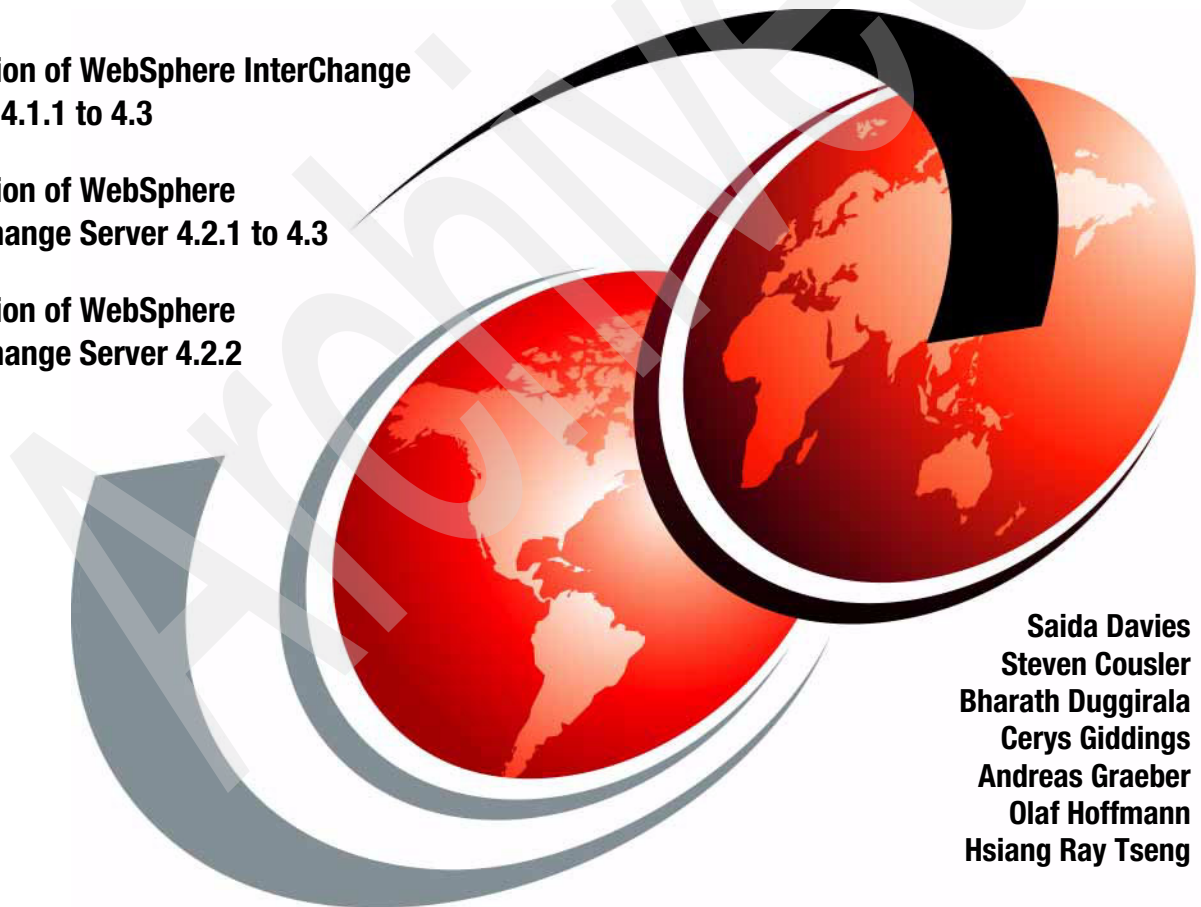


# WebSphere InterChange Server Migration Scenarios

Migration of WebSphere InterChange  
Server 4.1.1 to 4.3

Migration of WebSphere  
InterChange Server 4.2.1 to 4.3

Migration of WebSphere  
InterChange Server 4.2.2



Saida Davies  
Steven Cousler  
Bharath Duggirala  
Cerys Giddings  
Andreas Graeber  
Olaf Hoffmann  
Hsiang Ray Tseng





International Technical Support Organization

## **WebSphere InterChange Server Migration Scenarios**

February 2005

Archived

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

## **First Edition (February 2005)**

This edition applies to:

AIX:

Version 5, Release 2, Modification 2 of Operating System AIX  
Version 5, Release 3, Modification 0, FixPack 7 of WebSphere MQ  
Version 7, Release 2, Modification 0, FixPack 6 of DB2  
Version 8, Release 1 Modification 0, FixPack 5 of DB2  
VisiBroker  
Version 4, Release 1, Modification 1.9 of WebSphere InterChange Server  
Version 4, Release 2, Modification 1.7 of WebSphere InterChange Server  
Version 4, Release 2, Modification 2.4 of WebSphere InterChange Server  
Version 4, Release 3 of WebSphere InterChange Server  
Version 1, Release 3, Modification 1 of IBM Object Request Broker (IBM JRE)  
Version 1, Release 4, Modification 2 of IBM Object Request Broker (IBM JRE)  
Version 3, Release 3, Modification 2 of C-compiler for stored procedures GNU gcc

Windows:

Microsoft Windows 2000 Professional, Servicepack 4 Operating System  
Version 5, Release 3, Modification 0, FixPack 3 of WebSphere MQ  
Version 5, Release 3, Modification 0, FixPack 5 of WebSphere MQ  
Version 5, Release 3, Modification 0, FixPack 7 of WebSphere MQ  
Version 7, Release 2, Modification 0, of DB2  
Version 8, Release 1, Modification 0, FixPack 2 of DB2  
Version 8, Release 1, Modification 0, FixPack 5 of DB2  
VisiBroker  
Version 4, Release 1, Modification 1.9 of WebSphere InterChange Server  
Version 4, Release 2, Modification 1.7 of WebSphere InterChange Server  
Version 4, Release 2, Modification 2.4 of WebSphere InterChange Server  
Version 4, Release 3, Modification 0 of WebSphere InterChange Server  
Version 4, Release 2, Modification 1.6 of Toolset  
Version 4, Release 2, Modification 1.7 of Toolset  
Version 4, Release 2, Modification 2.4 of Toolset  
Version 4, Release 3, Modification 0 of Toolset Version  
Version 1, Release 3, Modification 1 of IBM Object Request Broker (IBM JRE)  
Version 1, Release 4, Modification 2 of IBM Object Request Broker (IBM JRE)  
Version 1, Release 1, Modification 0 of C-compiler for stored procedures Microsoft .NET SDK & Framework  
Version 5, Release 0, Modification 0 of WebSphere Application Server  
Version 5, Release 0, Modification 2 of WebSphere Application Server

**© Copyright International Business Machines Corporation 2005. All rights reserved.**

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

# Contents

<b>Notices</b> .....	ix
Trademarks .....	x
<b>Preface</b> .....	xi
The team that wrote this redbook .....	xii
Become a published author .....	xv
Comments welcome .....	xvi
<b>Part 1. New features and migration planning</b> .....	1
<b>Chapter 1. Introduction</b> .....	3
1.1 The scope of this redbook .....	4
1.1.1 Intended audience .....	4
1.1.2 Overview of topics covered .....	5
1.1.3 What is not covered .....	7
1.1.4 Assumptions .....	8
1.2 Planning for the future .....	8
<b>Chapter 2. Product overview</b> .....	11
2.1 WebSphere Business Integration Server .....	12
2.1.1 IBM Business Integration Reference Architecture overview .....	12
2.1.2 Components of WebSphere Business Integration Server 4.3.0 .....	15
2.2 WebSphere InterChange Server .....	18
2.2.1 Collaborations .....	18
2.2.2 Business objects .....	19
2.2.3 Maps .....	19
2.2.4 WebSphere Business Integration Toolset .....	20
2.3 New features in WebSphere InterChange Server 4.3 .....	20
2.3.1 Move to Java 1.4.2 and Eclipse 2.1 .....	21
2.3.2 End-to-end security .....	21
2.3.3 Role-based access control .....	22
2.3.4 Hot deployment .....	24
2.3.5 Large-object support .....	24
2.3.6 JMS transport optimization .....	25
2.3.7 Database connection resilience .....	25
2.3.8 Fail Events Management API .....	25
2.3.9 Failed Event Manager .....	25
2.3.10 Dynamic Service Calls .....	25
2.3.11 Bidirectional language support .....	26

2.3.12	One Persistent Name Server hosts multiple ICSs and adapters . . .	26
2.3.13	Web services. . . . .	26
2.3.14	Remote service support for Integrated Test Environment. . . . .	26
2.3.15	Improved tooling facilities for Activity Editor, Map Designer, and Connector Configurator . . . . .	26
2.3.16	IBM Tivoli License Manager . . . . .	27
2.3.17	Linux support. . . . .	28
<b>Chapter 3.</b>	<b>What is new?</b> . . . . .	<b>29</b>
3.1	New features by release . . . . .	30
3.1.1	Changes from WebSphere InterChange Server V4.1.1 to V4.3 . . .	30
3.1.2	Changes from WebSphere InterChange Server V4.2.1 to V4.3 . . .	37
3.1.3	Changes from WebSphere InterChange Server V4.2.2 to V4.3 . . .	41
3.2	Product history . . . . .	43
3.2.1	Security features . . . . .	43
3.2.2	Availability features . . . . .	44
3.2.3	Usability, administrative, and serviceability features . . . . .	45
3.2.4	Scalability and performance features . . . . .	46
3.2.5	National Language Support features . . . . .	47
3.2.6	Rebranding and renaming features. . . . .	47
3.2.7	Development features . . . . .	48
3.2.8	Miscellaneous features . . . . .	48
3.3	System architecture changes . . . . .	49
3.3.1	Supported operating systems . . . . .	49
3.3.2	Supported databases . . . . .	50
3.3.3	Other supported software . . . . .	50
<b>Chapter 4.</b>	<b>General planning</b> . . . . .	<b>53</b>
4.1	Migration methods. . . . .	54
4.1.1	In-place database migration . . . . .	54
4.1.2	Without-in place database migration. . . . .	54
4.2	Hardware considerations. . . . .	55
4.3	Software considerations . . . . .	56
4.4	Human resource roles. . . . .	58
4.5	Time considerations . . . . .	60
4.5.1	Duration. . . . .	60
4.5.2	Schedule . . . . .	60
4.5.3	Status . . . . .	61
4.5.4	Other time considerations . . . . .	61
4.6	Fault planning . . . . .	61
4.7	Stage checklists . . . . .	61
<b>Part 2.</b>	<b>Migration steps</b> . . . . .	<b>65</b>

<b>Chapter 5. Technical impact of major design changes</b>	67
5.1 VisiBroker to IBM ORB change	69
5.1.1 IBM ORB setup	70
5.1.2 Using Server Access Interface with the IBM ORB	75
5.1.3 WebSphere MQ triggering replacing Object Activation Daemon	77
5.1.4 Setting up WebSphere MQ triggering	78
5.2 Tooling changes to WebSphere InterChange Server	88
5.2.1 WebSphere InterChange Server development model	89
5.2.2 Configuration files updated to XML	91
5.2.3 Eclipse platform	91
5.2.4 System Manager	92
5.2.5 System Monitor	93
5.2.6 Business Object Probes	95
5.2.7 Web-based flow and failed-flow management	95
5.2.8 Integrated Test Environment	95
5.2.9 Collaboration debugger	96
5.2.10 Activity Editor	96
<b>Chapter 6. Installation of WebSphere InterChange Server 4.3</b>	99
6.1 Overview	100
6.2 Pre-installation considerations	100
6.2.1 Hard disks and file systems	100
6.2.2 Installation packages	101
6.2.3 Cannot be installed in the same location as old versions	102
6.3 Installation on Windows	102
6.4 Installation on UNIX (AIX)	104
<b>Chapter 7. Project environment setup</b>	109
7.1 Hardware and software environment setup	110
7.1.1 AIX environment	110
7.1.2 Windows environment	112
7.2 Migrated test cases	114
7.2.1 Long Lived Business Process	114
7.2.2 CustomerSync collaboration including relationships	115
7.2.3 Connection pool	115
7.2.4 Collaboration group	115
7.2.5 Polymorphic map	115
7.2.6 Server Access Interface	116
7.2.7 Transactional collaboration	116
7.2.8 Failed flows with different transports	116
7.2.9 Schedules	116
7.2.10 Imported external code	116
7.3 Migrated runtime data	117

7.3.1 Incomplete flows . . . . .	117
7.3.2 Relationship table content . . . . .	117
7.3.3 Historical data for the Web-based System Monitor . . . . .	118
<b>Chapter 8. Migration procedures for WebSphere InterChange Server</b>	
<b>V4.1.1.9 to V4.3 . . . . .</b>	<b>119</b>
8.1 In-place database migration of V4.1.1.9 to V4.3 on Windows . . . . .	120
8.1.1 Pre-migration steps . . . . .	120
8.1.2 Migration steps . . . . .	123
8.1.3 Post-migration steps . . . . .	137
8.2 Without in-place database migration of V4.1.1.9 to V4.3 on Windows . . . . .	143
8.2.1 Pre-migration steps . . . . .	143
8.2.2 Migration steps . . . . .	146
8.2.3 Post-migration steps . . . . .	164
8.3 In-place database migration of V4.1.1.9 to V4.3 on AIX . . . . .	171
8.3.1 Pre-migration steps . . . . .	171
8.3.2 Migration steps . . . . .	174
8.3.3 Post-migration steps . . . . .	187
8.4 Without in-place database migration of V4.1.1.9 to V4.3.0 on AIX . . . . .	194
8.4.1 Pre-migration steps . . . . .	195
8.4.2 Migration steps . . . . .	197
8.4.3 Post-migration steps . . . . .	215
<b>Chapter 9. Migration procedures for WebSphere InterChange</b>	
<b>Server V4.2.1 to V4.3 . . . . .</b>	<b>223</b>
9.1 In-place database migration: V4.2.1 to V4.3 on Windows . . . . .	224
9.1.1 Pre-migration steps . . . . .	224
9.1.2 Migration steps . . . . .	227
9.1.3 Post-migration steps . . . . .	241
9.2 Without in-place database migration of V4.2.1 to V4.3 on Windows . . . . .	247
9.2.1 Pre-migration steps . . . . .	248
9.2.2 Migration steps . . . . .	251
9.2.3 Post-migration steps . . . . .	265
9.3 In-place database migration of V4.2.1 to V4.3 on AIX . . . . .	271
9.3.1 Pre-migration steps . . . . .	271
9.3.2 Migration steps . . . . .	274
9.3.3 Post-migration steps . . . . .	287
9.4 Without in-place database migration of V4.2.1 to V4.3 on AIX . . . . .	294
9.4.1 Pre-migration steps . . . . .	294
9.4.2 Migration steps . . . . .	297
9.4.3 Post-migration steps . . . . .	313
<b>Chapter 10. Migration procedures for WebSphere InterChange Server</b>	
<b>V4.2.2 to V4.3 . . . . .</b>	<b>321</b>



10.1 In-place database migration: V4.2.2 to V4.3 on Windows . . . . .	322
10.1.1 Pre-migration steps . . . . .	322
10.1.2 Migration steps . . . . .	325
10.1.3 Post-migration steps . . . . .	337
10.2 Without in-place database migration for V4.2.2 to V4.3 on Windows . .	344
10.2.1 Pre-migration steps . . . . .	344
10.2.2 Migration steps . . . . .	347
10.2.3 Post-migration steps . . . . .	361
10.3 In-place database migration for V4.2.2 to V4.3. on AIX . . . . .	367
10.3.1 Pre-migration steps . . . . .	367
10.3.2 Migration steps . . . . .	370
10.3.3 Post-migration steps . . . . .	382
10.4 Without in-place database migration of V4.2.2 to V4.3 on AIX . . . . .	389
10.4.1 Pre-migration steps . . . . .	389
10.4.2 Migration steps . . . . .	392
10.4.3 Post-migration steps . . . . .	409
<b>Part 3. Appendixes . . . . .</b>	<b>417</b>
<b>Appendix A. WebSphere InterChange Server migration checklist. . . . .</b>	<b>419</b>
Migration checklist for WebSphere ICS 4.1.1. . . . .	420
Windows: in-place database migration . . . . .	420
Windows: without in-place database migration . . . . .	424
AIX: in-place database migration . . . . .	428
AIX: without in-place database migration . . . . .	431
Migration checklist for WebSphere ICS 4.2.1. . . . .	436
Windows: in-place database migration . . . . .	436
Windows: without in-place database migration . . . . .	439
AIX: in-place database migration . . . . .	443
AIX: without in-place database migration . . . . .	446
Migration checklist for WebSphere ICS 4.2.2. . . . .	450
Windows: in-place database migration . . . . .	450
Windows: without in-place database migration . . . . .	453
AIX: in-place database migration . . . . .	456
AIX: without in-place database migration . . . . .	460
<b>Appendix B. Sample code . . . . .</b>	<b>465</b>
DDL code for WebSphere InterChange Server 4.1.1.9 . . . . .	466
<b>Appendix C. Additional material . . . . .</b>	<b>469</b>
Locating the Web material . . . . .	469
Using the Web material . . . . .	469
How to use the Web material . . . . .	470

<b>Abbreviations and acronyms</b> .....	471
<b>Related publications</b> .....	473
IBM Redbooks .....	473
Online resources .....	473
How to get IBM Redbooks .....	474
Help from IBM .....	474
<b>Index</b> .....	475

# 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:

AIX®	Lotus Notes®	Redpaper™
ClearCase®	Lotus®	Redbooks (logo)  ®
DB2 Universal Database™	Notes®	Tivoli®
DB2®	Parallel Sysplex®	WebSphere®
Everyplace®	pSeries®	z/OS®
IBM®	Rational®	
Informix®	Redbooks®	

The following terms are trademarks of other companies:

Adobe, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Interchange, Red Hat, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Enterprise JavaBeans, J2EE, Java, JavaBeans, JDBC, JDK, JRE, JSP, JVM, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Intel, Pentium, 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

IBM® WebSphere® InterChange Server (WebSphere ICS), a key component of IBM WebSphere Business Integration Server, provides the capability to enable heterogeneous business applications to exchange data. It addresses the need for efficiency and flexibility in automating and synchronizing business activities.

The focus for Release 4.3 is to deliver support for new operating systems, and include the reliability, availability, and scalability enhancements requested by customers, together with significant performance and security improvements. This release represents an important base level for consolidation of features since Release 4.1.1.

The aim of this book is to provide a step-by-step guide to migration to WebSphere InterChange Server 4.3 from the following versions:

- ▶ WebSphere InterChange Server 4.1.1
- ▶ WebSphere InterChange Server 4.2.1
- ▶ WebSphere InterChange Server 4.2.2

These migrations are demonstrated for each version with an existing WebSphere Business Integration Server scenario that we migrate end to end. Advice on planning and pre-migration tasks is provided, as is information to assist in understanding the new features of WebSphere Business Integration Server 4.3 and changes since the above previous versions.

This book has two sections: Part 1, “New features and migration planning” on page 1 gives a brief introduction to WebSphere InterChange Server and its new features. A history of WebSphere ICS is provided for each of the releases since 4.1.1. A section is also provided with recommended planning tasks for migrating a previous release of WebSphere InterChange Server to Version 4.3.

Part 2, “Migration steps” on page 65 provides a technical step-by-step guide to migrating a previous release of WebSphere InterChange Server to Version 4.3 on AIX® and Windows®. For each migration, the following tasks are discussed:

- ▶ Pre-migration steps
- ▶ Migration steps
- ▶ Post-migration and validation steps

A chapter is included with step-by-step instructions for the installation of WebSphere InterChange Server 4.3 for AIX and Windows. Another chapter gives an overview of the scenario used by the team for migration in this redbook.

## The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.



*From left, back row: Bharath, Olaf, Andreas  
Front row: Saida, Ray, Cerys, Steven*

**Saida Davies** is a Project Leader for the International Technical Support Organization (ITSO) and has 15 years of experience in IT. Saida has published several Redbooks® about various business integration scenarios. She has experience in the architecture and design of WebSphere MQ solutions, extensive knowledge of the z/OS® operating system, and detailed working knowledge of both IBM and Independent Software Vendors' operating system software. Her customer-facing role as a senior IT specialist with IBM Global Services included the development of services for z/OS and WebSphere MQ within the z/OS and Windows platforms. This covered the architecture, scope, design, project management, and implementation of the software on stand-alone systems or on systems in a Parallel Sysplex® environment. She has received Bravo Awards for her project contribution. She has a degree in Computer Science, and her background includes z/OS systems programming.

**Steven Cousler** is a Staff Software Engineer for the IBM Application Integration and Middleware (AIM) division in Research Triangle Park, NC. He provides level 2 technical support for IBM customers using WebSphere Business Integration products such as IBM WebSphere InterChange Server and IBM WebSphere Business Integration Adapters. During his 10 years with IBM, he has worked with customer and technical support issues related to IBM Payment Suite, IBM WebSphere Commerce Suite, WebSphere Application Server, IBM HTTP Server, DB2® UDB, Visual Age for Java™, and WebSphere Studio Application Developer Edition. Steve comes to WebSphere Business Integration support from positions in level 2 and technical sales support for WebSphere Commerce Suite. He helped produce IBM certification exams for WebSphere Commerce Suite V5.1 and has participated in a previous redbook residency concerning WebSphere Commerce Suite deployment.

**Bharath Duggirala** is a member of the World Wide support team for IBM WebSphere Business Integration and has been working at the Software Labs in Bangalore, India, since 2003. His area of expertise includes J2EE™, WebSphere Application Server, WebSphere Interchange™ Server, and WebSphere Business Integration Adapters. He is certified in IBM WebSphere Application Server V5.0. Bharath graduated in Electrical and Electronics Engineering from the College of Engineering, Andhra University, India.

**Cerys Giddings** is a Staff Software Engineer and has worked on the WebSphere MQ family of products since joining IBM Hursley in 2000. She is a team leader in the WebSphere MQ Test organization specializing in usability, documentation, and testing from a customer perspective. She has also been involved in the development of Ease of Use features and samples for the WebSphere Business Integration Message Broker product. Cerys has participated in producing the IBM Certified System Administrator for WebSphere Business Integration Message Broker V5 certification program. She is co-author of the redbook WebSphere Business Integration Message Broker Basics. Cerys

is also actively involved with IBM activities in the community, including being one of the coordinators of IBM's global community program MentorPlace and IT training for teachers. She has 10 years of experience in providing IT education and support, and holds a degree and Masters from the University of Wales, as well as the BCS Professional Examinations at Certificate and Diploma levels.

**Andreas Graeber** is a Software Support Specialist and has been working with the WebSphere InterChange Server and WebSphere Business Integration Adapters products for more than two years. Before joining his current department, Andreas was part of support for IBM Informix® Databases. He came into Information Technology in 1998 when he joined the support and training organization for a supplier of Digital Mockup Software, which is used by car manufacturing and aerospace industries. Andreas holds a University Diploma in Process Engineering.

**Olaf Hoffmann** is a Software Technical Support Specialist located in Mainz, Germany. He is the technical team leader for WebSphere InterChange Server and the WebSphere Business Integration Adapter in the WebSphere Business Integration EMEA Support Team, ITS, IBM Global Services. His areas of expertise include WebSphere Business Integration on distributed platforms, using WebSphere Business InterChange Server, WebSphere Business Integration Message Broker, WebSphere Business Integration Adapter, and WebSphere MQ, and he utilizes his experience to solve EMEA customer issues related to these products. In addition, he assists in critical client projects and training and coaching within IBM Global Services. Before joining Software Support, Olaf gained software development experience in the area of WebSphere MQ, WebSphere MQ Everyplace®, WebSphere MQ Integrator, and Lotus® Notes®. This included Java and C programming. He also ran his own PC Business and worked for two years as a Network Administrator. Olaf holds an Academy Diploma in Information Technology which is equivalent to a Master of Science degree.

**Hsiang Ray Tseng** is a member of the Level 2 Technical Support team for WebSphere Business Integration. He has supported the WebSphere InterChange Server and WebSphere Business Integration Adapters for the past two years since joining IBM in 2003. Besides directly helping customers resolve problems, Ray also develops, submits, and presents customer-requested changes and enhancements for development. He has specialized in the InterChange Server and Toolset components, and has participated in design reviews for current and future features for these components. Ray holds a Bachelor of Science degree in Computer Engineering and Master of Science degree in Electrical Engineering from Stanford University.



The International Technical Support Organization thanks the following people for their assistance in the planning of this project:

Subbu Gooty, WBI Server Development Manager, Software Group/Application Integration Middleware  
IBM Burlingame Lab, CA

Todd Martin, Consulting IT Specialist, Software Group/Application Integration Middleware  
IBM Burlingame Lab, CA

Mark Cox, WW Customer Support Manager for WBI, Software Group/Application Integration Middleware  
BM Burlingame Lab, CA

Sriram V Madapura, IT Specialist, Software Group/Application Integration Middleware  
IBM Burlingame Lab, CA

The ITSO and the redbook team thanks Weixin (Wilson) Xu for his participation during the first week of the residency and his valuable contribution, specifically in the planning of various migration scenarios. His deep knowledge of all versions of WebSphere InterChange Server that are covered in this book and the documentation he supplied saved the team valuable time. Weixin is Staff Engineer in WebSphere Business Integration Server Development Team, located at IBM Burlingame Lab, CA, USA, and his current role is Technical lead in WebSphere InterChange Server 4.3.

A special thanks from the project team to the following people for providing the high-quality material and making an important contribution to this redbook:

Alexander Ziegler, Manager SWG Education Central Region, Manager WBI Education EMEA  
IBM Germany

Christel Therkildsen, Training Coordinator  
IBM Germany

## **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 will team with IBM technical professionals, Business Partners, and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will 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](http://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](http://ibm.com/redbooks)

- ▶ Send your comments in an e-mail to:

[redbook@us.ibm.com](mailto: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

# New features and migration planning

This part provides an introduction to the scope and aims of the redbook. An overview of IBM WebSphere Business Integration and IBM WebSphere InterChange Server (WebSphere ICS) is given together with details of the new features in WebSphere InterChange Server 4.3.

Chapter 1, “Introduction” on page 3 provides an introduction to the book, with an overview of the contents and scope of the book.

Chapter 2, “Product overview” on page 11 includes an overview of WebSphere InterChange Server and the new features and enhancements that are available

for the 4.3 release. This chapter also provides a high-level overview of business integration that could be useful for less experienced users.

In Chapter 3, “What is new?” on page 29, users of previous versions can benefit from information about important features for migration since the release of IBM CrossWorlds 4.1.1, including a full product history. Technical details of the most significant design changes can be found in Chapter 5, “Technical impact of major design changes” on page 67.

In Chapter 4, “General planning” on page 53, guidance is given to the high-level planning considerations that are necessary for migration of earlier releases to Version 4.3.

# Introduction

This chapter provides an overview of the scope of the redbook, information about the intended audience, and the aspects of WebSphere InterChange Server (WebSphere ICS) migration covered in this book. In addition, information regarding the importance of migrating to Version 4.3.0 is provided.

## 1.1 The scope of this redbook

The aim of this redbook is to provide a step-by-step guide to migrate from selected versions of WebSphere InterChange Server and some related components to Version 4.3.0. Migrations are demonstrated by taking an existing WebSphere Business Integration Server scenario for each version and migrating it end to end. These scenarios have been tested to determine that the migration is successful:

- ▶ 4.1.1.9 → 4.3.0
- ▶ 4.2.1.7 → 4.3.0
- ▶ 4.2.2.4 → 4.3.0

We tested only on AIX and Windows platforms, but the general steps are applicable to other platforms. For more about other UNIX® platforms, refer to the *WebSphere InterChange Server System Installation Guide for UNIX* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left-side pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **UNIX** in the right-side pane.

Details of the specific scenarios covered in this redbook are given in Chapter 7, “Project environment setup” on page 109.

### 1.1.1 Intended audience

This redbook has two parts, and each part is directed at a different audience. The first part of the redbook is targeted at a management and architect level, and as such contains more high-level information and includes planning considerations. The second part of the book covers detailed technical considerations and migration steps, and is targeted at the user roles required to perform the actual migration tasks.

This book is not targeted at new users, and as such does not give information about using the product or other related components. Readers must refer to the product documentation for details about setting up a working configuration and using the product components.

## 1.1.2 Overview of topics covered

This section gives a short overview of the topics covered in each of the chapters in this book. A list is provided at the end of the section to give an overview of basic components of the scenario that are covered for the appropriate versions.

### **Chapter 2, “Product overview”**

This chapter gives a high-level introduction to the WebSphere Business Integration Server and a more detailed overview of the WebSphere ICS and its functions. It also covers the most significant new features in the 4.3.0 release of WebSphere ICS.

### **Chapter 3, “What is new?”**

This chapter provides a table of changed and enhanced features for WebSphere ICS and related components since Release 4.1.1, and the most significant changes in each version are discussed. A systems architecture perspective on the updates is also provided.

### **Chapter 4, “General planning”**

This chapter gives a non-technical guide to the planning considerations for the migration, including hardware and software considerations, required resources, and roles. We also offer fault plans and a stage diagram of the required steps.

### **Chapter 5, “Technical impact of major design changes”**

This chapter provides technical details about the major architectural and design changes since WebSphere InterChange Server 4.1.1 that affect customers who use any version before 4.2.2.

### **Chapter 6, “Installation of WebSphere InterChange Server 4.3”**

This chapter discusses the reasons for moving to WebSphere InterChange Server 4.3, and provides information about performing a straightforward installation of the product on AIX and Windows, including pre-installation considerations.

### **Chapter 7, “Project environment setup”**

This chapter details the scenario and technical configuration used in the following chapters of this book to perform the migration demonstration. Specific details of the hardware and software that are used in the demonstration are given, together with an overview of the data and configuration used in the scenario.

### **Chapter 8, “Migration procedures for WebSphere InterChange Server V4.1.1.9 to V4.3”**

This chapter provides detailed instructions for performing the migration from Version 4.1.1.9 to WebSphere ICS 4.3.0 using the scenario detailed in Chapter Chapter 7, “Project environment setup” on page 109. It includes guidance on tasks that must be carried out

both before and after migration. The chapter has separate sections for migration on Windows and AIX.

#### **Chapter 9, “Migration procedures for WebSphere InterChange Server V4.2.1 to V4.3”**

This chapter provides detailed instructions for performing the migration to WebSphere InterChange Server 4.3 from Version 4.2.1.7 using the scenario detailed in Chapter Chapter 7, “Project environment setup” on page 109. It includes guidance on tasks that must be carried out both before and after migration. The chapter has separate sections for migration on Windows and AIX.

#### **Chapter 10, “Migration procedures for WebSphere InterChange Server V4.2.2 to V4.3”**

This chapter provides detailed instructions for performing the migration to WebSphere InterChange Server 4.3 from Version 4.2.2 using the scenario detailed in Chapter Chapter 7, “Project environment setup” on page 109. It includes guidance on tasks that must be carried out both before and after migration. The chapter has separate sections for migration on Windows and AIX.

### **Basic components of the scenarios**

The scenario that is used to demonstrate the migration steps covers the following areas and features of a WebSphere InterChange Server configuration. This scenario is covered in detail in Chapter 7, “Project environment setup” on page 109, and must be referred to for further information.

- ▶ Long-lived business processes
- ▶ CustomSync collaboration including relationships
- ▶ Connection pool
- ▶ Collaboration group
- ▶ Polymorphic map
- ▶ Server Access Interface
- ▶ Transactional collaboration
- ▶ Failed flows with different transports: WebSphere MQ, Java Message Service (JMS), and Interface Definition Language (IDL)
- ▶ Schedules
- ▶ Imported external code



### 1.1.3 What is not covered

Because this book focuses only on tasks relating to migration of the WebSphere InterChange Server, it does not include information about using the WebSphere ICS or the WebSphere Integration Server product for any of the versions. For information about using these products or any new features, refer to the product documentation.

In this book, we discuss only the installation of WebSphere InterChange Server and related components; no other products are described, although tips may be given where appropriate when installing service on prerequisite products (for example, DB2 and WebSphere MQ). Use the product documentation to assist with installation and applying service.

Although the new features of WebSphere ICS are discussed in the first part of this book, only features that are required to get an existing configuration up and running after migration are included in the demonstration. For example, the new feature *Role-based access* is enabled as part of the migration, as this is required in order to have the same security configuration as before.

This redbook does not cover the migration of adapters or custom code, but general hints are provided to assist with this process. Refer to the documentation for individual adapters for specific adapter migration steps.

This redbook does not describe the steps to configure WebSphere ICS or related components or environments for High Availability or performance tuning. We include descriptions of new features that relate to performance and availability, but no details are provided for their setup and configuration, as this is beyond the scope of the book. You might wish to refer to the Redpaper™ for performance tuning, *Introduction to WebSphere InterChange Server v4.2.2*, REDP-9124, which gives some hints on performance tuning that are also applicable for Version 4.3.0. You can download this Redpaper at:

<http://www.redbooks.ibm.com/abstracts/redp9124.html>

If the existing WebSphere ICS configuration is on a platform or database version that is not supported on Version 4.3.0, an upgrade to the operating system or database may be required. We do not demonstrate this migration route step by step in this redbook. There is also no discussion of implications or actions required in migrating the operating system or database to a different version, such as migrating from Windows to AIX or from Oracle® to DB2.

Instructions for the migration of the Web-based System Monitor are provided only for Windows. For instructions for UNIX, refer to the product documentation at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

## 1.1.4 Assumptions

This redbook makes a number of assumptions in order to simplify the information here and make it useful to as many customers as possible.

The following technical assumptions are made:

- ▶ Customers are familiar with the use of WebSphere InterChange Server Version 4.3.0 and at least one previous version.
- ▶ Customers have an existing and working configuration of WebSphere InterChange Server at least to the level of one of the following releases:
  - IBM CrossWorlds 4.1.1.9
  - WebSphere InterChange Server 4.2.1.7
  - WebSphere InterChange Server 4.2.2.4
- ▶ Customers are using WebSphere MQ 5.3 at the latest supported level for the appropriate WebSphere InterChange Server version.

The migration is demonstrated using WebSphere MQ Fix Pack 5.

**Note:** The team recommends that the latest FixPack information is reviewed when planning the migrations.

## 1.2 Planning for the future

There are a number of reasons for migrating your existing WebSphere InterChange Server version to Version 4.3.0, including the enhancements and new features that are available in this version. WebSphere ICS 4.3.0 has enhancements for security, scalability, resilience, usability, and the addition of Linux® for Intel® as a supported platform. Chapter 2, “Product overview” on page 11 details the most significant of these new features. Chapter 3, “What is new?” on page 29 covers the whole product history from Version 4.1.1. so that a feature comparison can be made for all the new features and functions provided for users of versions since WebSphere InterChange Server 4.1.1.

Business Integration is being driven by the middleware market to adopt open standards and to provide common runtime and tooling based on open standards. These ultimately lead to lower total costs of ownership, greater flexibility, scalability, and skill reuse.

Many of the new features and enhancements for WebSphere InterChange Server that are described in Chapter 2, “Product overview” on page 11 and Chapter 3, “What is new?” on page 29 support this evolution. WebSphere InterChange Server 4.3 therefore represents a baseline from which migration to

future releases can be carried out. It is expected that customers migrate to WebSphere InterChange Server Version 4.3.0 in order to keep updated with the new standards and associated changes.

**Important:** It is unlikely that documentation will be made available in the future to migrate from versions of WebSphere InterChange Server prior to 4.3.0 to later evolutions of WebSphere InterChange Server.

The following recommended guidelines for development on WebSphere InterChange Server should make migration of WebSphere ICS artifacts for future versions more straightforward:

- ▶ Document the system and component design.
- ▶ Use the development tooling to edit integration artifacts.
- ▶ Leverage best practices for defining rules with the tooling and Java snippets.

When developing Java code within collaboration templates, maps, common code utilities, and other components, take these considerations into account:

- ▶ Use only the published Application Programmable Interfaces (APIs).
- ▶ Use Activity Editor.
- ▶ Use adapters to access Enterprise Information Systems (EISs).
- ▶ Avoid external dependencies in Java snippet code.
- ▶ Adhere to Java 2 Platform, Enterprise Edition (J2EE) development for portability.



## Product overview

This chapter provides an introduction to the WebSphere Business Integration Server, together with an overview of the WebSphere InterChange Server (WebSphere ICS). This chapter also details the most significant new features in WebSphere InterChange Server Version 4.3.0.

## 2.1 WebSphere Business Integration Server

WebSphere Business Integration Server is a family of products that provide modelling, transformation, integration, and management of business processes in a single solution. The different products have a range of capabilities that can be selected and used to solve specific customer business problems. WebSphere Business Integration Server is a selection of products from the WebSphere platform that focus on the key business integration requirements for enterprise environments.

This section gives an overview of the IBM Business Integration Reference Architecture as an introduction to business integration. This is followed by a brief description of the components of WebSphere Business Integration Server 4.3.0 with a guide to their function and position in the IBM Business Integration Reference Architecture.

Several other redbooks give a more in-depth guide to the IBM Business Integration Reference Architecture and WebSphere Business Integration Server. For more detailed information about the issues and history of business integration, refer to:

- ▶ *Implementing and Administering WebSphere Business Integration Server V4.2.2*, SG24-7006  
<http://www.redbooks.ibm.com/abstracts/sg247006.html>
- ▶ *WebSphere Business Integration for SAP*, SG24-6354  
<http://www.redbooks.ibm.com/abstracts/sg246354.html>
- ▶ *Using Web Services for Business Integration*, SG24-6583  
<http://www.redbooks.ibm.com/abstracts/sg246583.html>

### 2.1.1 IBM Business Integration Reference Architecture overview

The IBM Business Integration Reference Architecture shows the key areas of integration capability that are required for comprehensive solutions in an enterprise environment. It provides the capability to perform the following business integration functions:

- ▶ Model business functions and processes.
- ▶ Transform applications, processes, and data.
- ▶ Integrate islands of applications, processes, and information.
- ▶ Transparency of resources.
- ▶ Manage performance against business objectives.
- ▶ Accelerate implementation of intelligent processes.

Each of these business integration functions is described in the following sections with reference to the IBM Business Integration Reference Architecture. The key with IBM Business Integration Reference Architecture is that it is modular, meaning that business problems can be solved one at a time with the possibility of other applications, processes, or data being added to the solution in the future.

### **Model business functions and processes**

A business integration solution must have a way to graphically represent the business processes that have to be integrated. Different tools are available for modeling different processes, for example, modeling of data flows, charting and simulating business processes, configuring connections between systems and applications, and development of business logic. The Business Integration Reference Architecture provides a unified development platform as a common infrastructure for the development of these business models.

Functions important to the traditional programmer are also included for building maintainable, flexible, and reusable business logic components.

### **Transform applications, processes, and data**

Transformation in the IBM Business Integration Reference Architecture is a mechanism for reusing existing business processes and systems. This can be used to leverage existing IT infrastructure and systems to gain greater efficiency, or to reuse existing applications, data formats, and processes in integration.

### **Integrate islands of applications, processes, and information**

Integration enables the connection of applications and data that are running on disparate systems, platforms, and networks within and beyond the enterprise. Integration has a wide variety of use within business; for example, in utilizing legacy data sources with new technologies, or communication between applications and business partners using different data formats, multiple operating systems, or both. Integration can also be used to provide a common view of data, regardless of the format or storage location.

A variety of services are supplied with the IBM Business Integration Reference Architecture to enable new application components to be included in the integrated system. These application components provide new business logic required to adapt existing business processes to meet changing competitive and customer demands of the enterprise over time.

### **Transparency of resources**

The interaction side of the IBM Business Integration Reference Architecture is primarily about user interaction and the user perspective of business data and

business processes. It is invisible to the user that the presented information may come from disparate sources existing on multiple platforms and accessed through diverse channels. The information can be personalized for the user, and it creates a single user experience across applications on a variety of devices. In addition, the concept of users and roles to determine the user's function provides security to the system.

The IBM Business Integration Reference Architecture also contains a set of services that are oriented toward the integration of people, processes, and information. These services control the flow of interactions and data among people and automated application services in ways that are appropriate to the realization of a business process.

### **Manage performance to match business objectives**

There are a variety of management tools associated with the IBM Business Integration Reference Architecture to perform the following functions:

- ▶ Monitor performance of processes.
- ▶ Simulate impacts on processes.
- ▶ Derive new business processes.
- ▶ Generate reports and statistics.
- ▶ Manage system resources.
- ▶ Manage business processes.
- ▶ Manage relationships of system processes.
- ▶ Automate and manage enterprise-wide resources.
- ▶ Manage security.

Overall, these tools enable the monitoring and simulation of systems and business processes to improve performance and efficiency, and to track down problems and inefficiencies in the solutions. They can also be used for optimization and provisioning of resources.

### **Accelerate implementation of intelligent processes**

Because many business integration solutions utilize similar techniques and components using open standards, the IBM Business Integration Reference Architecture has many tools and templates to accelerate implementation. Many of these tools and templates that are available are industry-specific to reduce effort on development tasks, which ultimately reduces the costs in time and resources. The modular nature of the IBM Business Integration Reference Architecture means that projects can be developed at a small level to deal initially with critical business problems, then add other applications, data, or processes within the enterprise or belonging to suppliers or partners at a later stage. This makes the IBM Business Integration Reference Architecture flexible to dynamic market and business conditions.



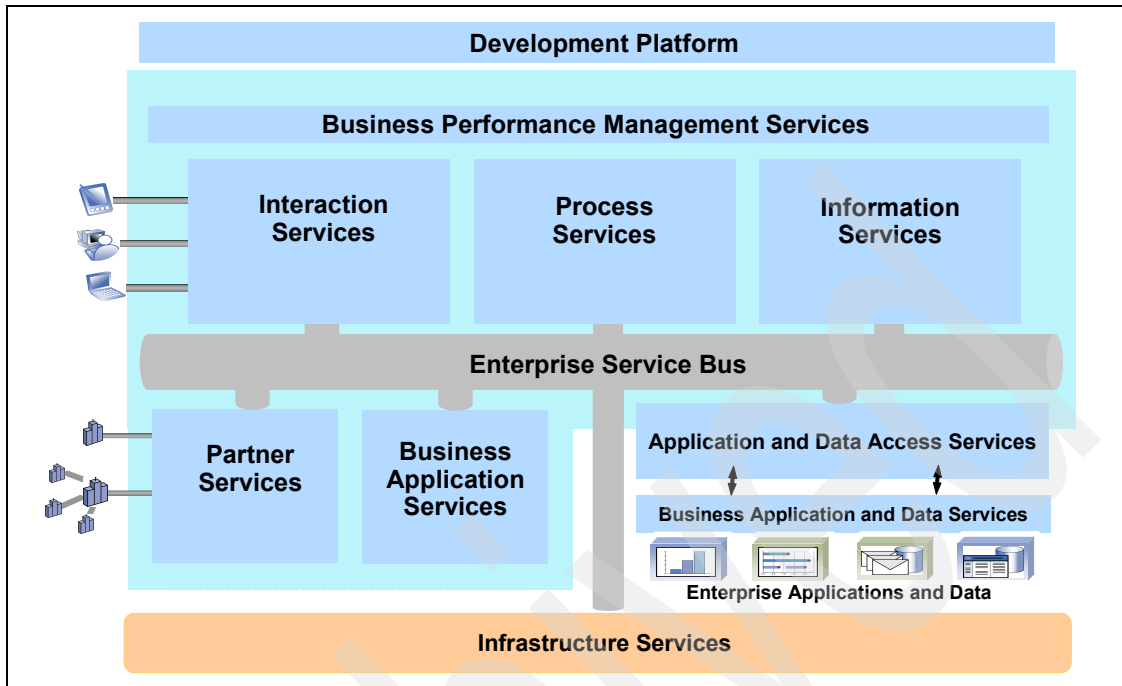


Figure 2-1 IBM Business Integration Reference Architecture

## 2.1.2 Components of WebSphere Business Integration Server 4.3.0

WebSphere Business Integration Server V4.3.0 is one offering in the WebSphere family for business integration. It is comprised of the following components:

- ▶ WebSphere ICS V4.3.0
- ▶ WebSphere MQ Workflow V3.5
- ▶ WebSphere Business Integration Message Broker V5.0.1
- ▶ WebSphere Business Integration Toolset V4.3.0
- ▶ Selected WebSphere Business Integration adapters

WebSphere MQ V5.3.0.2 and DB2 Universal Database™ Enterprise Server Edition V8.1 are included, only for restricted use with these components.

### WebSphere InterChange Server V4.3.0

WebSphere InterChange Server is an integration broker that exchanges data in the form of business objects with a set of adapters. The adapters enable communication with a variety of applications, systems, and data transport infrastructures. An overview of the product and related technologies is described in more detail in 2.2, “WebSphere InterChange Server” on page 18.

## **WebSphere MQ Workflow V3.5**

WebSphere MQ Workflow is used to support long-running processes that require the participation and intervention of human workgroups. These processes may have a duration from as little as a few seconds to as long as months. Processes that do not require human interaction can be fully automated, improving efficiency and productivity. It enables organizations to accelerate its process flow, optimize costs, eliminate errors, and improve workgroup productivity.

## **WebSphere Business Integration Message Broker V5.0.1**

WebSphere Business Integration Message Broker is a powerful integration broker driven by business rules. It has functions for publish/subscribe, transformation, enrichment, and routing of in-flight data according to rules defined in message flows. Diverse applications can exchange information in dissimilar forms with the broker handling processing to ensure that the message reaches the correct destination in the correct format as defined in the flows.

Messages can be transformed to new formats, enriched with data from other messages or external databases, and then routed to multiple destinations dependant on the business rules defined in the message flow.

Messages can be self-defining, or a template can be made in the form of a message set to predefine the information that a message may contain. These message sets can contain complex information in industry-standard formats, and then be converted easily into other message formats, with the broker performing all required processing.

## **WebSphere Business Integration Toolset V4.3.0**

The WebSphere Business Integration Toolset provides administrative and development tools for use with WebSphere ICS. The WebSphere Business Integration Toolset is supported only on Windows 2000 and Windows XP, but as the System Monitor and Failed Event Manager are Web-based, these can be used on any platform supporting an appropriate Web browser.

The administrative tools in the WebSphere Business Integration Toolset are:

- ▶ **System Manager**

The System Manager provides a single interface to monitor, control, and analyze the entire IBM WebSphere Business Integration solution. From the System Manager, you can access the designer tools for creating collaboration templates, business object definitions, maps, and relationships. The System Manager can also be used to manipulate modular components that are used in a WebSphere InterChange Server system.

- ▶ System Monitor

System Monitor provides a centralized graphical and browser-based tool to monitor and control key system components within the IBM WebSphere InterChange Server environment. From the system monitor, current and historical data can be viewed and key components can be controlled.

- ▶ Flow Manager

Flow Manager is a tool installed with WebSphere InterChange Server to graphically locate, view, and process failed events.

- ▶ Failed Event Manager

Failed Event Manager is for managing failed events from a Web browser.

- ▶ Log Viewer

The Log Viewer provides a user-friendly interface for viewing trace files from WebSphere InterChange Server and WebSphere Business Integration Adapters to assist with debugging and monitoring.

- ▶ Relationship Manager

Relationship Manager is a visual tool that monitors and manages relationships between application objects to simultaneously synchronize data across multiple applications. This tool is also used for creating relationship definitions and viewing details about the participants in the relationships.

The design tools provided in WebSphere Business Integration Toolset are:

- ▶ Map Designer

Map Designer is a graphical development tool for creating and modifying maps. Maps transform application-specific business objects into and from generic business object format.

- ▶ Process Designer

With Process Designer, users can graphically sketch and refine the logical flow of business processes and collaboration template development.

- ▶ Business Object Designer

Business Object Designer is a graphical tool for generating and maintaining business objects.

- ▶ Connector Configurator

The Connector Configurator provides an interface to view and edit adapter connector configuration properties.

## WebSphere Business Integration Adapters

IBM WebSphere Business Integration Adapters provide a large suite of ready-to-use adapters for interfacing to various Enterprise Information Systems. There are two main types of adapter: *application adapters* and *technology adapters*.

Application adapters extract data and transaction information from cross-industry and industry-specific packaged applications and connect them to a central hub for integration with other applications.

Technology adapters provide connectivity to access data, technologies, and protocols that enhance integration infrastructure.

Adapters are based on a common framework, which means they can be configured to operate in a consistent manner while being used to integrate widely different systems.

## 2.2 WebSphere InterChange Server

WebSphere InterChange Server (WebSphere ICS), a key component of WebSphere Business Integration Server, is an integration broker with a set of adapters that enable heterogeneous business applications to exchange data, which is transferred in the form of business objects. A distributed, hub-and-spoke infrastructure enables the execution of business processes across the Internet and across applications on local networks.

WebSphere ICS uses the adapters provided by the WebSphere Business Integration Adapters product, together with modular and customizable components such as collaborations, business objects, maps, and data handlers to perform these integration tasks. A description of each of these components follows.

### 2.2.1 Collaborations

WebSphere InterChange Server collaborations are software modules containing logic that describes a distributed business process. A collaboration can be simple, consisting of just a few steps, or complex, involving several steps and other collaborations, which form collaboration groups.

A typical WebSphere ICS solution includes one or more collaborations and a set of business objects that represent business information relevant to the integration. There are two types of collaboration: *collaboration templates* and *collaboration objects*. Collaboration objects are collaboration templates bound to

components in the System Manager and loaded into the WebSphere ICS environment on startup.

The toolset available with WebSphere ICS includes a collaboration design tool called Process Designer, which can be used to customize and create collaboration templates.

Additionally, IBM WebSphere Business Integration collaborations provide prebuilt and customizable business process templates that define rules for basic integration and business processes for cross-industry or industry-specific applications.

## 2.2.2 Business objects

Business objects represent the data and processing instructions between an integration broker and connected applications in a business integration system. Business objects can represent one of the following:

- ▶ A request to an adapter from an integration broker
- ▶ An event from an adapter to an integration broker
- ▶ A call from an external site using Server Access Interface

Adapters use application-specific business objects to transfer data between the adapter and the WebSphere InterChange Server. These are unique to the adapter and model the appropriate application entities.

Data that is processed within the business logic of a WebSphere InterChange Server collaboration object is in the form of a generic business object. These contain a superset of information for the application entities that need to communicate.

## 2.2.3 Maps

Maps transform data to and from generic business objects and application-specific business objects so that adapters can communicate with their applications using application-specific entities, while collaboration objects can apply business logic in an application-independent way.

Typically, application-specific business objects are mapped to generic business objects when they are received by the WebSphere InterChange Server. When a business object is sent to an adapter, the generic business object is mapped to the appropriate application-specific business object. The response is mapped back to the generic business object. Generic business objects are application-independent and can therefore be reused in the future.

## 2.2.4 WebSphere Business Integration Toolset

The WebSphere Business Integration Toolset that is provided with WebSphere InterChange Server is used to develop and customize components, organize components for deployment and production, and monitor the deployed WebSphere InterChange Server system. An overview of the major tools is given in 2.1.2, “Components of WebSphere Business Integration Server 4.3.0” on page 15.

## 2.3 New features in WebSphere InterChange Server 4.3

This section gives an overview of the major new features of WebSphere InterChange Server 4.3. Chapter 3, “What is new?” on page 29 covers other enhancements and updates.

The full list of new features, improvements, and workarounds for the individual components can be found in the release notes from the Business Integration support Web site at:

<http://www.ibm.com/software/integration/wbiserver/support/>

More detailed release notes for the individual WebSphere ICS components can also be found in the product documentation at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

The major changes to the WebSphere InterChange Server are:

- ▶ Moved to Java 1.4.2 and Eclipse 2.1
- ▶ End-to end security
- ▶ Role-based access control
- ▶ Hot deployment
- ▶ Large Object support
- ▶ JMS transport optimization
- ▶ Database connection resilience
- ▶ Fail Events Management API
- ▶ Dynamic service call
- ▶ Bidirectional language support
- ▶ Single Persistent Naming Server hosting multiple WebSphere InterChange Servers and adapters
- ▶ Web Services enhancements

- ▶ Remote service support for Integrated Test Environment
- ▶ Improved tooling facilities for Activity Editor Map, Designer, and Connector Configurator
- ▶ IBM Tivoli® License Manager support
- ▶ Linux support

Further information about the above changes is given in the following sections.

### 2.3.1 Move to Java 1.4.2 and Eclipse 2.1

The Java Runtime Environment (JRE™) and Java Development Kit (JDK™) versions for WebSphere InterChange Server, Adapter Framework, and WebSphere Business Integration Toolset have been upgraded from Version 1.3.1 to Version 1.4.2 with this release. This is provided on the product CD. Refer to the installation instructions for WebSphere InterChange Server for information about how to install the JDK, and any other prerequisite software requirements at:

<ftp://ftp.software.ibm.com/software/websphere/integration/wicserver/library/doc/wics43/pdf/>

### 2.3.2 End-to-end security

End-to-end security is a method of ensuring privacy and integrity of messages as they flow, beginning from the node of origin, through any intermediate nodes, to the destination node. This privacy and integrity is maintained during transit, even when a message is stored on a queue or a disk.

End-to-end security also allows *asymmetric security*, where the security can be different between the inbound and outbound data, and enables security parameters to be dynamic. Using end-to-end security, the security can be set at an application layer so that messages can be secured before they enter and leave the application, which prevents unauthorized sources from accessing the data while it resides in an intermediate node. This security method is most significant for JMS transportation.

Four levels of security are provided for the WebSphere InterChange Server:

#### **None**

None (the default level) is where no security level is set. Messages are sent from the WebSphere InterChange Server and the adapters as normal. None is the default security level after migration.

## Integrity

With Integrity, the sender (for example, the WebSphere InterChange Server) adds a digital signature to the message. The receiver then verifies the public key of the sender to provide authentication. The signature forms a part of the message, so the integrity of the message is guaranteed.

## Privacy

With Privacy, the sender encrypts the message with a secret key using the receiver's public key. This enables the receiver to decrypt the message, ensuring that the message can only be decrypted by the correct receiver.

## Integrity Plus Privacy

The Integrity Plus Privacy security level ensures both integrity and privacy by adding a digital signature to the message, then performs the encryption as described in the Privacy security level.

When using end-to-end security, the WebSphere InterChange Server and adapters require a key store to store public and private keys. The WebSphere InterChange Server maintains a store containing its own public and private keys, plus the public keys of all installed adapters. Each individual adapter maintains a key store with its own public and private keys, and the public keys of the WebSphere InterChange Server and any other processes to which it communicates.

The following types of messages can be secured in the WebSphere InterChange Server using end-to-end security:

- ▶ All communication between WebSphere ICS and adapters (All).
- ▶ Administrative messages (Admin).
- ▶ All business objects (BO).
- ▶ Individual business object (BO specification name).

Information about configuration of end-to-end security can be found in the *WebSphere InterChange Server System Administration Guide*:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Administration** → **PDFs** and click **System administration** in the right-side pane.

## 2.3.3 Role-based access control

Role-based access control produces user authentication to restrict access to specific components of the system depending on a predefined role. A user or process is assigned a user ID and password for accessing the system. It is not



desirable for all users to have the same access to information and components on the system, but it is time-consuming to manage each user's individual requirements. Role-based access control enables an administrator to set up a selection of roles with specific privileges and access on the WebSphere InterChange Server and related components. Users can then be assigned to one or more roles to control their access to the system.

## Authentication

User names and passwords are stored in a user registry. By default, users and passwords are stored in the local WebSphere InterChange Server repository. This user registry can be used by multiple WebSphere InterChange Servers. Alternatively, you can configure the WebSphere InterChange Server to use a separate database or to use a Lightweight Directory Access Protocol (LDAP) directory as a user registry.

## Authorization and roles

Two predefined roles are provided with WebSphere InterChange Server: administrator and guest. The Administrator role can perform all operations on the server including creating users, creating further roles, and granting permissions to roles. An administrator can perform other administrative tasks such as resetting passwords and logging out users and closing sessions. The Guest user can log on to an unlimited number of sessions. (Other users are restricted to a limited number of sessions.) If role-based access control is enabled and no user information is provided, the default action is for Guest to be used.

These operations can be secured:

- ▶ Server
  - Start, stop, deploy, export, delete, query, monitoring, shutdown
  - Administer security
  - Edit configuration files
- ▶ All components
  - Start, stop, pause, delete
- ▶ Collaboration
  - Execute, submit failed events, delete failed events, shutdown, cancel long-lived process flow, resolve transactional status
- ▶ Connector
  - Submit failed events, delete failed events

These role-based access control policies can be imported and exported as files for backups or for use with other servers:

- ▶ Permissions assigned to components
- ▶ Role definitions and role membership
- ▶ Users and passwords

## Auditing

Tracking authenticated tasks such as login and logout, component start and stop, shutdown, and so on is available from a log of audit information. The time and date of the operation, user ID, and status of the operation including success, exception, and access denied events are captured in the log.

Components are not secured by default, so if an operation of any component is not assigned to any role, that operation becomes unsecured because all roles can perform this operation. Role-based access control must be configured after installation. Refer to the *WebSphere InterChange Server System Administration Guide* for more about role-based security on WebSphere InterChange Server.

## 2.3.4 Hot deployment

This feature enables the deployment of first class components (business objects, relationships, maps, connectors, collaboration templates, and collaboration objects) to the server without requiring a server reboot. Previously this action required a reboot and resulted in system down time.

## 2.3.5 Large-object support

Prior to Version 4.3, the WebSphere InterChange Server and adapters could not process large business objects without incurring a `java.lang.OutOfMemory` error, even when sufficient memory was available in the Java heap. This *memory fragmentation* is addressed in Version 4.3.0 with the large-object support feature.

A business object is considered to be large if any of the following is true:

- ▶ The object has a large number of child objects.
- ▶ It is a wrapper object for multiple business objects used for batch processing.
- ▶ It is larger than 1 MB in its serialized form.

To enable large-object support, some configuration of WebSphere MQ (for log space and maximum message length) is required. The size of the available system memory is likely to affect this feature, and there is a fundamental limit on the maximum size of a business object that the WebSphere InterChange Server can process. Refer to the product documentation for more information about large-object support.

### 2.3.6 JMS transport optimization

Enhancements have been made in Version 4.3 of WebSphere InterChange Server for performance and scalability for JMS transport. By default optimization is switched off, but it can be switched on using the Connector Configurator. Refer to the product documentation for more about JMS Transport optimization.

### 2.3.7 Database connection resilience

Enhancements have been made to improve the connection resilience between the WebSphere InterChange Server and the databases in use. This improved resilience means that the InterChange Server is capable of tolerating temporary database unavailability, which may be caused by network failures or scheduled database maintenance.

### 2.3.8 Fail Events Management API

The following actions can be performed on failed events using the Fail Events Management API:

- ▶ Save failed events.
- ▶ Query failed events.
- ▶ Drop failed events.
- ▶ Resubmit failed events.

Role-based access control is also implemented for the Fail Events Management API.

### 2.3.9 Failed Event Manager

Failed Event Manager, an implementation of the Fail Events Management API, enables failed events to be viewed and managed from the Web. This tool is available for use with IBM WebSphere Application Server or with Tomcat. If an Administrative Toolset installation of WebSphere Business Integration Server is selected, Failed Event Manager is automatically installed and configured (if WebSphere Application Server version 5.0.2 or 5.1 is detected on the system).

### 2.3.10 Dynamic Service Calls

Dynamic Service Calls provide a mechanism for collaboration developers to make service requests to any collaboration or connector at runtime without having to bind explicitly to a destination port. This enables business objects to be dispatched to multiple destinations that are unknown when the collaboration was designed. Dynamic Service Calls can be enabled using a new API that is available to collaboration developers.

### **2.3.11 Bidirectional language support**

WebSphere Business Integration products provide bidirectional support for languages with a bidirectional script. These are languages that are written from right to left and have embedded numbers or sections of text in Western scripts that are written from left to right. Arabic and Hebrew are the two major language groups that use bidirectional scripts.

See the product documentation for more information about bidirectional scripts and bidirectional support in WebSphere InterChange Server.

### **2.3.12 One Persistent Name Server hosts multiple ICSs and adapters**

The Persistent Name Server can host multiple WebSphere InterChange Servers and the associated adapters starting with Version 4.3.0.

### **2.3.13 Web services**

In WebSphere InterChange Server 4.3.0, Web services have complete and integrated support within the toolset. Using the System Manager, Web services can be discovered, tested, and deployed. There is a wider scope of use for Web services, and these can be implemented in both maps and collaborations.

### **2.3.14 Remote service support for Integrated Test Environment**

Integrated Test Environment (ITE) tools can now debug a remote server, in addition to the local server. This support for a remote WebSphere InterChange Server removes the requirement to install a local WebSphere InterChange Server on the developer's workstation.

ITE also supports remote servers as test servers, in addition to currently supported local servers. In Version 4.3.0, any WebSphere InterChange Server becomes a test server if started with the -test option.

### **2.3.15 Improved tooling facilities for Activity Editor, Map Designer, and Connector Configurator**

Updates made in WebSphere Business Integration Toolset Version 4.3.0 include:

#### **Activity Editor**

Enhancements enable import and export of activity definitions and contextual labelling of context custom function blocks. The properties of the custom function blocks can now be edited directly by the Map Designer tool. In addition, functionality has been enhanced for Java source parsing.

## Map Designer

Usability has been improved in the Map Designer tool with the addition of auto-mapping features. Auto-mapping can be set to ignore uppercase-lowercase differences, ignore mismatched data types, and map by prefix or suffix.

With today's enterprise business applications, business objects can consist of hundreds of attributes. With such large business objects, creating maps is very time-consuming and repetitive. To improve usability, the Map Designer has been enhanced with the following auto-mapping features:

- ▶ Ignore uppercase-lowercase differences
- ▶ Ignore mismatched data types.
- ▶ Map by prefix or suffix.
- ▶ Reverse mapping.
- ▶ Synonym mapping.

Refer to the Data Mapping section of the product documentation for information about the different types of mapping and how to use them.

As mentioned in the Activity Editor section, you can now choose function blocks for custom transformation, which in previous versions was possible only through the Activity Editor.

## Connector Configurator

These new features are provided for the Connector Configurator:

- ▶ Extended validation
- ▶ Validate button
- ▶ Security configuration

Extended validation has been provided for standard and connector-specific properties based on string subtypes and the target operating system.

### 2.3.16 IBM Tivoli License Manager

The IBM WebSphere Business Integration Toolset provides enablement for use with IBM Tivoli License Manager providing support only for inventory functions. With Tivoli License Manager, customers pay for software based on utilization rather than full machine capacity by measuring the usage of products and reporting the use to IBM. You can use Tivoli License Manager for the following purposes for each site:

- ▶ Collection and extraction of inventory data
- ▶ Storage and maintenance of software procurement, contract information, license terms, and conditions

- ▶ Comparison of procured, installed, and used licenses
- ▶ Compilation of metering and inventory results to identify the license entitlement
- ▶ Supervision that software products are used in accordance with the agreements between customers and vendors

### **2.3.17 Linux support**

Version 4.3.0 of WebSphere InterChange Server adds support for Intel 32-bit architectures. The following Linux releases are supported for use with WebSphere InterChange Server:

- ▶ Red Hat® Enterprise Linux, Advanced Server 3.0 with Update 1
- ▶ SUSE® LINUX Enterprise Server (SLES) 8.1 with SP3
- ▶ SUSE LINUX Standard Server (SLSS) 8 with SP3

## What is new?

This chapter describes the new features of the WebSphere InterChange Server (WebSphere ICS) on a release-by-release basis and provides a comprehensive list of changes and updates since WebSphere InterChange Server 4.1.1. We also give an overview of the major changes for each release.

In this chapter, a full product history is given with the key features introduced in each release. In addition, information is provided about major environment and architectural changes across the releases.

## 3.1 New features by release

This section compares the major new features and enhancements of WebSphere InterChange Server with previous releases. You can view the changes between an older version and Version 4.3.0 by referring to the relevant subsection:

- ▶ For WebSphere InterChange Server Version 4.1.1, refer to 3.1.1, “Changes from WebSphere InterChange Server V4.1.1 to V4.3” on page 30.
- ▶ For WebSphere InterChange Server Version 4.2.1, refer to 3.1.2, “Changes from WebSphere InterChange Server V4.2.1 to V4.3” on page 37.
- ▶ For WebSphere InterChange Server Version 4.2.2, refer to 3.1.3, “Changes from WebSphere InterChange Server V4.2.2 to V4.3” on page 41.

Chapter 2, “Product overview” on page 11 discusses the major features of WebSphere InterChange Server release 4.3.0 and should be referred to for further information.

### 3.1.1 Changes from WebSphere InterChange Server V4.1.1 to V4.3

This section details the key feature releases since 4.1.1 that are important for the migration of WebSphere InterChange Server Version 4.1.1 to Version 4.3.0. Not all of the major new features in WebSphere InterChange Server 4.3.0 (and other releases since 4.1.1) affect the migration of an existing 4.1.1 system, so these are not discussed in detail here.

More specific information about changes to platforms and software requirements can be found in 3.3, “System architecture changes” on page 49. Tables are also provided showing all new or changed features since 4.1.1, including deprecated components.

#### **Replacement of VisiBroker Object Request Broker**

Beginning with the 4.2.2 release, the WebSphere InterChange Server system no longer used the VisiBroker Object Request Broker (ORB) to handle communication between WebSphere InterChange Server and its clients (such as adapters, WebSphere Business Integration tools, SNMP agents, and access clients). The InterChange Server system now uses the IBM ORB.

The WebSphere InterChange Server 4.3.0 installer automatically installs the IBM ORB as part of the Java Runtime Environment (JRE). WebSphere InterChange Server now uses the IBM Transient Naming Server instead of the VisiBroker Smart Agent to provide its naming service.

Before WebSphere InterChange Server 4.2.2, automatic restart of adapter agents was done using a facility within the VisiBroker component known as the



Object Activation Daemon (OAD). VisiBroker is no longer used to implement OAD, as the IBM ORB does not provide an equivalent OAD functionality.

An alternative approach is provided by WebSphere MQ triggering. Each adapter instance has a named queue used for triggering of its adapter agent. This approach can be used wherever WebSphere MQ is available. In particular, it is available across the firewalls and the Internet, which have historically presented issues for the OAD mechanism.

Additionally a number of other updates are needed to make a WebSphere InterChange Server 4.1.1 scenario function correctly with the IBM ORB supplied with WebSphere InterChange Server 4.3.0, such as upgrading scripts and applications.

Technical information about the change from VisiBroker to the IBM ORB can be found in 5.1, “VisiBroker to IBM ORB change” on page 69, including steps for setting up WebSphere MQ triggering.

### **Java Development Kit upgrade**

WebSphere InterChange Server system now uses the Java Runtime Environment (JRE) and the Java Development Kit (JDK) at level 1.4.2.

It is important to check custom Java code for maps, collaboration templates, and third-party libraries for compatibility and deprecated methods for the new version. Errors are generated for incompatibilities and deprecated methods at compile time.

Additionally, the following components utilized by WebSphere InterChange Server are embedded in JDK 1.4.2:

- ▶ XSLT4J Java 2.6.0
- ▶ XmlCommonExternal 1.2.02
- ▶ XML4J 4.3.1
- ▶ ORB 1.4.2
- ▶ JCE 1.4.2

### **WebSphere MQ upgrade**

Since WebSphere InterChange Server Version 4.1.1, the supported version of WebSphere MQ has been updated to WebSphere MQ Version 5.3. This means that WebSphere MQ Version 5.2 is no longer supported and must be upgraded before WebSphere ICS is installed. Check the WebSphere InterChange Server product documentation for the latest supported fix pack for WebSphere MQ.

## Changes in database support

Since WebSphere InterChange Server Version 4.1.1, the supported versions of databases have been updated. This means that either the database in use must be upgraded, or the existing data must be moved to a supported database. Check the WebSphere InterChange Server product documentation for the latest supported fix pack for the database.

Specific information about supported databases for each platform can be found in the WebSphere InterChange Server installation documentation.

## Integration component library

Since Version 4.2.0, development of WebSphere ICS components is performed locally rather than in the WebSphere InterChange Server instance. An integration component library is now used to store developed components. When upgrading from Version 4.1.1, it is necessary to create an integration component library within System Manager on the Windows machine running the WebSphere Business Integration Toolset. Development tools load and save object definitions from integration component libraries locally in System Manager and not on the WebSphere InterChange Server.

## Map and collaboration formats

For versions of WebSphere InterChange Server since 4.2.0, all collaboration and map information is stored as part of the collaboration template and map definition in the repository. Therefore, map and collaboration template formats must be migrated to the new format to be compatible with WebSphere InterChange Server 4.3.0. This process is explained in the migration chapters.

## Connectors versus adapters

The term *connectors* was replaced with *WebSphere Business Integration Adapters* beginning with WebSphere InterChange Server 4.20. There is no longer a single installer for all adapters, so each adapter must be installed separately with its own installer. WebSphere Business Integration Adapters is considered to be a separate product from WebSphere InterChange Server.

The WebSphere Business Integration Adapter Framework V2.6 is required to support WebSphere InterChange Server connectivity, as well as new features such as role-based access. The WebSphere Business Integration Adapter Framework must be installed in order to use adapters with WebSphere ICS.

The Connector Development Kit is no longer included. It is replaced by the WebSphere Business Integration Adapter Development Kit, an integrated development toolkit that provides the framework for developing custom adapters. The development kit provides programming interfaces in both Java and C++. The

WebSphere Business Integration Adapters Development Kit is not provided with the WebSphere InterChange Server and must be installed separately.

## Security features

Most of the new security features for WebSphere InterChange Server are implemented in Version 4.3.0. Refer to 2.3, “New features in WebSphere InterChange Server 4.3” on page 20 for detailed information about these new features. The main consideration for migration is ensuring that the security that exists in the 4.1.1 scenario is still in place after migration to 4.3.0. The individual migration chapters detail the steps required to set up role-based access security for these scenarios.

## Other features

Table 3-1 shows the complete list of new features, enhancements, and changes that have been made in releases 4.2 through 4.3. Refer to the WebSphere ICS product documentation for details of any of the features not covered in Chapter 2, “Product overview” on page 11 or this chapter.

*Table 3-1 New features, enhancements, and changes since 4.1.1*

Feature	Release introduced/deprecated
Additional National Language Support	Introduced in 4.2.0
Configuration files converted to XML	Introduced in 4.2.0
Long-lived business processes supported	Introduced in 4.2.0
Hierarchical property support	Introduced in 4.2.0
Guaranteed Event Delivery	Introduced in 4.2.0
Installation of upgrades	Introduced in 4.2.0
Persistent Monitoring	Introduced in 4.2.0
Remote Agent	Introduced in 4.2.0
Serverless Trading Agent	Deprecated in 4.2.0
Flow Control	Introduced in 4.2.0
Code generation	Introduced in 4.2.0
Recovery Optimization	Introduced in 4.2.0
High Availability	Introduced in 4.2.0
Database support for 8.1	Introduced in 4.2.0
Database support for 7.2	Deprecated in 4.2.0

Feature	Release introduced/deprecated
Messaging Support - WebSphere MQ 5.3.	Introduced in 4.2.0
Messaging Support - WebSphere MQ 5.2	Deprecated in 4.2.0
Access Options	Introduced in 4.2.0
Meta-data Management Subsystem (MMS)	Introduced in 4.2.0
IBM CrossWorlds Full Toolkit	Deprecated in 4.2.0
IBM WebSphere Business Integration Toolset	Introduced in 4.2.0
System Manager implemented in Eclipse Framework	Introduced in 4.2.0
Web-based System Monitoring	Introduced in 4.2.0
Integrated Test Environment	Introduced in 4.2.0
Process Designer enhancements for WSDL, long-lived transactions, Business Object Probes, table-driven editors, property editors, decision nodes	Introduced in 4.2.0
Map Designer enhancements for Cross-Reference Transformation, validation options, removal of verb view, map-specified messages	Introduced in 4.2.0
Connector Configurator enhancements for connector-specific template wizards and Connector Properties	Introduced in 4.2.0
Business objects use XRmi protocol to communicate with ODAs - direct connection to WebSphere InterChange Server deprecated	Introduced in 4.2.0
Repos_copy outputs repository contents in .jar containing XML format objects	Introduced in 4.2.0
SNMP tools internationalized	Introduced in 4.2.0
Visual Test Connector redesigned for Eclipse	Introduced in 4.2.0
Flow Manager developed as stand-alone application; replaces unresolved flows in System Manager	Introduced in 4.2.0
Connection Development Kit	Deprecated in 4.2.0
WebSphere Business Integration Adapters Development Kit	Introduced in 4.2.0

Feature	Release introduced/deprecated
Connectors	Deprecated in 4.2.0
WebSphere Business Integration Adapters replace connectors	Introduced in 4.2.0
Deadlock Retry feature for MS-SQL Server	Introduced in 4.2.1
Server Component Management view	Introduced in 4.2.1
VisiBroker Object Request Broker	Deprecated in 4.2.2
IBM Object Request Broker	Introduced in 4.2.2
Flow Monitoring	Introduced in 4.2.2
Collaboration Debugger	Introduced in 4.2.2
Configurable stack trace	Introduced in 4.2.2
High-availability scripts moved to Support Pack	Introduced in 4.2.2
Embedded JRE for Windows	Deprecated in 4.2.2
IBM JRE	Introduced in 4.2.2
Support for Windows XP	Introduced in 4.2.2
Support for Windows NT®	Deprecated in 4.2.2
Business Object Probes hierarchical support	Introduced in 4.2.2
System View for System Manager	Introduced in 4.2.2
Test Connector Viewer renamed Client Simulator Viewer	Introduced in 4.2.2
Dependency Viewer	Deprecated in 4.2.2
Outline Viewer	Introduced in 4.2.2
Hot Deployment	Introduced in 4.3
Role-based access control	Introduced in 4.3
End-to-end security	Introduced in 4.3
JMS Optimization	Introduced in 4.3
Large-object support	Introduced in 4.3
Configurable event sequencing	Introduced in 4.3
Failed-flow administration API	Introduced in 4.3

Feature	Release introduced/deprecated
Database connection resilience	Introduced in 4.3
IBM Tivoli License Manager	Introduced in 4.3
Remote debug on Intergration Test Environment	Introduced in 4.3
Compile and deploy progress report	Introduced in 4.3
Deployment of Java classes when maps and collaborations are deployed	Introduced in 4.3
Dynamic service call	Introduced in 4.3
Bidirectional support	Introduced in 4.3
Support for DB2 Information Integrator	Introduced in 4.3
Failed Event Manager	Introduced in 4.3
Execute Web Services natively	Introduced in 4.3
System Manager editors for roles and security policy	Introduced in 4.3
Import and export of user registries	Introduced in 4.3
Import and export of roles and security policies	Introduced in 4.3
Security Administration enhancements	Introduced in 4.3
Support for deploying, importing, and exporting class files	Introduced in 4.3
WebSphere Business Integration Adapter Framework required for Client Simulator	Introduced in 4.3
RMI communication between ITE and WebSphere ICS	Deprecated in 4.3
CORBA communication between ITE and WebSphere ICS	Introduced in 4.3
Configure retry times for Client Simulator	Introduced in 4.3
Automatic Mapping	Introduced in 4.3
Import and export of activity setting configurations	Introduced in 4.3
Addition of security to SNMP Configuration Manager	Introduced in 4.3

### 3.1.2 Changes from WebSphere InterChange Server V4.2.1 to V4.3

This section discusses the key feature changes since release 4.2.1 that are important for the migration of WebSphere ICS Version 4.2.1 to Version 4.3.0. Not all of the major new features in WebSphere InterChange Server 4.3.0 and other releases since 4.2.1 have an impact on the migration of an existing 4.2.1 system so these are not discussed in detail here. Refer to Chapter 2, “Product overview” on page 11 for more information about major features of release 4.3.0.

More specific information about changes to platforms and software requirements can be found in 3.3, “System architecture changes” on page 49. That section provides tables that shows all new or changed features since 4.1.1, including deprecated components.

#### **Replacement of VisiBroker ORB**

Beginning with the 4.2.2 release, the WebSphere InterChange Server system no longer uses the VisiBroker Object Request Broker (ORB) to handle communication between WebSphere InterChange Server and its clients (such as adapters, WebSphere Business Integration tools, SNMP agents, and access clients). Instead, the InterChange Server system uses the IBM ORB.

The WebSphere InterChange Server 4.3.0 installer automatically installs the IBM ORB as part of the Java Runtime Environment (JRE). WebSphere InterChange Server now uses the IBM Transient Naming Server instead of the VisiBroker Smart Agent to provide its naming service.

Before WebSphere InterChange Server 4.2.2, automatic restart of adapter agents was done using a facility within the VisiBroker component known as the Object Activation Daemon (OAD). VisiBroker is no longer used to implement OAD, as the IBM ORB does not provide an equivalent OAD functionality.

An alternative approach is provided by WebSphere MQ triggering. Each adapter instance has a named queue used for triggering of its adapter agent. This approach can be used wherever WebSphere MQ is available. In particular, it is available across the firewalls and the Internet, which have historically presented issues for the OAD mechanism.

Additionally a number of other updates are needed to make a WebSphere InterChange Server 4.2.1 scenario function correctly with the IBM ORB supplied with WebSphere InterChange Server 4.3.0, such as upgrading scripts and applications.

Technical information about the change from VisiBroker to the IBM ORB can be found in 5.1, “VisiBroker to IBM ORB change” on page 69, including steps for setting up WebSphere MQ triggering.

## Java Development Kit upgrade

WebSphere InterChange Server system now uses the Java Runtime Environment (JRE) and the Java Development Kit (JDK) at level 1.4.2.

It is important to check custom Java code for maps, collaboration templates, and third-party libraries for compatibility and deprecated methods for the new version. Errors are generated for incompatibilities and deprecated methods at compile time.

Additionally, the following components utilized by WebSphere InterChange Server are embedded in JDK 1.4.2:

- ▶ XSLT4J Java 2.6.0
- ▶ XmlCommonExternal 1.2.02
- ▶ XML4J 4.3.1
- ▶ ORB 1.4.2
- ▶ JCE 1.4.2

## Changes in database support

Refer to 3.3.2, “Supported databases” on page 50 for the databases that are supported for WebSphere InterChange Server 4.3.0. If the supported versions of databases have been updated, then either the database in use must be upgraded, or the existing data must be moved to a supported database. Check the WebSphere InterChange Server product documentation for the latest supported fix pack for the database.

## Adapters

The WebSphere Business Integration Adapter Framework V2.6 is required to support WebSphere InterChange Server connectivity and new features such as role-based access. The WebSphere Business Integration Adapter Framework must be installed in order to use adapters with WebSphere InterChange Server.

There is no longer a single installer for all adapters, so each adapter must be installed separately with its own installer.

## Security features

Most of the new security features for WebSphere InterChange Server are implemented in Version 4.3.0. Refer to 2.3, “New features in WebSphere InterChange Server 4.3” on page 20 for detailed information about these new features. The main consideration for migration is ensuring that the security that exists in the 4.1.1 scenario is still in place after migration to 4.3.0. The individual migration chapters detail the steps required to set up role-based access security for these scenarios.



## Other features

Table 3-2 shows the complete list of new features, enhancements, and changes that have been made in releases 4.2 through 4.3. Refer to the WebSphere ICS product documentation for details of any of the features not covered in Chapter 2, “Product overview” on page 11 or this chapter.

*Table 3-2 New features, enhancements, and changes since release 4.2.1*

Feature	Release introduced/deprecated
VisiBroker Object Request Broker	Deprecated in 4.2.2
IBM Object Request Broker	Introduced in 4.2.2
Flow Monitoring	Introduced in 4.2.2
Collaboration Debugger	Introduced in 4.2.2
Configurable stack trace	Introduced in 4.2.2
High-availability scripts moved to Support Pack	Introduced in 4.2.2
Embedded JRE for Windows	Deprecated in 4.2.2
IBM JRE	Introduced in 4.2.2
Support for Windows XP	Introduced in 4.2.2
Support for Windows NT	Deprecated in 4.2.2
Business Object Probes hierarchical support	Introduced in 4.2.2
System View for System Manager	Introduced in 4.2.2
Test Connector Viewer renamed Client Simulator Viewer	Introduced in 4.2.2
Dependency Viewer	Deprecated in 4.2.2
Outline Viewer	Introduced in 4.2.2
Hot Deployment	Introduced in 4.3
Role-based access control	Introduced in 4.3
End-to-end security	Introduced in 4.3
JMS Optimization	Introduced in 4.3
Large-object support	Introduced in 4.3
Configurable event sequencing	Introduced in 4.3
Failed-flow administration API	Introduced in 4.3

Feature	Release introduced/deprecated
Database connection resilience	Introduced in 4.3
IBM Tivoli License Manager	Introduced in 4.3
Remote debug on Intergration Test Environment	Introduced in 4.3
Compile and deploy progress report	Introduced in 4.3
Deployment of Java classes when maps and collaborations are deployed	Introduced in 4.3
Dynamic service call	Introduced in 4.3
Bidirectional support	Introduced in 4.3
Support for DB2 Information Integrator	Introduced in 4.3
Failed Event Manager	Introduced in 4.3
Execute Web Services natively	Introduced in 4.3
System Manager editors for roles and security policy	Introduced in 4.3
Import and export of user registries	Introduced in 4.3
Import and export of roles and security policies	Introduced in 4.3
Security Administration enhancements	Introduced in 4.3
Support for deploying, importing, and exporting class files	Introduced in 4.3
WebSphere Business Integration Adapter Framework required for Client Simulator	Introduced in 4.3
RMI communication between ITE and InterChange Server	Deprecated in 4.3
CORBA communication between ITE and InterChange Server	Introduced in 4.3
Configure retry times for Client Simulator	Introduced in 4.3
Automatic Mapping	Introduced in 4.3
Import and export of activity setting configurations	Introduced in 4.3
Addition of security to SNMP Configuration Manager	Introduced in 4.3

### 3.1.3 Changes from WebSphere InterChange Server V4.2.2 to V4.3

This section provides information about the key feature releases in Version 4.3.0 that are important for the migration of WebSphere InterChange Server Version 4.2.2 to V4.3.0. Not all of the major new features in V4.3.0 affect the migration of an existing 4.2.2 system, so these are not discussed in detail here.

More specific information about changes to platforms and software requirements can be found in 3.3, “System architecture changes” on page 49. Tables are also provided showing all new or changed features since 4.2.2, including deprecated components.

#### Java Development Kit upgrade

WebSphere InterChange Server system now uses the Java Runtime Environment (JRE) and the Java Development Kit (JDK) at level 1.4.2.

It is important to check custom Java code for maps, collaboration templates, and third-party libraries for compatibility and deprecated methods for the new version. Errors are generated for incompatibilities and deprecated methods at compile time.

Additionally, the following components utilized by WebSphere InterChange Server are embedded in JDK 1.4.2:

- ▶ XSLT4J Java 2.6.0
- ▶ XmlCommonExternal 1.2.02
- ▶ XML4J 4.3.1
- ▶ ORB 1.4.2
- ▶ JCE 1.4.2

#### Changes in database support

Refer to 3.3.2, “Supported databases” on page 50 for the databases that are supported for WebSphere InterChange Server 4.3.0. If the supported versions of databases have been updated, then either the database in use must be upgraded, or the existing data must be moved to a supported database. Check the WebSphere InterChange Server product documentation for the latest supported fix pack for the database.

#### Adapters

The WebSphere Business Integration Adapter Framework V2.6 is required to support WebSphere InterChange Server connectivity and new features such as role-based access. The WebSphere Business Integration Adapter Framework must be installed in order to use adapters with WebSphere InterChange Server.

There is no longer a single installer for all adapters, so each adapter must be installed separately with its own installer.

## Security features

Most of the new security features for WebSphere InterChange Server are implemented in Version 4.3.0. Refer to 2.3, “New features in WebSphere InterChange Server 4.3” on page 20 for detailed information about these new features. The new security features do not require any direct tasks to migrate an existing scenario on a previous release. The main consideration for migration is ensuring that the security that exists in the 4.2.2 scenario is still in place after migration to 4.3.0.

## Other features

Table 3-3 shows the complete list of new features, enhancements, and changes that have been made in the releases of WebSphere InterChange Server between 4.2.2 and 4.3.0. Refer to WebSphere InterChange Server product documentation for details of any of the features not covered in Chapter 2, “Product overview” on page 11 or this chapter.

*Table 3-3 New features, enhancements, and changes since release 4.2.2*

Feature	Release introduced
Hot Deployment	Introduced in 4.3
Role-based access control	Introduced in 4.3
End-to-end security	Introduced in 4.3
JMS Optimization	Introduced in 4.3
Large-object support	Introduced in 4.3
Configurable event sequencing	Introduced in 4.3
Failed-flow administration API	Introduced in 4.3
Database connection resilience	Introduced in 4.3
IBM Tivoli License Manager	Introduced in 4.3
Remote debug on Intergration Test Environment	Introduced in 4.3
Compile and deploy progress report	Introduced in 4.3
Deployment of Java classes when maps and collaborations are deployed	Introduced in 4.3
Dynamic service call	Introduced in 4.3

Feature	Release introduced
Bidirectional support	Introduced in 4.3
Support for DB2 Information Integrator	Introduced in 4.3
Failed Event Manager	Introduced in 4.3
Execute Web Services natively	Introduced in 4.3
System Manager editors for roles and security policy	Introduced in 4.3
Import and export of user registries	Introduced in 4.3
Import and export of roles and security policies	Introduced in 4.3
Security Administration enhancements	Introduced in 4.3
Support for deploying, importing, and exporting class files	Introduced in 4.3
WebSphere Business Integration Adapter Framework required for Client Simulator	Introduced in 4.3
RMI communication between ITE and WebSphere ICS	Deprecated in 4.3
CORBA communication between ITE and WebSphere ICS	Introduced in 4.3
Configure retry times for Client Simulator	Introduced in 4.3
Automatic Mapping	Introduced in 4.3
Import and export of activity-setting configurations	Introduced in 4.3
Addition of security to SNMP Configuration Manager	Introduced in 4.3

## 3.2 Product history

The tables in this section show the full product history of new features, changes, and enhancements since WebSphere InterChange Server 4.1.1. These features are organized into functional areas (security, usability, performance, and scalability). See 3.3, “System architecture changes” on page 49 for information about features that relate to changes to platforms and software requirements.

### 3.2.1 Security features

Table 3-4 on page 44 shows a comparison of the security features across the releases of WebSphere InterChange Server from 4.1.1 to 4.3.0.

Table 3-4 Security feature comparison across releases

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Role-based access control	✗	✗	✗	✓
End-to-end security	✗	✗	✗	✓
System Manager - editors for roles and security policy	✗	✗	✗	✓
Import and export of user registries	✗	✗	✗	✓
Import and export of roles and security policies	✗	✗	✗	✓
Security Administration enhancements	✗	✗	✗	✓
Addition of security to SNMP Configuration Manager	✗	✗	✗	✓

### 3.2.2 Availability features

Table 3-5 gives a comparison of the features relating to availability across the releases of WebSphere InterChange Server from 4.1.1 to 4.3.0.

Table 3-5 Availability feature comparison across releases

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Long-lived business processes support	✗	✓	✓	✓
Guaranteed Event Delivery	✗	✓	✓	✓
Remote Agent	✗	✓	✓	✓
Serverless Trading Agent	✓	✗	✗	✗
Deadlock Retry feature for MS-SQL Server	✗	✓	✓	✓
High Availability scripts moved to Support Pack	✗	✗	✓	✓
Hot Deployment	✗	✗	✗	✓
Database connection resilience	✗	✗	✗	✓
Configure retry times for Client Simulator	✗	✗	✗	✓

### 3.2.3 Usability, administrative, and serviceability features

Table 3-6 shows a comparison of the administrative features, as well as those related to usability and serviceability, across the releases of WebSphere InterChange Server from 4.1.1 to 4.3.0.

Table 3-6 Usability/administrative/serviceability feature comparison across releases

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Configuration files converted to XML	✗	✓	✓	✓
Hierarchical property support	✗	✓	✓	✓
Persistent monitoring	✗	✓	✓	✓
Code generation	✗	✓	✓	✓
Meta-data Management System	✗	✓	✓	✓
IBM CrossWorlds Full Toolkit	✓	✗	✗	✗
IBM WebSphere Business Integration Toolset	✗	✓	✓	✓
System Manager implemented in Eclipse	✗	✓	✓	✓
Web-based System Monitor	✗	✓	✓	✓
Integrated Test Environment	✗	✓	✓	✓
Process Designer enhancements for WSDL, long-lived transactions, Business Object Probes, table-driven editors, property editors, decision nodes	✗	✓	✓	✓
Map Designer enhancements for Cross-Reference Transformation, validation options, removal of verb view, map-specified messages	✗	✓	✓	✓
Connector Configurator enhancements for connector-specific template wizards and Connector Properties	✗	✓	✓	✓
Visual Test Connector redesigned for Eclipse	✗	✓	✓	✓
Flow Manager developed as stand-alone application, replaces unresolved flows in System Manager	✗	✓	✓	✓
Server Component Management View	✗	✓	✓	✓

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Flow monitoring	✗	✗	✓	✓
Collaboration Debugger	✗	✗	✓	✓
Configurable stack trace	✗	✗	✓	✓
Business Object Probes hierarchical support	✗	✗	✓	✓
System View for System Manager	✗	✗	✓	✓
Dependency Viewer	✗	✗	✓	✓
Outline Viewer	✗	✗	✓	✓
Configurable event sequencing	✗	✗	✗	✓
Failed flow administration	✗	✗	✗	✓
Remote debug on Integration Test Environment	✗	✗	✗	✓
Failed Event Manager	✗	✗	✗	✓
Automatic mapping	✗	✗	✗	✓
Import and export of activity setting configurations	✗	✗	✗	✓

### 3.2.4 Scalability and performance features

Table 3-7 shows a comparison of features related to scalability and performance across the releases of WebSphere InterChange Server from 4.1.1 to 4.3.0.

*Table 3-7 Scalability and performance feature comparison across releases*

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Flow control	✗	✓	✓	✓
Recovery optimization	✗	✓	✓	✓
High availability	✗	✓	✓	✓
JMS optimization	✗	✗	✗	✓
Large-object support	✗	✗	✗	✓



### 3.2.5 National Language Support features

Table 3-8 gives a comparison of National Language Support and related features across the releases of WebSphere InterChange Server from 4.1.1 to 4.3.0.

*Table 3-8 National Language Support feature comparison across releases*

Feature	4.1.1	4.2.1	4.2.2	4.3.0
English and Japanese language support	✓	✓	✓	✓
French, Italian, German, Spanish, Brazilian Portuguese, Korean, Simplified Chinese, Traditional Chinese	✗	✓	✓	✓
SNMP tools internationalized	✗	✓	✓	✓
Bidirectional support	✗	✗	✗	✓

### 3.2.6 Rebranding and renaming features

Table 3-9 shows a table of features associated with rebranding across the releases of WebSphere InterChange Server from 4.1.1 to 4.3.0.

*Table 3-9 Rebranding and renaming feature comparison across releases*

Feature	4.1.1	4.2.1	4.2.2	4.3.0
IBM CrossWorlds Access Framework	✓	✗	✗	✗
IBM CrossWorlds Access Framework for Enterprise JavaBeans™	✓	✗	✗	✗
WebSphere Business Integration Adapter for JCA Connection to WebSphere ICS	✓	✗	✗	✗
IBM WebSphere InterChange Server Access	✗	✓	✓	✓
IBM WebSphere InterChange Server Access for Enterprise JavaBeans	✗	✓	✓	✓
IBM WebSphere InterChange Server Access for J2EE Connector Architecture	✗	✓	✓	✓
Test Connector Viewer renamed Client Simulator	✗	✗	✓	✓

### 3.2.7 Development features

Table 3-10 shows a comparison of the development features not covered in the above sections.

Table 3-10 Development feature comparison across releases

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Connector Development Kit	✓	✗	✗	✗
WebSphere Business Integration Adapters Development Kit	✗	✓	✓	✓
Compile and deploy progress report	✗	✗	✗	✓
Deployment of Java classes when maps and collaborations are deployed	✗	✗	✗	✓
Dynamic Service Call	✗	✗	✗	✓
Execute Web Services natively	✗	✗	✗	✓
Support for deploying, importing, and exporting class files	✗	✗	✗	✓

### 3.2.8 Miscellaneous features

This section gives a comparison across releases of features that were not covered in the previous tables.

Table 3-11 Miscellaneous feature comparison across releases

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Installation of upgrades	✗	✓	✓	✓
Business objects use XRmi protocol to communicate with ODAs; direct connection to WebSphere ICS deprecated	✗	✓	✓	✓
Repos_copy outputs repository contents in .jar file containing XML format objects	✗	✓	✓	✓
Connectors	✓	✗	✗	✗
WebSphere Business Integration Adapters replace connectors	✗	✓	✓	✓
VisiBroker Object Request Broker	✓	✓	✗	✗
IBM Object Request Broker	✗	✗	✓	✓

Feature	4.1.1	4.2.1	4.2.2	4.3.0
IBM Tivoli License Manager Support	✗	✗	✗	✓
Support for DB2 Information Manager	✗	✗	✗	✓
WebSphere Business Integration Framework required for Client Simulator	✗	✗	✗	✓
RMI communication between ITE and WebSphere ICS	✓	✓	✓	✗
CORBA communication between ITE and WebSphere ICS	✗	✗	✗	✓

## 3.3 System architecture changes

The following tables show the main architecture and software dependency changes among the different versions of WebSphere InterChange Server. With these tables, you can determine which platforms and software are deprecated and require updating during migration to WebSphere InterChange Server 4.3.0.

### 3.3.1 Supported operating systems

Table 3-12 shows a comparison across the different releases of WebSphere InterChange Server of supported operating systems and those that are no longer supported. In order to migrate to Version 4.3.0, WebSphere InterChange Server must be installed or migrated onto one of the supported platforms.

Table 3-12 Supported operating systems

Feature	4.1.1	4.2.1	4.2.2	4.3.0
IBM AIX 4.3.3	✓	✗	✗	✗
IBM AIX 5.1	✗	✓	✓	✓
IBM AIX 5.2	✗	✗	✓	✓
HPUX 11i	✗	✓	✓	✓
Red Hat Enterprise Linux, Advanced Server 3.0	✗	✗	✗	✓
SUSE LINUX Enterprise Server (SLSS) 8	✗	✗	✗	✓
SUSE LINUX Standard Server (SLES) 8.1	✗	✗	✗	✓
Sun™ Solaris™ 7	✓	✓	✓	✗

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Sun Solaris 8	✗	✓	✓	✓
Sun Solaris 9	✗	✗	✗	✓
Windows NT (Workstation or Server)	✓	✓	✗	✗
Windows 2000 (Professional, Server, or Advanced Server)	✓	✓	✓	✓
Windows 2003 (Standard Edition or Enterprise Edition)	✗	✗	✗	✓
Windows XP Professional (supported for tooling only)	✗	✗	✓	✓

### 3.3.2 Supported databases

Table 3-13 shows a comparison across the different releases of WebSphere InterChange Server of the supported databases and those that are no longer supported. In order to migrate to Version 4.3.0, WebSphere InterChange Server must be installed or migrated onto one of the supported databases.

*Table 3-13 Database support across releases of WebSphere InterChange Server*

Feature	4.1.1	4.2.1	4.2.2	4.3.0
DB2 Universal Database 7.2	✓	✗	✗	✗
DB2 Universal Database 8.1	✗	✓	✓	✓
Oracle Database Server 8.1.6	✓	✗	✗	✗
Oracle Database Server 8.1.7	✓	✓	✓	✓
Oracle Database Server 9.2 (9i)	✗	✗	✓	✓
Microsoft® SQL Server® 7	✓	✓	✓	✗
Microsoft SQL Server 2000	✓	✓	✓	✓

### 3.3.3 Other supported software

Table 3-14 on page 51 shows a comparison across WebSphere InterChange Server releases of the supported levels of prerequisite software, including third-party software requirements. In order to migrate to Version 4.3.0, these software components must be installed or upgraded to the supported levels.

Table 3-14 Supported software comparison across releases

Feature	4.1.1	4.2.1	4.2.2	4.3.0
Borland VisiBroker Object Request Broker	✓	✓	✗	✗
DataDirect JDBC™ driver 3.1	✓	✗	✗	✗
DataDirect JDBC driver 3.2	✗	✓	✓	✗
DataDirect JDBC driver 3.3	✗	✗	✗	✓
IBM Object Request Broker	✗	✗	✓	✓
(AIX) IBM JDK 1.3.1	✓	✓	✓	✗
(AIX) IBM JDK 1.4.2	✗	✗	✗	✓
(HP) HP JDK 1.3.1	✓	✓	✓	✗
(HP) HP JDK 1.4.2	✗	✗	✗	✓
(Sun) Sun JDK 1.3.1	✓	✓	✓	✗
(Sun) Sun JDK 1.4.2	✗	✗	✗	✓
(Windows) Sun JDK 1.3.1	✓	✓	✗	✗
(Windows) IBM JDK 1.3.1	✗	✗	✓	✗
(Windows) IBM JDK 1.4.2	✗	✗	✗	✓
Xerces (XML Parser) 1.4.3	✓	✓	✓	✗
XML4J (XML Parser) 4.2.3	✗	✗	✗	✓
WebSphere Application Server 5.0.24	✗	✓	✓	✓
WebSphere Application Server 5.1	✗	✗	✓	✓
Tomcat 4.1.24	✗	✗	✓	✓
Tomcat 4.1.27	✗	✗	✓	✓

Chapter 4, “General planning” on page 53 gives more detailed information about how the new features, changes, and enhancements discussed in this chapter affect planning for migration. It also provides information about the steps and considerations needed to upgrade the software in the tables in this section.

For technical details of the impact of the change from the VisiBroker ORB to the IBM Object Request Broker, see 5.1, “VisiBroker to IBM ORB change” on page 69. It also covers changes from IBM CrossWorlds V4.1.1 tooling to the updated tooling in subsequent versions of WebSphere InterChange Server.



## General planning

This chapter describes the types of migration methods available and important considerations when developing a WebSphere InterChange Server (WebSphere ICS) migration project plan:

- ▶ General concepts
- ▶ Hardware
- ▶ Software
- ▶ Human Resource roles
- ▶ Time considerations
- ▶ Stage checklist

## 4.1 Migration methods

This book demonstrates two supported migration methods. After examining the WebSphere InterChange Server environment and the information provided in this chapter, choose a suitable migration path as the first step in a developing a migration plan.

### 4.1.1 In-place database migration

*In-place database migration* means re-using the old repository and letting WebSphere InterChange Server perform the repository upgrade during first startup of the server.

The existing repository database and other database tables are migrated intact to the new database tables using database migration tools such as database instance migrate and database migrate when the new database management system version is installed.

In-place database migration is not possible if customers want to change their database management system (for example, from Oracle to IBM DB2 or from Microsoft SQLServer to IBM DB2).

If you want to use the old repository, the database software must be upgraded to the versions that WebSphere InterChange Server Version 4.3.0 supports.

With in-place database migration, use the old repository database to start the newly installed server. The export and import using **repos\_copy** is not necessary. Users simply point the new installation of WebSphere InterChange Server to the original repository database. WebSphere InterChange Server automatically upgrades the repository during first server startup.

With in-place database migration, customers can save time moving because no repository or runtime data has to be moved.

With in-place database migration the migration of incomplete events is facilitated.

### 4.1.2 Without-in place database migration

*Without in-place database migration* means that the repository is exported and then imported using the **repos\_copy** utility into a new database.

Without in-place database migration can involve less production downtime than an in-place database migration. This is dependent on the following factors:

- Whether the migration takes place on a different machine



- The complexity of the WebSphere ICS environment that is migrated

Without in-place database migration involves more detailed execution of the **repos\_copy** command to capture all data, especially from significantly earlier WebSphere InterChange Server versions such as IBM CrossWorlds ICS V4.1.1.

Without in-place database migration does not propagate incomplete events to the new WebSphere InterChange Server v4.3 repository database or WebSphere MQ queues. Without in-place database migration may require additional machines and hardware resources, especially if the new WebSphere InterChange Server Version 4.3.0 is installed on a new or different machine.

## 4.2 Hardware considerations

The following hardware issues must be considered when planning the migration to WebSphere InterChange Server Version 4.3.0 from previous versions:

- Hard disk storage space beyond what is listed in the minimum requirements (Table 4-1) to deploy a new WebSphere InterChange Server environment for any version, as listed in their respective install guides:
  - More hard disk storage space than listed in the table is needed for complete system backups.
  - More hard disk storage space is required for the new WebSphere InterChange Server software environment, especially if the in-place or without in-place migration is performed on the same machine.
  - More hard disk storage space is required in fully developed, complex WebSphere InterChange Server environments with large repositories and relationship tables.

Table 4-1 Minimum hardware requirements

	AIX	Windows
Processor	IBM pSeries® 610 6E1 class or equivalent 375 Mhz	Pentium® III 1 Ghz
Memory	512 MB	512 MB
Hard disk storage space	40 GB	20 GB

- CPU capacity that is required for a migration will not differ substantially from a production environment:
  - The WebSphere InterChange Server is in a quiesced state during migration, which can imply that less processing power is required.

- However, repository migration, especially large repository migrations, will fully utilize CPU capacity and be much quicker with faster hardware and more memory.
- Even though WebSphere InterChange Server 4.3.0 incorporates significant performance improvements, a timely and successful migration requires the availability of more CPU or processing power and memory above the minimum level cited in the installation guides. Table 4-2 lists the recommended hardware configuration.

*Table 4-2 Recommended hardware configuration*

	<b>AIX</b>	<b>Windows</b>
Processor	Dual Processor IBM pSeries 610 6E1 class or equivalent 375 Mhz minimum	Pentium IV 1 Ghz
Memory	1024 MB	1024 MB
Hard disk drive space	40 GB	40 GB

- Depending on the migration method, additional hardware and machines may be required. For example, if a test or staging migration is performed before migrating, the production repository will require additional hardware or machines.

In general, the migration hardware environment must be modeled to at least the WebSphere InterChange Server V4.3 minimum hardware environment with more memory and increased hard disk space storage made available for the migration process and backups.

## 4.3 Software considerations

Most third-party software dependencies have been removed in the WebSphere InterChange Server V4.3 environment. In previous versions, a WebSphere ICS environment deployed the Inprise VisiBroker product as the object request broker agent. Version 4.3 uses the IBM ORB agent. The Version 4.3 environment can be configured as a total IBM software solution under:

- IBM AIX V5.1 or 5.2
- IBM WebSphere MQ 5.3
- IBM JRE/JDK 1.4
- IBM WebSphere InterChange Server V4.3
- IBM WebSphere Application Server 5.1
- IBM Universal Database Enterprise Edition V8.1

Table 4-3 briefly highlights the change progression in the software stack supported to run a typical WebSphere InterChange Server environment for AIX.

*Table 4-3 Software changes in the WebSphere InterChange Server environment*

<b>WebSphere InterChange Server</b>	<b>OS level and JRE/JDK version</b>	<b>DB2 version</b>	<b>WebSphere MQ</b>	<b>JDBC Data Direct Driver</b>	<b>ORB Agent and XML Parser</b>
(IBM CrossWorlds) 4.1.1	AIX 4.3.3 patch 9 1.3.1_02	UDB 7.2FP6	5.2		Inprise VisiBroker
4.2.1	AIX 4.3.3 patch 9 1.3.1	UDB 8.1FP2	5.3.01	3.2	Inprise VisiBroker and Xerces 1.4.3
4.2.2	AIX 5.1 ML 4 AIX 5.2 ML 1 1.3.1	UDB 8.1FP2	5.3.02 CSD 5	3.2	IBM ORB and Xerces 1.4.3
4.2.3	AIX 5.1 ML 5 AIX 5.2 ML 2 1.4.2	UDB 8.1FP5	5.3.02 CSD 7	3.3	IBM ORB and XML4J 4.2.3

In a Windows environment, software versions follow the same progression as in Table 4-3. Refer to 3.3, “System architecture changes” on page 49 for a detailed overview of the specific operating system levels supported for each version.

More software changes that should be considered during migration planning:

- ▶ Major operating system migrations may be required in some AIX WebSphere InterChange Server environments.
- ▶ Changing the object request broker replaces the Object Activation Daemon with WebSphere MQ triggering and requires some modification to Server Access Interface configurations.
- ▶ Changing the object request broker means replacing the VisiBroker naming service (osagent) with CosNaming service, which is shipped with IBM object request broker transient naming server (tnsnameserver). The major difference between these two is that osagent is a dynamic naming service (object request broker client can automatically find the object request broker service) and tnsnameserver is static naming service.
- ▶ The major new object request broker features listed above are covered in detail in Chapter 5, “Technical impact of major design changes” on page 67 of this redbook and must be thoroughly reviewed prior to migration.

- ▶ Some WebSphere InterChange Server environments may require significant development modifications with the availability of a newer JRE/JDK and XML parser.
- ▶ IBM DB2 Universal Database is upgraded from Version 7.2 to IBM DB2 Universal Database 8.1 FixPack 5 when migrating from IBM CrossWorlds 4.1.1 to WebSphere InterChange Server V4.3. IBM DB2 Universal Database fix packs may have to be applied for IBM DB2 Universal Database 8.1.
  - Upgrading the DBMS version and databases from IBM DB2 Universal Database 7.2 to IBM DB2 Universal Database 8.1 is fully supported by IBM with well-documented procedures and useful utilities:  
<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>
  - Actually performing the necessary DB2 procedures requires a database administrator or experienced DB2 professional.
- ▶ IBM WebSphere MQ is upgraded to IBM WebSphere MQ V5.3.02 with FixPack 7 (previously known as CSDs).
  - Upgrading WebSphere MQ versions and queues are also fully supported by IBM with well-documented procedures:  
<http://www.ibm.com/software/integration/wmq/support/>
  - Actually performing the necessary WebSphere MQ upgrade and fix pack procedures requires a WebSphere MQ administrator or an experienced WebSphere MQ professional.

## 4.4 Human resource roles

In this section, we assume that within most organizations specific roles are assigned to human resources. These roles or resources may be shared or delegated, but in planning it is helpful to know who performs a specific role and when that role is available during the migration process. Table 4-4 gives a brief overview of how human resources may be organized in a migration effort.

*Table 4-4 Human resource needs per stage*

Stage	Activity	Role or resource
Stage 1 Pre-migration	Develop migration plan	Project lead
Stage 1 Pre-migration	Hardware and software requirements	Project lead

Stage	Activity	Role or resource
Stage 1 Pre-migration	Backups of existing environment	Project lead, developer, system administrator, database administrator, WebSphere MQ administrator
Stage 2 Install and Migration	Developer workstations	Developer, project lead
Stage 2 Install and Migration	Application software upgrades, OS upgrades	Database administrator, WebSphere MQ administrator, system administrator
Stage 2 Install and Migration	Integrated development changes modifications	Project lead, developer, system administrator, database administrator, WebSphere MQ administrator, Quality Assurance analyst
Stage 2 Install and Migration	Migration steps	Project lead, developer, System administrator, database administrator, WebSphere MQ administrator
Stage 2 Install and Migration	Quality assurance	Project lead, developer, system administrator, Q/A analyst business user
Stage 2 Install and Migration	Production	System administrator, Q/A analyst, database administrator, WebSphere MQ administrator, project lead
Stage 3 Post-migration, Testing, and Remediation	Unit testing	Developer, project lead
Stage 3 Post-migration, Testing, and Remediation	System testing	Project lead, developer, system administrator, Q/A analyst

Stage	Activity	Role or resource
Stage 3 Post-migration, Testing, and Remediation	End-to end functional testing	Project lead, developer, system administrator, Q/A analyst
Stage 3 Post-migration, Testing, and Remediation	System testing	Project lead, developer, system administrator, Q/A analyst

## 4.5 Time considerations

When creating the project plan for migrating WebSphere InterChange Server, we recommend estimating the time needed to complete all required migration tasks. This gives an idea of the expected time frame for migration and any issues such as system down time that should be considered. This section gives advice about time considerations when calculating the total completion time for the migration. Specific timings for actions are not given because these are variable, depending on the individual WebSphere InterChange Server configurations, platforms involved, and volume of WebSphere InterChange Server artifacts.

### 4.5.1 Duration

The first consideration is to calculate an estimate of how long it can take to perform each individual migration task. This is unique to each WebSphere InterChange Server configuration. Calculating each task's duration enables production of an estimated time for pre-migration, migration, and post-migration tasks.

### 4.5.2 Schedule

When an estimate of the different stages of migration has been determined, you can draw up a schedule. It is recommended that for each migration task you consider the following questions:

- ▶ What is the task?
- ▶ When is the expected start?
- ▶ When is the expected completion?
- ▶ What is the absolute deadline?

The migration to WebSphere InterChange Server Version 4.3 requires performing the pre-migration, migration, and post-migration tasks in a specific order. This means that many stages of the migration rely on other tasks as

prerequisites. It is useful to include this information in the schedule to indicate which activities rely on the completion of others.

### 4.5.3 Status

To determine whether migration tasks are proceeding as planned, it is recommended that a mechanism for monitoring progress is defined.

### 4.5.4 Other time considerations

In addition to migration tasks themselves, there are other tasks for which a time for completion must be estimated. Not all of the tasks have to be performed for every migration (it depends upon individual circumstances), but to plan some time in the schedule for them can be a valuable activity:

- ▶ Creating a full system backup
- ▶ Validating the system backup
- ▶ Uninstalling unsupported products
- ▶ Installing supported products
- ▶ Migrating each component of the system
- ▶ Performing fault resolution during and after migration
- ▶ Performing system recovery in case migration fails

## 4.6 Fault planning

The expectation is that if suitable pre-planning is done, resources are available, and a sequential migration procedure is followed, then faults and problems in performing the migration are minimized. With a complex WebSphere InterChange Server deployment, unforeseen problems can occur when minor configuration steps may be missed or unique situations not covered in this document arise. It is crucial that an equal amount of time be devoted to system recovery and elimination of errors encountered during the migration. “Plan to succeed but allow for failure” must be the rule, not the exception, during migration.

## 4.7 Stage checklists

Planning for and assessment of a successful migration is categorized into three major stages: pre-migration, migration, and post-migration. The following tables show some of the considerations for each stage.

Table 4-5 Stage 1 planning checklist

Pre-migration (stage 1)	Completion or assessment
Hardware and software environment planning	Are the prerequisites fulfilled, and is the hardware available for backups and migration?
Human resources planning	Are the right personnel available and in place to perform the migration tasks?
Is existing custom code migratable?	Does custom code or configurations exist that can prevent successful migration?
What impact will be felt on production systems?	Migration can only be performed on quiesced systems; some production downtime is mandatory, however brief.
Does full system backup exist?	Starting migration on systems that are not backed up is not recommended.

Table 4-6 Stage 2 planning checklist

Migration (stage 2)	Completion or assessment
Evaluate and chose a migration path.	Depending on the environment to be migrated and its availability requirements, how long can this environment be out of production?
Create time schedule to complete migration tasks.	Most important consideration in the actual conduct of the migration tasks is allowing sufficient time to complete them and creating a completion schedule.
Are there any skill gaps that can prevent completion of specific migration tasks?	Need to review actual migration steps and ensure they are within the skill capabilities of the personnel assigned to complete them.
Fault tolerance allowance for errors and deployment; possible workarounds.	Even the most skilled make mistakes; allow for recovery time for most steps. Some steps may require workarounds if not applicable in current environment.



Table 4-7 Stage 3 planning checklist

Post-migration (stage 3)	Completion or assessment
Training and education	Are personnel trained on the new version? Certain new features, especially Role-Based Access Control (RBAC) must be deployed in the newest version, a feature that did not exist before.
Has testing been completed to verify no loss of function exists after migration?	What tests failed if any, and why? Are workarounds available or fixes that will be available soon? How far will testing proceed? Will all functions be tested, including new or just critical functions migrated and used by previous version?
Previous system: keep or delete?	After successful migration, if newly migrated versions pass all tests, is there a need to recover previous system or uninstall it to recover space?





## Part 2

# Migration steps

In this part, step-by-step instructions provide for performing migration of the following releases of WebSphere InterChange Server (WebSphere ICS) to Version 4.3 on Windows and AIX:

- ▶ WebSphere InterChange Server 4.1.1
- ▶ WebSphere InterChange Server 4.2.1
- ▶ WebSphere InterChange Server 4.2.2

Chapter 5, “Technical impact of major design changes” on page 67 gives detailed insight into technical issues of the change from the VisiBroker Object Request Broker to IBM Object Request Broker and the tooling change since WebSphere InterChange Server 4.1.1.

Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 is an overview of WebSphere InterChange Server 4.3 installation on Windows and AIX.

Chapter 7, “Project environment setup” on page 109 covers the scenario used in the following chapters to perform the migration. It provides details of the configuration and collaborations used in relation to the migration, and related

information about setting this up. It is assumed that readers have existing WebSphere InterChange Server scenarios to migrate.

## Technical impact of major design changes

This chapter provides technical details about the impact of major design changes in WebSphere InterChange Server (WebSphere ICS) since Version 4.1.1. This includes configuration steps to make existing scenarios function correctly when migrating to Version 4.3.

These changes do not apply to every previous version of WebSphere InterChange Server. Table 5-1 on page 68 shows the features that are discussed in this chapter and which versions they apply to. An important note after this table concerns all previous versions of WebSphere InterChange Server.

In 5.1, “VisiBroker to IBM ORB change” on page 69, we cover the major changes between VisiBroker Object Request Broker (ORB) and IBM Object Request Broker (IBM ORB). This section must be consulted for more information about the IBM ORB and its use and setup.

In 5.2, “Tooling changes to WebSphere InterChange Server” on page 88, we cover major changes to the tooling and the introduction of the WebSphere Business Integration Toolset.

Table 5-1 Version reference for new features in this chapter

WebSphere InterChange Server version	VisiBroker to IBM ORB change	WebSphere Business Integration Toolset change
4.1.1	See Section 5.1	See section 5.2
4.2.1	See Section 5.1	Not applicable
4.2.2	Not applicable	Not applicable

The IBM ORB is updated in WebSphere InterChange Server Version 4.3 because the JDK has changed version for this release. This means that the IBM ORB that is used in WebSphere InterChange Server 4.3 is different from the one that is used in WebSphere InterChange Server 4.2.2, but this change is not visible to the user.

For Version 4.2.2 users: The IBM ORB *must be stopped* while the new JDK and WebSphere InterChange Server are installed.

**Important:** This information applies to all versions of WebSphere InterChange Server prior to Version 4.3, including Version 4.2.2:

The IBM Object Request Broker, used with WebSphere InterChange Server Version 4.3, does *not* support using a dot (.) in naming. The use of a dot in the name of the WebSphere InterChange Server prevents the tooling and adapters from communicating with the WebSphere InterChange Server because it cannot be registered successfully with the IBM ORB.

If the WebSphere InterChange Server name currently contains one or more dots (for example, ICS.AIX.4.2.1), it must be changed.

If the WebSphere InterChange Server name is changed because of an unsupported name or for any other reason, failed flows cannot be migrated. Any existing failed flows will be lost after migration after the WebSphere InterChange Server name change.

Any failed flows and transactional flows must be resolved before the migration and name change of the WebSphere InterChange Server.

## 5.1 VisiBroker to IBM ORB change

An Object Request Broker (ORB) is a set of libraries and other components that client applications and object servers use to communicate. The ORB makes WebSphere InterChange Server accessible to its clients, the connector agent, and WebSphere Business Integration Toolset. When a WebSphere InterChange Server registers with the ORB's Name Service, clients can obtain the information needed to find and interact with the server. Object-to-object interactions are performed between components by using the Interface Definition Language (IDL) of the ORB. At a transport level, communication uses the Internet Inter-ORB Protocol (IIOP).

In WebSphere InterChange Server Version 4.2.2, the VisiBroker ORB was replaced by the IBM ORB. The primary reason for the change was to reduce dependency on third-party components, thereby improving cross-product support. The change was also expected to result in performance improvement for this area. Figure 5-1 shows the components that make up the IBM ORB.

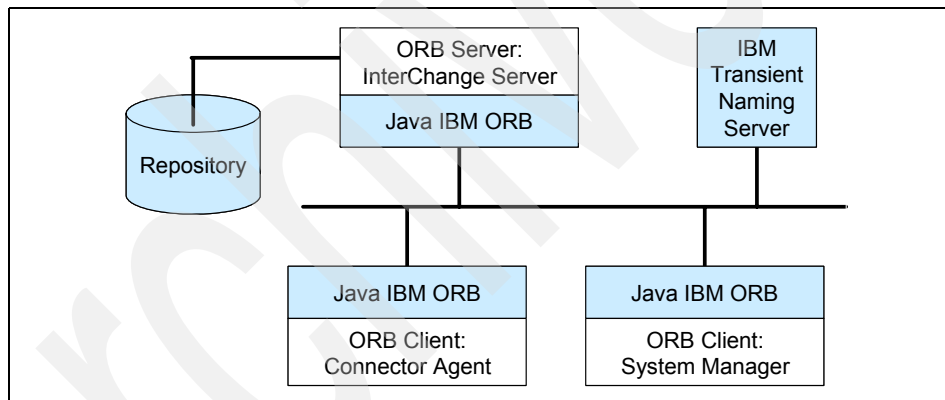


Figure 5-1 IBM ORB components

The naming service provided by the IBM ORB is known as the Transient Naming Server. When a component of the IBM WebSphere Business Integration system starts, it registers itself with the Transient Naming Server, and its Common Object Request Broker Architecture (CORBA) object is stored in the memory of the Transient Naming Server. When the component needs access to another business integration system component, it uses the naming service to determine the information it needs to locate and interact with that component. If the Transient Naming Server fails, its memory contents are lost. As a result, all components that had been registered with it must be rebooted so they can re-register with the naming service.

The Persistent Naming Server, which ships with WebSphere InterChange Server 4.2.2, extends the capability of the Transient Naming Server so that the collection of CORBA objects that are registered with the Transient Naming Server are stored in a naming repository. The existence of the naming repository means that these CORBA references, rather than being only in the Transient Naming Server memory, are persistent and are available to other processes and components in the event that the Transient Naming Server fails. Other components do not have to shut down and restart in order to re-register with the naming service.

As part of its startup process, WebSphere InterChange Server updates the naming repository by copying the CORBA objects currently registered with the Transient Naming Server into the naming-repository file. When each adapter starts, it updates the naming repository with its information. If WebSphere ICS has not yet started when the adapter starts, the naming repository is updated whenever WebSphere ICS does start.

**Restriction:** The IBM ORB has a limitation regarding ORB communication not present in the previous VisiBroker ORB.

When using VisiBroker, multiple Name Servers (osagents) running on a network automatically exchange information when running in the same subnet. As a result, it was very easy to connect tools and adapters to different instances of WebSphere InterChange Server.

However, the IBM Persistent Name Server does not exchange information with other instances of the Persistent Name Server running on the network. Since an adapter or tool can query only a single Name Server, a particular tool or adapter can communicate only with WebSphere InterChange Servers that are registered on the Name Server that it is querying. As a result, this requires the WebSphere ICS, tools, and adapters to use a single Persistent Name Server, typically located on the same host as the WebSphere InterChange Server.

### 5.1.1 IBM ORB setup

This section discusses the setup and configuration of the IBM ORB, including installation. Further information about the IBM ORB and CORBA can be found in the *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide* at:

<http://www.ibm.com/developerworks/java/jdk/diagnosis/>

#### Installation of IBM ORB

The IBM ORB is installed as part of the IBM Java Runtime Environment (JRE) software, which is shipped with WebSphere InterChange Server. If the components reside on a single machine, the installer configures everything



automatically using default values. If the components are across machines, the IP address and port number of the machine with the WebSphere InterChange Server must be configured in the CWSHaredEnv.bat/CWSHaredEnv.sh file.

### **Identification of registered components**

WebSphere InterChange Server provides the CosNameServer\_Dump tool to list all valid WebSphere InterChange Server ORB objects that are registered with the IBM Transient Naming Server. This tool is located in the /bin subdirectory of the product directory. It is invoked with the following command on UNIX:

```
CosNameServer_Dump.bat on Windows or CosNameServer_Dump.sh
```

### **Starting the Persistent Naming Server**

The Persistent Naming Server, which loads the CORBA object references into the naming repository, must be started before the WebSphere InterChange Server is started; otherwise, start of the WebSphere InterChange Server fails.

Run the Persistent Naming Server startup file, which is located in the /bin directory of the WebSphere InterChange Server installation directory. It is called PersistentNameServer.bat on Windows and PersistentNameServer.sh on UNIX. This file performs the following tasks:

- ▶ Starts the Transient Name Server.
- ▶ Loads the naming content from the naming repository into the naming server.

When the WebSphere InterChange Server and any adapter agents start, they update the naming repository.

If the naming server stops for any reason, it can be restarted using the PersistentNameServer.bat/PersistentNameServer.sh file without the need to restart the WebSphere InterChange Server or the adapter agents.

### ***Starting the Persistent Name Server as a Windows Service***

On Windows, the naming service can be started as a Windows Service. In a production system on Windows, it is recommended that you install the naming service as a Windows Service, as this is required in clustered environments. Thus the Persistent Name Server runs even when the current user is logged off.

A utility is provided with WebSphere InterChange Server to enable some of the components to be started as a Windows Service. This utility can be invoked by typing `cwservice` on the command line. An example of the appropriate parameters to use with this utility is:

```
cwservice -xi -tNAMESERVER -cX:\PersistentNameServer.bat -mode=Auto  
-s[Service Name]
```

## Configuring the IBM ORB

All settings of the Transient Naming Server executable are controlled by a batch file: CWSHaredEnv.bat on Windows and CWSHaredEnv.sh on UNIX platforms. By default, the IBM ORB is configured to run on the same machine as the WebSphere InterChange Server and listens on port 14500, but these can be configured so that multiple WebSphere InterChange Servers can connect to a single ORB on a different machine.

Several IBM ORB properties are equivalent to properties that can be set in the VisiBroker ORB, as shown in Table 5-2.

Table 5-2 IBM ORB properties compared to VisiBroker properties

VisiBroker property	IBM ORB property
vbroker.agent.addr	org.omg.CORBA.ORBInitialHost
vbroker.agent.port	org.omg.CORBA.ORBInitialPort
vbroker.se.iiop_tp.scm.iiop_tp.listener.port	com.ibm.CORBA.ListenerPort
vbroker.se.iiop_tp.host	com.ibm.CORBA.LocalHost
ORBtcpTimeout	com.ibm.CORBA.RequestTimeout
vbroker.se.iiop_tp.scm.iiop_tp.dispatcher.threadMax	com.ibm.CORBA.ThreadPool.MaximumSize
vbroker.se.iiop_tp.scm.iiop_tp.dispatcher.threadMax Idle	com.ibm.CORBA.ThreadPool.InactivityTimeout
Vbroker.orb.streamChunkSize	com.ibm.CORBA.BufferSize

There are two ways to change properties for the IBM ORB, either in the startup script or in the WebSphere ICS configuration file, Interchangsystem.cfg. If the properties are specified in both the startup script and the configuration file, then those in the startup script take precedence. Table 5-3 shows the basic properties that can be customized for the IBM ORB. Other properties that can be configured are described in upcoming sections.

Table 5-3 Properties that can be customized for the IBM ORB

IBM ORB property	Configuration file property	Description
com.ibm.CORBA.ListenerPort	OApport	Port number on which the ORB server (within WebSphere InterChange Server) listens for incoming requests.

IBM ORB property	Configuration file property	Description
com.ibm.CORBA.LocalHost	OAipAddr	IP address or host name of the machine on which the ORB server (within WebSphere InterChange Server) is running.
com.ibm.CORBA.ThreadPool.MaximumSize	OAThreadMax	Maximum number of threads that the connection manager can create. The default value (zero) indicates that no size restriction exists.
com.ibm.CORBA.ThreadPool.InactivityTimeout	OAThreadMaxIdle	The time (in seconds) before an idle thread is destroyed.
com.ibm.CORBA.RequestTimeout	None	Number of seconds that a CORBA request waits before timing out. By default, there is no timeout; the ORB waits indefinitely for a response.
com.ibm.CORBA.LocateRequest	None	Timeout value (in seconds) for Locate Requests.
com.ibm.CORBA.FragmentTimeout	None	Maximum length of time that the ORB waits for second and subsequent message fragments before it times out. Set this property to zero to indicate no timeout. The default value is 300000 ms.

In `Interchangesystem.cfg` and adapter local configuration files, IBM ORB properties are specified under the CORBA section. Example 5-1 on page 74 illustrates how to customize the port on which the IBM ORB listens.

In the `CWSharedEnv.sh` or `CWSharedEnv.bat` file, the IBM ORB properties have to be specified using `-D`. This is an example of the ORB properties section from the startup script:

```
set ORB_PORT=14500
set ORB_HOST=localhost
set ORB_PROPERTY=-DORBNamingProvider=CosNaming
-Dorg.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
-Dorg.omg.CORBA.ORBInitialPort=%ORB_PORT%
-Dorg.omg.CORBA.ORBInitialHost=%ORB_HOST%
-Dcom.ibm.CORBA.ThreadPool.InactivityTimeout=1500
-Dcom.ibm.CORBA.Debug.Output=nu1
```

These `-D` Java properties can also be used on the command line when the Transient Name Server is started.

## Configuring debug properties for the IBM ORB

A set of properties relating to debugging can be set for the IBM ORB (Table 5-4).

Table 5-4 Properties related to debugging

Property	Value	Default	Description
com.ibm.CORBA.CommTrace	true or false	false	Used for wire tracing. Every GIOP message entering or leaving the system is written to trace log when this is set to true.
com.ibm.CORBA.Debug	trace or message or any value	false	Set to trace to turn on tracing only, set to message for messages only. Any other value, or an empty value, means both trace and message.
com.ibm.CORBA.Debug.Output	filename or empty	orbtrc.DDMM YYYY.HHmm. SS.txt if value is empty or not specified.	Redirects trace to a file.

## IBM ORB configuration for Access Client connectivity

If WebSphere InterChange Server is being used with an Access Client, the Interchangesystem.cfg file must be configured to contain the OAPort property because it provides a persistent Inter Orb Reference in the .ior file. This task must be performed manually by editing the file. The code in Example 5-1 shows the definition of the OAPort property with a port number 15000. This setting can also be set with System Manager connected to WebSphere InterChange Server.

Example 5-1 CORBA definition in the Interchangesystem.cfg

```
<tns:property>
  <tns:name>OAPort</tns:name>
  <tns:value xml:space="preserve">15000</tns:value>
  <tns:isEncrypted>false</tns:isEncrypted>
  <tns:updateMethod>system restart</tns:updateMethod>
  <tns:location>
    <tns:reposController>false</tns:reposController>
    <tns:reposAgent>false</tns:reposAgent>
    <tns:localConfig>true</tns:localConfig>
  </tns:location>
</tns:property>
```

Further information about setting these properties can be found in the *IBM WebSphere InterChange Server Access Development Guide*. Go to:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left-side pane, expand **Interchange Server and Toolset** → **Development** → **PDFs** and click **Access development** in the right-side pane.

### 5.1.2 Using Server Access Interface with the IBM ORB

Access Clients use IOR files to establish communication with WebSphere InterChange Server. An early version of this file was generated with VisiBroker ORB, but early versions are not compatible with IOR files used by WebSphere InterChange Server 4.3. The IBM ORB generates an IOR file after WebSphere InterChange Server startup, and these are used to replace the old IOR.

If a standalone client relies on an ORB other than VisiBroker ORB or IBM ORB, no changes are required. If the standalone client uses VisiBroker ORB, several changes are required. The sample access client, `DevelopmentKits\sadk\ServerAccessInterfaces\AccessSample` in the WebSphere directory, may be useful in helping to make these changes.

The changes that are required to switch from VisiBroker to IBM ORB are:

- ▶ Non-IBM CORBA JAR references must be replaced with IBM ORB files:
  - For Windows and AIX, no extra .jar files are required.
  - For Solaris and HP-UX, the `ibmorb.jar` in the `%CROSSWORLDS%/jre/lib/ext` directory must be included in the classpath of the client.
- ▶ The `AccessInterfaces.idl` file must be recompiled with the *idle* compiler:
  - The compiler is a part of the IBM JDK on Windows and AIX.
  - The `idlj` compiler for Solaris and HP-UX is in `%CROSSWORLDS%/jre/ibm_bin`.
- ▶ The client code must initialize the IBM ORB instead of the VisiBroker ORB.

#### ***Code changes required for Access Clients***

The code shown in Example 5-2 (the original code) and Example 5-3 on page 76 is used to provide an example of some of the custom code changes that are needed for Access Clients to work correctly with the new IBM ORB.

#### *Example 5-2 Previous Access Client code*

---

```
Properties orbProperties=new java.util.Properties();
orbProperties.setProperty("org.omg.CORBA.ORBClass","com.inprise.vbroker.orb.ORB");
```

```
orbProperties.setProperty("org.omg.CORBA.ORBSingletonClass","com.inprise.vbroker.orb.ORBSingleton");  
org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init((String[])null, orbProperties);
```

---

The two lines that contain the `setProperty()` method calls the reference that the VisiBroker classes need to have removed. This enables the framework to use the default ORB implementation of the JDK rather than VisiBroker.

Access client can still use `IgetInterchangeAccessSession()` method to get its access session using the following syntax:

```
accessSession = accessEngine.IgetInterchangeAccessSession(userName,  
    password);
```

However, with the new security mechanisms introduced in WebSphere InterChange Server 4.3, the usage shown in Example 5-3 is recommended.

*Example 5-3 Access Client code*

---

```
Properties props = new java.util.Properties();  
props.put("username", "admin");  
props.put("password", "null");  
accessSession =  
Server.SecurityServices.loginClient.SecureLoginUtility.login(accessEngine ,  
    props);
```

---

The usage shown in Example 5-3 encodes the user name and password to set security, preventing information being insecure during transmission.

Ensure that the following WebSphere InterChange Server .jar files are included in the classpath:

- ▶ `CrossWorlds.jar` (%CROSSWORLDS%\lib ).
- ▶ `ibmorb.jar` (%CROSSWORLDS%\jre/lib).

See the *IBM WebSphere InterChange Server Access Development Guide* for more examples of client code and changes that illustrate initializing IBM ORB.

If Access is used from within a servlet, the IBM ORB is contained in the WebSphere Application Server runtime and the same changes as detailed above must be made. If Access for Enterprise JavaBeans is being used, the IBM ORB is contained in the WebSphere Application Server runtime. The only change required is to remove the VisiBroker JAR references in the classpath. The Access JAR file contains all other necessary artifacts, such as compiled IDL, session bean, and so on.

### 5.1.3 WebSphere MQ triggering replacing Object Activation Daemon

The IBM ORB does not contain the Object Activation Daemon functionality that was available with VisiBroker. WebSphere MQ triggering can be used to provide the same functionality and provides the following additional features:

- ▶ Auto restart of an adapter over the Internet enables the adapter to be restarted anywhere.
- ▶ Remotely control an adapter agent with other applications in addition to WebSphere InterChange Server.

#### What is WebSphere MQ triggering?

Triggering is a facility supplied by WebSphere MQ that enables an application to be started automatically when a triggering event occurs. A WebSphere MQ queue manager defines certain conditions that constitute a trigger event. When a trigger event occurs, the queue manager sends a trigger message to an initiation queue. The presence of the trigger message on the initiation queue indicates that a trigger event has occurred.

The program that processes the initiation queue is called a trigger monitor application, and its function is to read the trigger message. Based on the contents of the trigger message, the trigger monitor selects an appropriate action. Usually this action is to start some other application to process the queue that generated the trigger message.

If triggering is enabled for a queue, a process-definition object can be created and associated with it. This object contains information about the application that processes the message that caused the trigger event. If the process-definition object is created, the queue manager extracts this information and places it in the trigger message for use by the trigger monitor application. Each queue can specify a different process definition, or several queues can share the same process definition.

WebSphere InterChange Server uses WebSphere MQ triggering to remotely start adapters by placing trigger messages onto *agent activation* queues. These messages invoke the startup script for the adapter. When the adapter has successfully contacted WebSphere InterChange Server, the trigger messages are removed from the agent activation queues by the WebSphere InterChange Server.

These definitions may be useful in understanding triggering in WebSphere MQ:

<b>Application queue</b>	A local queue with triggering set on it.
<b>Process definition</b>	An application queue can have a process definition object associated with it that holds details of the

	application that get messages from the application queue.
<b>Transmission queue</b>	Required if a trigger has to start a channel.
<b>Trigger event</b>	Causes a trigger message to be generated by the queue manager. This message usually arrives on an application queue, but it can also occur at other times.
<b>Trigger message</b>	Created by the queue manager when it recognizes a trigger event. It copies into the trigger message information about the application to be started. This information comes from the application queue and the process definition object associated with this queue.
<b>Initiation queue</b>	A local queue on which the queue manager puts trigger messages. A queue manager can own more than one initiation queue, and each one is associated with one or more application queues.
<b>Trigger monitor</b>	A continuously running program that serves one or more initiation queues. When a trigger message arrives on an initiation queue, the trigger monitor retrieves the message. The trigger monitor uses the information in the trigger message to issue a command to start the application that is to retrieve the messages arriving on the application queue, passing it information contained in the trigger message header, which includes the name of the application queue.
<b>Agent-activation queue</b>	Receives triggering events for an adapter from WebSphere InterChange Server.

### 5.1.4 Setting up WebSphere MQ triggering

This section provides step-by-step instructions for setting up triggering using WebSphere MQ for use with WebSphere InterChange Server. The JText Adapter is used as an example for configuring WebSphere MQ triggering. Steps that are missing from the product documentation are included here.

There are some differences between the way that WebSphere MQ triggering is set up in Windows and UNIX, so separate sections describe each way.

It is recommended that WebSphere MQ triggering should not be set up until after migration has been completed, the WebSphere InterChange Server is running, and the adapter has been tested successfully. The OADAutoRestartRetry property should be set to False for each adapter that used the Object Activation



Daemon prior to migration in order to temporarily disable this function during WebSphere MQ triggering setup.

WebSphere MQ Client or Server must be installed on each remote adapter host. The WebSphere MQ trigger monitor also must be installed and enabled on the remote adapter host. One WebSphere MQ queue must be configured for each adapter to transport the triggering event.

## Setting up WebSphere MQ triggering on Windows

These instructions for setting up WebSphere MQ triggering are specific to Windows. For UNIX, see “Setting up WebSphere MQ triggering on UNIX” on page 83.

### **Create a connector script file for WebSphere MQ triggering**

The following instructions show how to create a startup .bat file for the JText Adapter. The example can be applied to other adapters to generate suitable startup .bat files.

1. Make a copy of the start\_JText.bat file and name it start\_JText\_MQ.bat.
2. Edit the newly created file to make the following changes:
  - a. Change %1 to the adapter name (e.g. JText NOT JTextConnector).
  - b. Change %2 to the ICS name.
  - c. Replace the remaining % variables with the required command line parameters. (For example, replace %3 with -cConfigFile.cfg if a configuration file is being used).
3. Remove the pause at the end of the file.
4. Run start\_JText\_MQ.bat to confirm that the script starts the adapter correctly.

### **Run the WebSphere MQ triggering batch file**

A script is provided with the WebSphere InterChange Server install that helps with the WebSphere MQ portion of the configuration. This script is called mqtriggersetup.bat and is located in the %CROSSWORLDS%\bin directory. This script must be run with the following syntax on the command line:

```
mqtriggersetup WICS_queueManager connName connStartupScript ICSinstance
```

The parameters that are used in the command line syntax are explained as:

<b>WICS_queueManager</b>	Name of the WebSphere MQ queue manager configured for use with WebSphere ICS.
<b>connName</b>	Name of the adapter for which remote restart is being enabled. For example, the JTextConnector has a

	connName of JText. The trailing “Connector” part of the name is removed.
<b>connStartupScript</b>	The full path name for the startup script for the connName Adapter. This startup script has the name start_ <i>connName</i> .
<b>ICSInstance</b>	The name of the WebSphere ICS instance.

To configure and run the WebSphere MQ triggering bat file:

1. Navigate to the /bin directory of the WebSphere InterChange Server install and open the mqtriggersetup.bat file in a text editor such as Notepad.
2. Open a command window and enter the syntax for the **mqtriggersetup** command in the following format:

```
mqtriggersetup WICS_queueManager connName connStartupScript ICSInstance
```

For example, the **mqtriggersetup** command used with the JText Adapter is:

```
mqtriggersetup ICS_QUEUE_MANAGER JText
C:\IBM\WebSphereICS\connectors\JText\start_JText_MQ.bat MYICSSERVER
```

3. Ensure that the command executes successfully by examining the output to the console window.
4. Launch the WebSphere MQ Explorer and examine the queues for the queue manager. These queues must be present:
  - INITIATION.QUEUE
  - AGENTACTIVATIONQUEUE/JTEXTCONNECTOR
5. Expand the **Advanced** folder under the queue manager and click **Process Definitions**.
6. Confirm that a trigger process is present; for example, PROCESS.JTEXT.TRIGGER.

If the path to the batch file, the adapter name, or the InterChange Server name has been mistyped or has to be changed, it can be corrected here. To do this:

- a. Right-click the process definition and select **Properties**.
- b. Make any required changes.
- c. Click **OK**.

### ***Test that triggering works without InterChange Server***

Is it useful to validate that WebSphere MQ triggering has been set up and is working correctly without using WebSphere InterChange Server:

1. Open a command window and enter the following command, substituting where appropriate the queue manager name:

```
runmqtrm -m WICS_queueManager -q INITIATION.QUEUE
```

2. Put a test message on the agent activation queue; for example, AGENTACTIVATIONQUEUE/JTEXTCONNECTOR queue.
3. Confirm that the adapter is started successfully.
4. Clear all messages from the agent activation queue after testing; in this example, clear the messages from the AGENTACTIVATIONQUEUE/JTEXTCONNECTOR queue.

### ***Configure the adapter for remote restart capability***

The adapters have to be configured for remote restart capability. This section provides instructions for configuring an adapter to restart automatically if the process stops, using the JText adapter as an example. This configuration also provides the functionality to manually restart the remote adapter.

1. Launch the System Manager.
2. Make the following changes to the Adapter properties. If the adapter has already been deployed, connect to the server and make the modifications. The changes take effect immediately. If the adapter is not yet deployed, make the changes in the component library and then deploy.
  - a. Set OADAutoRestartAgent to true.
  - b. Set OADMaxNumRetry to an appropriate value. (This is the number of times that the server attempts to restart the adapter.)
  - c. Set OADRetryTimeInterval to the number of minutes that can elapse between attempts to restart an adapter.

### ***Testing triggering with WebSphere ICS***

You can verify that triggering now works correctly using WebSphere InterChange Server before configuring the trigger monitor as a Windows Service. To test triggering using WebSphere InterChange Server:

1. Open a command window and enter the following command, substituting the appropriate queue manager name:

```
runmqtrm -m WICS_queueManager -q INITIATION.QUEUE
```

2. Open a new command window and start the adapter; for example, for the JText adapter, type a command similar to:

```
C:\IBM\WebSphereICS\connectors\JText\start_JText_MQ.bat
```

3. After the adapter has started successfully, press Ctrl+C in its command window. This aborts the adapter and causes a trigger message to be sent from WebSphere InterChange Server to the WebSphere MQ trigger.
4. Confirm that the following message is seen in the command window of the trigger monitor:

```
Waiting for a trigger message
start C:\IBM\WebSphereICS\connectors\JText\start_JText.bat JText
MYICSSERVER "TMC      2AGENTACTIVATIONQUEUE/JTEXTCONNECTOR
PROCESS.JTEXT.TRIGGER
start C:\IBM\WebSphereICS\connectors\JText\start_JText.bat JText
MYICSSERVER
                                MY_QUEUE_MANAGER
"
End of application trigger.
```

---

```
Waiting for a trigger message
```

Additionally, a new command window opens and the adapter starts.

5. To complete these steps, shut down the adapter in the System Manager to prevent the ICS from trying to automatically restart the agent while completing the next section. Right-click the adapter and select **Shut Down adapter name**, for example, **Shut Down JText Connector**.

### ***Install the trigger monitor service***

After WebSphere MQ triggering has been configured and verified, it is recommended that you install the trigger monitor as a service on Windows. Follow the instructions below to install the trigger monitor as a service.

1. Press Ctrl+C in the command window where the trigger monitor is manually running.
2. Launch WebSphere MQ Services.
3. Expand the local WebSphere MQ Services folder.
4. Right-click the queue manager and select **New → Trigger Monitor**.
5. On the Parameters tab, enter the value INITIATION.QUEUE
6. On the General tab, set the start type to **Automatic**.
7. Start the service.
8. Return to the System Manager. In the InterChange Server Component Manager, right-click the JTextConnector and select **Boot Agent**.
9. Verify that the adapter is running by using one of the WebSphere InterChange Server tools such as System View, the Web-based System Monitor, or the CosNameServer\_Dump.bat file. It must be verified in this way because no console windows are opened while the adapter is running as a service.

## Setting up WebSphere MQ triggering on UNIX

This section gives instructions for setting up WebSphere MQ triggering for UNIX and includes details of changes that are specific to any particular flavor of UNIX. For Windows-specific instructions, see “Setting up WebSphere MQ triggering on Windows” on page 79.

The JText Adapter is used as an example for configuring WebSphere MQ triggering. An additional prerequisite for UNIX platforms when configuring triggering is to ensure that Fix Pack 7 for WebSphere MQ is installed.

### ***Run the WebSphere MQ triggering script file***

A script is provided with the WebSphere InterChange Server install that helps with the WebSphere MQ portion of the setup. This script is called `mqtriggersetup.sh` and is located in `%CROSSWORLDS%/bin` directory. This script must be run with the following syntax on the command line:

1. Navigate to the `/bin` directory of the WebSphere InterChange Server install.
2. Examine the `mqtriggersetup.sh` file using the command:

```
$ more mqtriggersetup.sh
```

3. Run the `$CROSSWORLDS/bin/mqtriggersetup.sh` script with the appropriate syntax as shown:

```
mqtriggersetup WICS_queueManager connName connStartupScript
```

The parameters that are used in the command line syntax are explained as:

<b>WICS_queueManager</b>	Name of the WebSphere MQ queue manager configured for use with WebSphere ICS.
<b>connName</b>	Name of the adapter for which remote restart is being enabled. For example, the JTextConnector has a <code>connName</code> of JText. The trailing “Connector” part of the name is removed.
<b>connStartupScript</b>	The full path name for the startup script for the <code>connName</code> Adapter. This startup script has the name <code>connector_manager_connName</code> .

The final command looks similar to this:

```
$CROSSWORLDS/bin/mqtriggersetup.sh ITS0.queue.manager JText  
/opt2/af26/bin/connector_manager_JText
```

4. Ensure that the command executed successfully by examining the output to the console window.
5. Use `runmqsc` commands or the WebSphere MQ Explorer remotely to examine the queues for the queue manager. The following queues can be seen:
  - INITIATION.QUEUE

– AGENTACTIVATIONQUEUE/JTEXTCONNECTOR

6. Use **runmqsc** commands or WebSphere MQ Explorer remotely to examine the process definitions. The PROCESS.JTEXT.TRIGGER process definition must be present.

If values for the path to the adapter manager script file or the adapter name must be changed, they can be modified using these tools.

### ***Additional configuration steps for WebSphere MQ on UNIX***

These additional steps must be taken to configure WebSphere MQ triggering on UNIX. (More steps that are specific to AIX are described in “WebSphere MQ environment settings specific to AIX” on page 85.)

Learn more about these steps in the *WebSphere InterChange Server System Installation Guide for UNIX* section about setting up an Object Activation Daemon. For this guide and more about other UNIX platforms, go to:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left-side pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **UNIX** in the right-side pane.

**Attention:** Be aware of this issue when using **runmqtrm** to invoke the WebSphere MQ trigger monitor daemon on UNIX: The mqm user is the owner of the trigger monitor, but it is possible that mqm does not have the correct paths and permissions to execute commands initiated by another user (for example, cwadmin).

To circumvent this issue, you can make a copy of the **runmqtrm** command which is then disassociated from the mqm user and group, and is used solely for the initiation of the WebSphere MQ trigger monitor daemon.

1. Run this command to make a copy of runmqtrm called runmqtrm2:

```
cp /opt/mqm/bin/runmqtrm /opt/mqm/bin/runmqtrm2
```

2. Run this command to remove the user and group settings from runmqtrm2:

```
chmod ug-s /opt/mqm/bin/runmqtrm2
```

The **runmqtrm2** command is then used to invoke the WebSphere MQ trigger monitor daemon, which is owned by the user issuing that command.

Further changes are required to ensure that the user issuing the runmqtrm2 command has all of the permissions required to access the queue manager, the initiation queue, and the dead letter queue. The **setmqaut** command can be used to give selected groups the authority to access WebSphere MQ objects, but note that individual users can not be given authority.

In Example 5-4, a user who is a member of the appdev security group needs authority to run **runmqtrm2**. The queue manager name is CALVIN, the initiation queue is CALVIN.INITQ, and the dead letter queue is SYSTEM.DEAD.LETTER.QUEUE. These commands give the appdev security group authority to run the trigger monitor.

*Example 5-4 The setmqaut command*

---

```
setmqaut -m CALVIN -t qmgr -g appdev +connect +inq  
setmqaut -m CALVIN -t queue -n CALVIN.INITQ -g appdev +get  
setmqaut -m CALVIN -t queue -n SYSTEM.DEAD.LETTER.QUEUE -g appdev +put +inq  
+passall
```

---

Note that after running these commands, every member of the appdev security group has permission to access the queue manager.

**WebSphere MQ environment settings specific to AIX**

The instructions in this section relate to settings related only to AIX. Two environment variables must be exported for starting the trigger monitor:

- ▶ export AMQ\_NO\_SIGWAIT\_SIGTRAP=1
- ▶ export MQS\_NO\_SYNC\_SIGNAL\_HANDLING=1

More details can be found in Technote 1159677. Search for this on the IBM Support Web site:

<http://www.ibm.com/support>

**Test whether triggering works without WebSphere ICS**

You can test whether WebSphere MQ triggering is configured and working correctly without using WebSphere InterChange Server. Use these instructions to test WebSphere MQ triggering with the adapter:

1. Run the **runmqtrm2** command with the following syntax, substituting the appropriate queue manager name:  

```
runmqtrm2 -m ICS.queue.manager -q INITIATION.QUEUE
```
2. Put a test message to the agent activation queue; for example, AGENTACTIVATIONQUEUE/JTEXTCONNECTOR queue.
3. Confirm that the adapter starts successfully.
4. Clear all messages from the agent activation queue after testing.

**Configure the adapter for remote restart capability**

To configure an automatic restart of an adapter if the process stops (with functionality to restart the remote adapter manually):

1. Launch the System Manager.

2. Make the following modifications to the Adapter properties. If the adapter is deployed, connect to the server and make the changes; they will take effect immediately. If the adapter is not yet deployed, make the changes in the component library and then deploy.
  - a. Set **OADAutoRestartAgent** to true.
  - b. Set **OADMaxNumRetry** to a suitable value for the number of times that the server attempts to restart the adapter.
  - c. Set **OADRetryTimeInterval** to the number of minutes that can elapse between attempts to restart an adapter.

### ***Testing triggering with WebSphere InterChange Server***

To verify that WebSphere MQ triggering is configured correctly for use with WebSphere InterChange Server:

1. Enter the following command, substituting the queue manager name:
 

```
runmqtrm2 -m ITS0.queue.manager -q INITIATION.QUEUE
```
2. Start the adapter (in this example, the JText adapter) with this command:

```
connector_manager_JText -start
```

3. Verify that the adapter started.
4. Run the following command to abort the adapter and cause a trigger message to be sent from WebSphere ICS to the WebSphere MQ trigger monitor:

```
connector_manager_JText -kill
```

5. Check for a message similar to the following in the trigger monitor window:

```
Waiting for a trigger message
exec /opt2/af26/bin/connector_manager_JText -start 'TMC
2AGENTACTIVATIONQUEUE/JTEXTCONNECTOR PROCESS.JTEXT.TRIGGER
exec/opt2/af26/bin/connector_manager_JText-start
ITS0.queue.manager
The connector name is JText
The options used for this connector are -b TMC
2AGENTACTIVATIONQUEUE/JTEXTCONNECTOR PROCESS.JTEXT.TRIGGER exec
/opt2/af26/bin/connector_manager_JText -start ITS0.queue.manager
Starting the JText Connector Script
  UID  PID  PPID  C   STIME   TTY   TIME CMD
  cwadm43 100642    1   9           0:00 <defunct>
Checking if there is another connector agent with same name running...
org.omg.CORBA.OBJECT_NOT_EXIST:  vmcid: 0x0  minor code: 0  completed:
No
No response from agent AIX43JTextConnector. It is probably not running
yet.
Unable to connect to Agent JTextConnector
```



```

Check passed, now start the connector agent...
Logging output to /opt2/af26/logs/connector_manager_JText.log
  UID   PID  PPID  C   STIME   TTY   TIME CMD
  cwadm43 96280 97008 105 10:09:14 pts/14 0:04
/opt2/af26/AdapterJRE/bin/jav
The JText Connector has been Started
Error starting triggered application.

```

---

Waiting for a trigger message

The message Error starting triggered application can be ignored.

## Triggering troubleshooting tips

The following tips can be useful when attempting to troubleshoot triggering on WebSphere MQ.

- ▶ The trigger monitor can be restarted using this command-line syntax.  
To invoke the trigger monitor:
 

```
runmqtrm -m WICS_queueManager -q INITIATION.QUEUE
```
- To start the client trigger monitor:
 

```
runmqtmc -m WICS_queueManager -q INITIATION.QUEUE
```
- ▶ Clear the AGENT.ACTIVATION and INITIATION.QUEUE queues of stale messages. The trigger monitor may start the triggering process if there are any stale messages on the queues. Stale messages may be generated if WebSphere MQ triggering was set up incorrectly, or the if the trigger monitor or WebSphere InterChange Server was stopped unexpectedly.
- ▶ Problems can occur if the length of the path to the adapter startup script is too long.
- ▶ Send log and trace information to a file so that it can be used at a later time to help troubleshoot problems.
- ▶ Check the syntax of the **mqtriggersetup** command. Mistyped parameters are a common problem.
- ▶ The supplied trigger monitors do not log error messages if the trigger monitor fails to start an application.
- ▶ Remove **pause** commands from scripts; otherwise, it is possible to end up with orphaned processes.

- The **BootAgent** command may appear to fail if it was submitted after shutting down or restarting the agent. It may be necessary to wait up to the number of minutes that `OADRetryTimeInterval` is set to.

For example, if `OADRetryTimeInterval` is set to 10 minutes, and the agent is restarted, the **BootAgent** does not work for the 10 minutes following the restart of the agent.

Refer to the WebSphere InterChange Server and WebSphere MQ product documentation for more information about WebSphere MQ triggering. See:

<http://www.ibm.com/software/integration/wbiserver/ics/library/infocenter/>

For WebSphere MQ product documentation, select **latest multiplatform books** in the WebSphere MQ Messaging menu at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

Choose the *WebSphere MQ System Administration Guide* for the appropriate information. For this book, the WebSphere MQ version that is required is 5.3.

## 5.2 Tooling changes to WebSphere InterChange Server

This section about tooling changes in WebSphere InterChange Server is for users migrating from WebSphere ICS Version 4.1.1 to Version 4.3, because the tooling changes are most significant for these users. Brief information is given for the new functionality that the tooling provides, but detailed specifics are not discussed. This section concentrates on changes that require some kind of updates to the data or configuration or that provide significant new functionality.

Table 5-5 lists major supported tooling additions and changes since Version 4.1.1. Many of the new and enhanced tools are discussed in this chapter, and further information about role-based security can be found in 2.3, “New features in WebSphere InterChange Server 4.3” on page 20.

*Table 5-5 Supported tooling since WebSphere InterChange Server 4.1.1*

Tools	V4.1.1	V4.2.0	V4.2.2	V4.3
Business Object Designer	✓	✓	✓	✓
Map Designer	✓	✓	✓	✓
Connector/Adapter Designer	✓	✓	✓	✓
Relationship Designer	✓	✓	✓	✓
Relationship Manager	✓	✓	✓	✓

Tools	V4.1.1	V4.2.0	V4.2.2	V4.3
Process Designer	✓	✓	✓	✓
System Monitor	✓	✓	✗	✗
System View & Statistics	✗	✗	✓	✓
Web-based System Manager	✗	✓	✓	✓
Flow Manager	✓	✓	✓	✓
Business Object Probes	✗	✓	✓	✓
Activity Editor	✗	✓	✓	✓
Failed Event Manager	✗	✗	✗	✓
Integrated Test Environment	✗	✓	✓	✓
Collaboration Debugger	✗	✗	✓	✓
Role-based Security	✗	✗	✗	✓

The WebSphere InterChange Server development model has changed significantly since Version 4.1.1. In 5.2.1, “WebSphere InterChange Server development model” on page 89 an overview of the WebSphere InterChange Server development model shows how the new tools fit into the model.

Further information about tooling and how to use the tools can be found in the product documentation. A high-level view of the function of the different tools can be found in Chapter 3, “What is new?” on page 29 and at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

The IBM WebSphere Business Integration Toolset Version 4.2 replaces the IBM CrossWorlds Full Toolkit, supplying enhanced National Language Support from U.S. English and Japanese to include French, Italian, German, Spanish, Brazilian Portuguese, Korean, Simplified Chinese, and Traditional Chinese.

## 5.2.1 WebSphere InterChange Server development model

The WebSphere InterChange Server development model has changed significantly since Version 4.1.1, so this section provides a brief introduction to the new development model. The model details the steps that are expected to be necessary for developing and deploying a successful WebSphere Business Integration System using WebSphere InterChange Server. Find more about the

development model in the *Implementation Guide for WebSphere InterChange Server* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Development** → **PDFs** and click **System implementation** in the right-side pane.

## Development environment

Integration components for WebSphere InterChange Server are developed using the WebSphere Business Integration Toolset. These components are developed locally and stored in an integration component library on the local file system. This is different from the previous model of developing directly with component definitions in the repository. An integration component library can be imported, exported, or stored within a version-control system in order to facilitate group development. A *user project* contains links to the components in the integration component library. The library and user project are created using the System Manager described in 5.2.4, “System Manager” on page 92.

After development, completed components are deployed using the user project to the local WebSphere InterChange Server instance for testing.

If multiple developers are involved in creating components for a single integration solution, then the various components must be combined into a single integration component library. This collection should then be deployed for integration testing, even if each of the components has already been tested individually.

## Test environment

After the integration components have been deployed to the local WebSphere InterChange Server instance, they can be tested in depth before being released into a production environment. The new Integrated Test Environment tool is provided with WebSphere InterChange Server to assist with component testing.

It is recommended that performance testing be carried out after any integration testing has been completed. This attempts to replicate the production environment as closely as possible to be most accurate.

## Production environment

After completion of satisfactory testing to determine that the Business Integration System satisfies functional and performance requirements, the integration components can be migrated into the production environment:

- ▶ Export a package that contains the components in the business integration system.
- ▶ Create an integration component library in the production environment.

- ▶ Import the package into the new library.
- ▶ Update the component definitions as necessary for environment-specific properties.
- ▶ Deploy the package of modified components into the production server repository.

## 5.2.2 Configuration files updated to XML

A major change was made to the configuration files used with WebSphere InterChange Server since Version 4.2.0 to convert their format to XML. This fit with the major tooling changes made in V4.3.0, with a number of the tools now running on the Eclipse platform. These configuration file changes include server and adapter configuration.

When migrating to a WebSphere InterChange Server version newer than 4.1.1, any manual changes made in the configuration files must be replicated in the new WebSphere InterChange Server 4.3 configuration files.

If the server and some connectors were sharing the configuration file, it is recommended that you generate a new configuration (.cfg) file. For each adapter, open the existing connector configuration file in Connector Configurator. Check that the data is correct, save the file to the disk, and then start the adapter.

## 5.2.3 Eclipse platform

The Eclipse platform is an open-source integrated development environment (IDE) for the creation of tools. It provides tools developers with a development kit and runtime for writing plugins that enable the user to work with a particular type of resource. IBM has two branded versions of the Eclipse platform:

- ▶ WebSphere Studio WorkBench
- ▶ WebSphere Studio Application Developer Integration Edition.

WebSphere Studio Workbench is an IBM-branded release of the Eclipse platform. This product is delivered and installed with WebSphere InterChange Server.

WebSphere Studio Application Developer Integration Edition is an IBM-branded release of the Eclipse platform, but it can also be used to develop new plugins. It is not delivered with WebSphere InterChange Server because the ability to create new plugins is not required to develop integration components. If it is already installed, it can be used to run the required System Manager and Integrated Test Environment plugins, and the WebSphere InterChange Server installer provides an option to update it to install the necessary plugins.

WebSphere InterChange Server provides these tools that run in the Eclipse environment as plugins:

- ▶ System Manager
- ▶ Integrated Test Environment
- ▶ Collaboration Debugger

These tools are described in the following sections of this chapter.

### ***Eclipse workbench***

The workbench is the collection of perspectives, editors, and views that are active in the Eclipse-based tooling framework, which are in turn affected by the collection of plugins that are installed and enabled. It is a general term used to refer to the Eclipse-based interface, independent of the tool in use.

### ***Workspace***

A workspace is a container: a directory in the file system where projects are stored. This may be a default location or a user-defined location, depending on the product that Eclipse was installed by.

### ***Projects***

Projects are groups of resources that are stored in directories on the file system. Different types of projects have different functions, and often have related builders or compilers that are invoked by the different Eclipse plug-ins. An example of a project is an integration component library, which is a type of project specific to WebSphere InterChange Server.

## **5.2.4 System Manager**

System Manager replaced the CrossWorlds System Manager in WebSphere InterChange Server Version 4.2.0. This is now also re-implemented using the Eclipse open standard for tooling. The System Manager is now the main hub for storing object definitions during development, and other development tools can be launched from inside the interface, including Business Object Designer, Connector Configurator, Map Designer, Process Designer, and Relationship Designer. Source code management is also available through the Eclipse framework through third-party programs such as CVS and Rational® ClearCase®.

Object definitions are saved in integration component libraries in the System Manager. This means that development of WebSphere InterChange Server components is performed locally rather than being done in the WebSphere InterChange Server instance. In order to migrate from WebSphere InterChange Server V4.1.1, it is necessary to create an integration component library within

the System Manager. After the library has been created, components from the WebSphere InterChange Server instance can be imported.

Refer to Chapter 8, “Migration procedures for WebSphere InterChange Server V4.1.1.9 to V4.3” on page 119 for the steps to create an integration component library.

### 5.2.5 System Monitor

The Web-based System Monitor enables monitoring of the WebSphere InterChange Server system from the Web without installing the toolset. Its functionality is similar to the System Manager, but despite additional monitoring functions, it cannot perform dynamic changes to components.

The Web-based System Monitor is now known as the System Monitor in the WebSphere InterChange Server product documentation, which includes sections about monitoring the system with the System Monitor and the System Manager.

#### Monitoring the system with the System Monitor

The System Monitor provides a quick, high-level view into the health of the system and facilitates management of key components. Functions that can be performed in the System Monitor (and those provided by the Windows-based System Manager) include:

- ▶ Enabling historical analysis
- ▶ Persistent statistics
- ▶ Selectively saving historical counts by server, collaboration object, and connector instance and graphs
- ▶ Tracking state changes (such as start, stop, and pause)
- ▶ Maintaining statistics across system restarts
- ▶ Configuring the way data is viewed
- ▶ Starting, stopping, and pausing components
- ▶ Business Object Probes

Table 5-6 shows the statistics that are available through the System Monitor.

Table 5-6 Statistics available through System Monitor

Statistics	Persistent statistics	Collaboration statistics	Connector statistics
Server statistics	✓	✗	✗

Statistics	Persistent statistics	Collaboration statistics	Connector statistics
Successful calls	✓	✓	✗
Failed calls	✓	✓	✗
Total calls	✓	✓	✗
Successful events	✓	✓	✗
Failed events	✓	✓	✗
Total events	✓	✓	✗
Total business objects sent	✗	✗	✓
Total business objects received	✗	✗	✓

### ***Prerequisites for the System Monitor***

The System Monitor requires a compatible Web server to run on and compatible browsers on the remote machines. Web server compatibility means supporting JSP™ Version 1.1 or later and Servlets 2.2 or later. The documented supported Web servers for WebSphere InterChange Server Version 4.3 are:

- ▶ WebSphere Application Server 5.0.2.4
- ▶ WebSphere Application Server 5.1
- ▶ Tomcat 4.1.24 using IBM JDK 1.4.2
- ▶ Tomcat 4.1.27 using IBM JDK 1.4.2

Supported browsers are Microsoft Internet Explorer® 5.5 SP2 or higher and Netscape 4.7x. These browsers must have the Adobe® SVG Viewer 3.0 plugin.

In addition, the DB2 client is needed for Business Object Probe data if the WebSphere InterChange Server Repository is using DB2.

Find more information in the WebSphere InterChange Server product documentation in the IBM WebSphere Business Integration Information Center.

- ▶ Product documentation can be found here:  
<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>
- ▶ Documentation for WebSphere InterChange Server 4.3.0 can be downloaded from:  
<http://www.ibm.com/software/integration/wbiserver/ics/library/infocenter/>



Refer to “Online resources” on page 473 for more details.

### ***Database capacity planning***

Persistent Monitoring data is stored in database tables in the WebSphere InterChange Server repository, so capacity planning for the database is required. It is estimated that each row of persisted monitoring data uses approximately 150 bytes. The total space required for persistent monitoring varies depending on how many components are monitored and the frequency of the monitoring. This must be taken into account when planning the size of the repository and when determining whether an increase in size is required.

## **5.2.6 Business Object Probes**

A Business Object Probe monitors business object instance values during runtime. The probe is placed on a transition link during the creation of an activity diagram and is activated or deactivated during runtime through the System Manager Collaboration Properties dialog. Business Object Probes can be used to monitor any business object except for service call links or the incoming transition link for a decision node.

The specific attributes to monitor can be chosen for each business object. The values for these attributes are then presented in a report by the System Monitor.

## **5.2.7 Web-based flow and failed-flow management**

In WebSphere InterChange Server 4.1.1, a single tool was available to manage flows: the Flow Manager. This functionality still exists, but a new Web-based tool, the Failed Event Manager, is available with WebSphere InterChange Server 4.3. This tool is similar to the Java-based Flow Manager. This enables the resolution of failed flows over the Web if role-based security has been configured for WebSphere InterChange Server.

## **5.2.8 Integrated Test Environment**

A new product called the Integrated Test Environment was added to WebSphere InterChange Server in Version 4.2.0. This is an Eclipse-based tool designed to enhance collaboration testing and enable testing of a business process before deployment. It has these additional features:

- ▶ Artifact deployment
- ▶ Artifact state monitor and control
- ▶ Embedded test connector
- ▶ Business Object Probes and tracing
- ▶ Graphical interface highlights data flows during tests
- ▶ Drill-down capability support

- ▶ Business object value review anywhere along the process
- ▶ Step through map transformations and static map debugging
- ▶ Save test data for future test runs and to share among test scenarios
- ▶ ICS server management
- ▶ Access Client

The WebSphere ICS server must be configured as a *test server*. To perform a test in Integrated Test Environment, a test unit must be created, the components deployed to the server, the server started, the adapters emulated, and business objects exchange between the adapters. An ITE test unit is created for each collaboration; group collaborations need multiple ITE test units.

Starting the WebSphere InterChange Server in test mode performs these actions:

- ▶ Starts the WebSphere InterChange Server in the Integrated Test Environment
- ▶ Redirects logging and tracing information to the Integrated Test Environment

The main change introduced by the Integrated Test Environment is the change to the model of development. No actual changes to any existing code have to be implemented for this new functionality.

### 5.2.9 Collaboration debugger

The collaboration debugger tool was added in WebSphere InterChange Server Version 4.2.2. This tool helps in the testing of collaboration objects executing in WebSphere InterChange Server by enabling a user to step through all of the business logic in a collaboration template to ensure that it works as designed. It enables collaboration debugging at the diagram node level and provides functionality for setting multiple breakpoints in a scenario. Relevant data can be collected at each breakpoint to enable the user to collect relevant data about the triggering event and its flow through the scenario.

The tool is implemented in WebSphere Studio Workbench and WebSphere Studio Application Developer Integrated Edition, and is invoked from a System Manager server view or from the Integrated Test Environment collaboration viewer. The collaboration must be deployed to the server and must be running before the debugger can be attached. The debugger supports node-level breakpoints, global variable viewer, and multiple collaboration instances.

### 5.2.10 Activity Editor

The Activity Editor was introduced in WebSphere InterChange Server 4.2.0 to replace the Code Editor for maps and collaboration templates. Activity Editor is

designed to help users define activity graphically and enables users with no Java knowledge to define customized transformations.

The definitions for the activities are defined on a canvas using drag-and-drop components. A subset of commonly used functionalities is provided as graphical icons. Code Generation automatically generates Java code for the graphical activity.

In addition to the Activity Editor producing Java code from the graphical definition, users more familiar with Java can enter custom Java code directly into the editor.





# Installation of WebSphere InterChange Server 4.3

This chapter gives an overview of and hints for the installation of WebSphere InterChange Server (WebSphere ICS) Version 4.3.0:

- ▶ Reasons why upgrading to WebSphere InterChange Server 4.3.0 is recommended
- ▶ Pre-installation considerations, such as which components to install
- ▶ A summary of the WebSphere InterChange Server installation process on Windows
- ▶ A summary of the installation process on AIX

## 6.1 Overview

In Version 4.3.0, IBM improved the reliability and speed of WebSphere InterChange Server. The use of Java 1.4.2 technology significantly increases processing speed, and the new architecture of data handlers enables processing of larger business objects with better performance.

Since introducing WebSphere InterChange Server 4.2.2, IBM has also replaced use of Borland VisiBroker Object Request Broker (ORB) for communication with IBM ORB. In addition to the increase in communication speed, this change also has the advantage of having this ORB component developed by the same vendor as the WebSphere InterChange Server. The result is better integration and reliability between the ORB and WebSphere InterChange Server. The speed and quality of support for the ORB also increases, because the ORB is now a component IBM creates, manages, and repairs if needed.

In addition to increased speed and reliability, IBM has introduced security features in WebSphere InterChange Server 4.3.0. Users and roles can now be defined in WebSphere InterChange Server and are saved in an encrypted format in a database or stored in an LDAP server. Furthermore, auditing mechanisms have been introduced that enable an administrator to track changes made to the WebSphere InterChange Server.

## 6.2 Pre-installation considerations

Before installation, several items should be considered. These were discussed in detail in Chapter 4, “General planning” on page 53 of this book, so this section acts as a summary of the most important considerations prior to installation of WebSphere InterChange Server and its components.

### 6.2.1 Hard disks and file systems

If the machine that hosts WebSphere InterChange Server and its additional components (such as WebSphere MQ and database) contains only a single hard disk—or a RAID entity that the operating system views as a single hard disk—this paragraph can be skipped. If the system has more than one hard disk or more than one RAID entity, take the following into consideration. Both WebSphere InterChange Server (via the database) and WebSphere MQ use significant disk input/output (I/O). The time a hard disk needs to reposition its read/write heads is much higher than the time needed to do the real read or write operation. Reducing the head activity of hard disks via intelligent data distribution over several hard disks and file systems can result in increased performance of WebSphere and the database, which results in better performance of

WebSphere InterChange Server. For exact locations (in terms of paths in a file system) of data storage, consult the installation and implementation manuals of the respective component:

- ▶ *Installation Guide for WebSphere InterChange Server 4.3*

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left-side pane, expand **Interchange Server and Toolset** → **Installation** and choose your format.

- ▶ *Installation Guide and Implementation Guide for WebSphere MQ 5.3*

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

- ▶ *Installation Guide and Implementation Guide for the Relational database management System (RDBMS) used with WebSphere InterChange Server:* Each database system (Oracle, Microsoft SQL Server IBM DB2) has its own set of documentation.

For Oracle, and Microsoft, refer the respective company's main Web site:

<http://www.oracle.com>

<http://www.microsoft.com>

For DB2:

<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>

Refer to the Redpaper *Introduction to WebSphere InterChange Server 4.2.2 Performance Tuning*, REDP-9124, for a more detailed discussion on performance tuning:

<http://publib-b.boulder.ibm.com/abstracts/redp9124.html>

## 6.2.2 Installation packages

In Version 4.1.1, all adapters, InterChange Server, WebSphere Business Integration Toolset, and pre-built collaborations came in a single installation package. For WebSphere InterChange Server 4.3 components are split over the following packages:

- ▶ WebSphere InterChange Server 4.3 and WebSphere Business Integration Toolset 4.3. on Windows
- ▶ WebSphere InterChange Server 4.3. on UNIX
- ▶ Adapter Framework 2.6 on Windows
- ▶ Adapter Framework 2.6 on UNIX
- ▶ Adapter Development Kit (Windows only)

All adapters and some data handlers come separately either for Windows or UNIX. All pre-built collaborations come separately.

One of the major changes in Version 4.3.0 is the separation of the Adapter Framework code completely from the WebSphere InterChange Server code. As a result, Adapter Framework has to be installed on every machine that hosts an adapter and it must not be installed in the same directory as WebSphere InterChange Server 4.3. For exact installation information, refer to the *WebSphere InterChange Server Installation Guide* for UNIX or Windows. Also, keep in mind that only data handlers and adapters from WebSphere Business Integration Adapters 2.6 can be used. For more detailed instructions about installing or upgrading adapters, refer to the individual adapter guides at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

### 6.2.3 Cannot be installed in the same location as old versions

WebSphere InterChange Server 4.3.0 cannot be installed on top of previous versions of WebSphere InterChange Server. On Windows, the previous version of WebSphere InterChange Server has to be uninstalled before the new version of WebSphere InterChange Server can be installed. On UNIX, it is good practice to install WebSphere InterChange Server to a new directory, with a new user ID. Using this approach, it is possible to revert to the previous version of WebSphere InterChange Server if the upgrade or migration is not completed for any reason.

**Important:** As mentioned before, WebSphere InterChange Server 4.3 on UNIX must not be installed using the root user.

## 6.3 Installation on Windows

This section describes the minimal set of steps that must be performed and states that have to be achieved to successfully install WebSphere InterChange Server 4.3 on Windows. For in-depth explanations, refer to these guides:

- ▶ *IBM WebSphere InterChange Server System Installation Guide for Windows*

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left-side pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **Windows** in the right-side pane.

- ▶ Installation Guide and Implementation Guide for WebSphere MQ 5.3

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

- ▶ Installation Guide and Implementation Guide for the Relational database management System (RDBMS) used with WebSphere InterChange Server



Each database system (Oracle, Microsoft SQL Server IBM DB2) has its own set of documentation.

For Oracle, and Microsoft, refer the respective company's main Web site:

<http://www.oracle.com>

<http://www.microsoft.com>

For DB2:

<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>

For the following steps, it is assumed that WebSphere MQ is a prerequisite and must be installed for either delivery transport or for use as the Object Activation Daemon (OAD).

Additionally, many of these steps recommend making a backup. Be sure to test whether the backup can be restored on a different machine without problems.

1. Install or upgrade the database to bring it to a supported level. Follow the instructions provided in the database documentation. If the database is upgraded, take the appropriate database backup steps prior to installation.
2. Install Java Development Kit (JDK) 1.4.2. The JDK is needed to compile maps and collaboration templates. A compatible JDK is provided with the WebSphere InterChange Server 4.3.0 product CD. Add the location of the JDK /bin directory to the beginning of the PATH environment variable.
3. Install WebSphere MQ 5.3.0.2 and apply the latest supported fix pack. To find the latest supported fix pack, visit:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To apply the fix pack, follow the instructions in the WebSphere MQ documentation. If WebSphere MQ is upgraded, take the appropriate backup steps prior to installation.

4. If WebSphere Application Server 5.0.2.4 or 5.1 is installed before running the WebSphere InterChange Server installer, the installer program automatically installs and configures System Monitor and deletes the previous version. This feature is available only on Windows when WebSphere Application Server is located on the same machine as the WebSphere InterChange Server. Install or upgrade WebSphere Application Server to Version 5.0.2.4 or 5.1 to use this feature. The System Monitor can also be installed at a later time. It also supports Tomcat as the Web server. For details, refer to the WebSphere InterChange Server 4.3 Installation Guide for Windows at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

5. Adjust settings for WebSphere MQ, database, and any third-party components involved to the recommended settings. For example, when using DB2 for the database, make sure that more than 50 connections are available

by setting the maxappls parameter. In the migration scenarios tested in this redbook, 75 connections were used.

6. Install WebSphere InterChange Server using the installer. Consider the following during the installation process:
  - Choose an appropriate install location for the product. Keep in mind that during the install process, this location is prepended to the PATH variable and the CLASSPATH environment variable. On Windows, the PATH or CLASSPATH environment variable has a limit of 2048 characters. If there is not sufficient empty space in these environment variables, the trailing part of the PATH and CLASSPATH is truncated.
  - The WebSphere InterChange Server name must not contain dots (.). For example, prd.1 is an invalid name, but prd\_1 is allowed.
  - The WebSphere InterChange Server name must not change if WebSphere MQ is used for delivery transport and failed flows are to be migrated. The queue definitions for WebSphere MQ delivery transport follow the *API/ConnectorName/WebSphereInterChangeServerName* naming convention, which cannot be modified. As a result, the new WebSphere InterChange Server cannot access the data in the previous queue definition.
  - The installer provides the option to install WebSphere InterChange Server as a service. The selected port number must be an unused port. The WebSphere InterChange Server can also be installed as a Windows Service after the installation process. It is recommended that you defer this task to a later time.
7. Install Adapter Framework 2.6 using the AdapterFramework installer.

To enable WebSphere Business Integration Adapters to run with WebSphere InterChange Server 4.3.0, Version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions on how to install adapters, refer to the individual adapter guides.
8. Install XML data handler using the XML data handler installer.
9. Install EmailAdapter using the EmailAdapter installer.

## 6.4 Installation on UNIX (AIX)

This section describes the minimal set of steps that have to be performed and states that must be achieved to successfully install WebSphere InterChange

Server on UNIX, specifically AIX. For in-depth explanations, refer to these guides:

- ▶ *IBM WebSphere InterChange Server System Installation Guide for UNIX*

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left-side pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **UNIX** in the right-side pane.

- ▶ Installation Guide and Implementation Guide for WebSphere MQ 5.3

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

- ▶ Installation Guide and Implementation Guide for the Relational database management System (RDBMS) used with WebSphere InterChange Server: Each database system (Oracle, Microsoft SQL Server IBM DB2) has its own set of documentation.

For Oracle, and Microsoft, refer the respective company's main Web site:

<http://www.oracle.com>

<http://www.microsoft.com>

For DB2:

<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>

For the following steps, it is assumed that WebSphere MQ is a prerequisite and must be installed for either delivery transport or for use as the Object Activation Daemon (OAD).

Additionally, many of these steps recommend making a backup. Be sure to test whether the backup can be restored on a different machine without any problems. A backup that cannot be restored has little value.

1. Confirm that the following groups are present; if not, create them.

- mqm

mqm group is needed for user mqm (WebSphere MQ admin) and ICS Administrator user.

- Database groups

For IBM DB2, this is normally the db2iadm, db2fadm, and db2asadm groups.

2. Confirm that the following users are present and create them not.

- mqm

The administrator for WebSphere MQ, this user must be available.

- icsadmin

Administrator for WebSphere InterChange Server. Its primary group must be mqm, and its secondary group must be the group of the database instance administrator. Any suitable user ID can substitute for icsadmin.

- database\_user

The user ID to access the database. Check the database installation guide and implementation guide for further details. Any suitable user ID can substitute for *database\_user*. For IBM DB2, this is *db2inst1* by default.

**Note:** Make sure that the primary group for icsadmin is mqm and secondary group is the database user group. This can be checked by logging on as the icsadmin. Open a terminal window and type `groups` at the prompt. All groups that the user is part of are displayed.

Example:

```
$ groups
mqm db2admin
$
```

### 3. Create file systems.

Calculate space needed for all required applications (database, WebSphere MQ, WebSphere InterChange Server, Access Framework, adapters, and so on). This can be estimated by checking the file system of an existing system. Take disk usage measurements of the binaries and data during a period of heavy usage. Do not forget to take into account the extra space that will be required if historical data is to be saved or archived.

The location of WebSphere MQ binaries is either `/usr/mqm` for AIX or `/opt/mqm` for all other UNIX. It cannot be modified. WebSphere MQ saves all messages (runtime data) at `/var/mqm/queue manager name/queues/queue name`.

4. Install or upgrade the database to bring it to a supported level. Follow the instructions provided in the database documentation. If the database is upgraded, take the appropriate database backup steps prior to installation.
5. Install Java Development Kit (JDK) 1.4.2. The JDK is needed to compile maps and collaboration templates. A compatible JDK is provided with the WebSphere InterChange Server 4.3.0 product CD. Use the operating system's package installer to install JDK 1.4.2. If a version of JDK 1.4.2. was previously installed on the system, it may be used if it is the same or newer than the JDK that is bundled with WebSphere InterChange Server.

6. Install the Object Request Broker (ORB).

The ORB is automatically installed during the installation of WebSphere InterChange Server.

7. Install WebSphere MQ and apply the latest supported fix pack. To find the latest supported fix pack, visit:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

Follow the instructions in the WebSphere MQ documentation. If WebSphere MQ is upgraded, take the appropriate backup steps prior to installation.

8. Adjust operating system (OS) kernel settings

Create a new kernel or reboot the machine (or both) as documented in the *Installation Guide for UNIX* so that the new settings take effect. It is recommended to save the old kernel settings. Check the OS manuals for information about how to boot with the previous kernel and kernel settings if there is a problem with the new kernel.

9. Adjust settings for WebSphere MQ, database, and any third-party components involved to the recommended settings. For example, when using DB2 for the database, make sure that more than 50 connections are available by setting the maxappls parameter. In the migration scenarios tested in this redbook, 75 connections were used.

10. Install WebSphere InterChange Server using the installer. Consider the following notes during the install process:

- WebSphere InterChange Server must not be installed by root.
- The WebSphere InterChange Server name must not contain dots (.). For example, prd.1 is an invalid name, but prd\_1 is allowed.
- The WebSphere InterChange Server name must not change if WebSphere MQ is used for delivery transport and there are failed flows to be migrated. The queue definitions for WebSphere MQ delivery transport follow the *AP/ConnectorName/WebSphereInterChangeServerName* naming convention, which cannot be modified. As a result, the data in the previous queue definition is not accessible by the new WebSphere InterChange Server.

11. Adjust the ORB\_HOST and ORB\_PORT variables in \$CROSSWORLDS/CWSharedEnv.sh.

12. Install Adapter Framework 2.6 using the Adapter Framework installer.

To enable WebSphere Business Integration Adapters to run with WebSphere InterChange Server 4.3.0, Version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test

Connector. For more detailed instructions about installing adapters, refer to the individual adapter guides.

13. Install XML data handler using the XML data handler installer.

14. Install EmailAdapter using the EmailAdapter installer.

15. Update the icsadmin user profile.

If the shell used by the icsadmin is Korn Shell, edit the file named `.profile`. If another shell is used, refer to the respective `man` page for the instructions to modify the environment. Make sure that the path to Java Compiler (`javac`) comes after the path entry to `$CROSSWORLDS/bin`. Verify that the correct `javac` is used by using the `which` command. An example follows:

```
which javac
/usr/java14/bin/javac
```

16. Install tools on Windows 2000/2003 client.

Follow the instructions given in the *WebSphere InterChange Server 4.3 Installation Guide for UNIX* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

## Project environment setup

This chapter provides details about the environment that was set up and used to run and test 12 migration scenarios from previous versions of the WebSphere InterChange Server (WebSphere ICS) to Version 4.3.0.

The description of the hardware and software setup is followed by sections that provide details about the test cases and runtime data used for migration and verification.

## 7.1 Hardware and software environment setup

The following sections describe in detail the hardware and software test environments for AIX and Windows that were used when writing this book. These test environments represent hardware systems that have been found sufficient for the purposes of this redbook and meet the minimum recommended requirements. They do not represent an actual production environment. The software test environments were built on bare operating system installations to avoid issues or conflicts with other applications.

### 7.1.1 AIX environment

Table 7-1 shows the hardware equipment, machines called sthelens and persian, that was used for the AIX migration scenarios.

*Table 7-1 AIX hardware environment*

Hardware	sthelens	persian
Machine type	RS 6000 44P	RS 6000 44P
CPU	DUAL 375 MHz	DUAL 375 MHz
Memory	1024 MB	3072 MB
Hard disk	2 x 36, 4 GB	2 x 36, 4 GB

The two machines were installed with the software summarized in Table 7-2.

*Table 7-2 AIX software environment*

Software	sthelens	persian
Operating system	AIX 5.2 ML 2	AIX 5.2 ML 2
WebSphere MQ	5.3 FixPack 7	5.3 FixPack 7
Database	DB2 7.2 FixPack 6	DB2 8.1 FP 5
VisiBroker	Running on port 14000	Running on port 14000
WebSphere InterChange Server	Version 4.1.1.9	Version 4.2.1.7/4.2.2.4/ three instances of 4.3
IBM ORB (IBM JRE 1.3.1)	None	Running on port 15500
IBM ORB (IBM JRE 1.4.2)	None	Running on port 14500
C-compiler for stored procedures	GNU gcc version 3.3.2	GNU gcc version 3.3.2



Figure 7-1 shows an overview of the AIX environment setup.

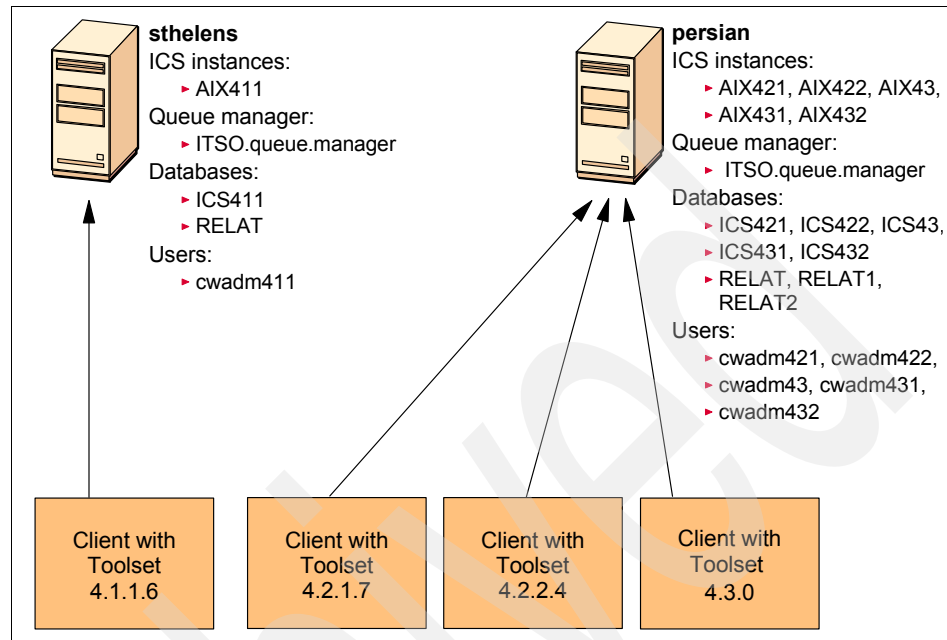


Figure 7-1 AIX environment setup

The main reason for only having one WebSphere InterChange Server installed on sthelens and five instances on persian is the restriction that IBM DB2 V7.2 and V8.1 cannot coexist on one machine.

All WebSphere InterChange Server instances were installed in separate folders in the /opt2/ directory.

WebSphere InterChange Server Version 4.3 instance AIX43 is used for Version 4.1.1 migration, AIX431 for Version 4.2.1 migration, and AIX432 for Version 4.2.2 migration.

Each version of WebSphere InterChange Server on persian is installed with separate users and is connected to separate databases. During in-place database migration, the Version 4.3 instances (AIX43, AIX431, and AIX432) were connected to the appropriate database of the previous versions. The RELATx databases were used for the relationships and the connection pool. Each database is cataloged using the loopback device to route using TCP/IP to spare shared memory segments on AIX. The cataloged database name has a preceding R. For example, the database ICS421 becomes RICS.

All WebSphere InterChange Servers were connected to the same queue manager called *ITSO.queue.manager* using different queue sets. During Version 4.1.1. in-place database migration, instance AIX43 is connected remotely to *ITSO.queue.manager* located on sthelens in order to be able to migrate failed flows using WebSphere MQ as transport.

## 7.1.2 Windows environment

Table 7-3 shows the hardware equipment used for the migration scenarios on Windows.

Table 7-3 Windows hardware environment

Hardware	Win411	Win421	Win422
Machine type	IBM ThinkCentre M50	IBM ThinkCentre M50	IBM ThinkCentre M50
CPU	2.4 GHz	2.4 GHz	2.4 GHz
Memory	2 GB	2 GB	2 GB
Hard disk	80 GB	80 GB	80 GB

Before migration, the three Windows servers were installed with the software described in Table 7-4.

Table 7-4 Windows software pre-migration environment

Software	Win411	Win421	Win422
Operating system	Microsoft Windows 2000 Professional Servicepack 4	Microsoft Windows 2000 Professional Servicepack 4	Microsoft Windows 2000 Professional Servicepack 4
WebSphere MQ	5.3 FixPack 3	5.3 FixPack 5	5.3 FixPack 5
Database	DB2 7.2	DB2 8.1 FixPack 2	DB2 8.1 FixPack 2
VisiBroker	Running on port 14000	Running on port 14000	None
WebSphere InterChange Server	Version 4.1.1.9	Version 4.2.1.7	Version 4.2.2.4
Toolset	Version 4.1.1.6	Version 4.2.1.7	Version 4.2.2.4
IBM ORB (IBM JRE 1.3.1)	None	None	Running on port 14500

Software	Win411	Win421	Win422
IBM ORB (IBM JRE 1.4.2)	None	None	None
C-compiler for stored procedures	Microsoft .NET SDK & Framework Version 1.1	Microsoft .NET SDK & Framework Version 1.1	Microsoft .NET SDK & Framework Version 1.1
WebSphere Application Server	None	Version 5.0.0	Version 5.0.2

After migration, the software environment is modified to the environment shown in Table 7-5.

*Table 7-5 Windows software post-migration environment*

Software	Win411	Win421	Win422
Operating system	Microsoft Windows 2000 Professional Servicepack 4	Microsoft Windows 2000 Professional Servicepack 4	Microsoft Windows 2000 Professional Servicepack 4
WebSphere MQ	5.3 FixPack 7	5.3 FixPack 7	5.3 FixPack 7
Database	DB2 8.1 FixPack 5	DB2 8.1 FixPack 5	DB2 8.1 FixPack 5
VisiBroker	None	None	None
WebSphere InterChange Server	Version 4.3	Version 4.3	Version 4.3
Toolset	Version 4.3	Version 4.3	Version 4.3
IBM ORB (IBM JRE 1.3.1)	None	None	None
IBM ORB (IBM JRE 1.4.2)	Running on port 14500	Running on port 14500	Running on port 14500
WebSphere Application Server	None	Version 5.0.2	Version 5.0.2

Figure 7-2 shows the Windows environment setup before and after the migration.

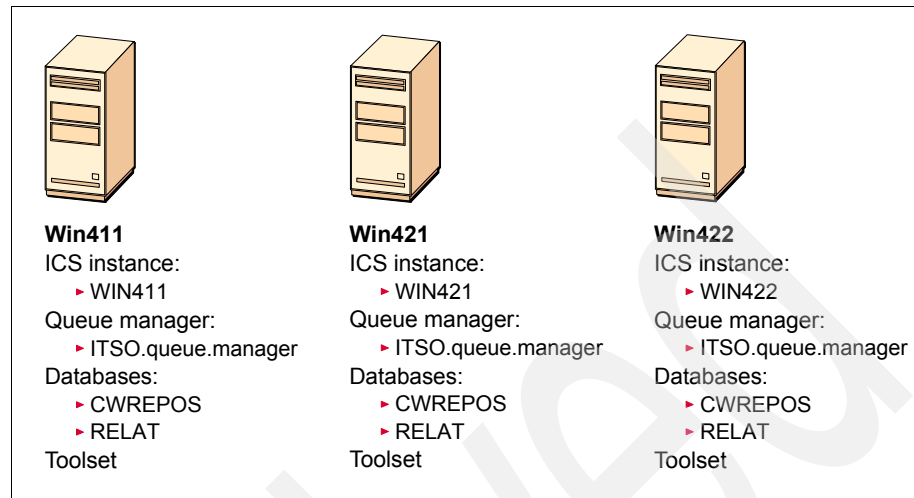


Figure 7-2 Windows environment setup

During migration the previous WebSphere InterChange Server is uninstalled and replaced by Version 4.3. The WebSphere InterChange Server names, queue manager names, and database names were not changed during migration. Because of this the content of Figure 7-2 is valid for the pre-migration and post-migration environment.

The RELAT databases were used for the relationships and the connection pool.

## 7.2 Migrated test cases

To test the success of the migration process, some common key features of the WebSphere InterChange Server were implemented in several test scenarios. The scenarios were simplified to represent only one specific feature to avoid side effects during the migration process. For the sake of consistency, the Visual Test Connector is used to represent the adapter agent in all scenarios.

The following sections describe the test cases we used in more detail.

### 7.2.1 Long Lived Business Process

The Long Lived Business Process scenario has been implemented in WebSphere InterChange Server Version 4.2.1 and 4.2.2 only because this feature is not available in Version 4.1.1. The collaboration gets a business object

from the source adapter, including an attribute used to store the transaction identifier. This business object is sent to the destination adapter, and the collaboration waits for a response at another port configured for asynchronous inbound calls. This scenario is used to test the Long Lived Business Process functionality after migration. In addition, flows were migrated that were in waiting status to check whether the response is accepted after migration. To test this, a time-out has been set to seven days for the asynchronous inbound port.

### **7.2.2 CustomerSync collaboration including relationships**

The customer sync collaboration represents the CustomerSync collaboration offered by IBM. In addition, a dynamic relationship and a static relationship are used in the map to:

- ▶ Maintain a relation between the two application key values (dynamic relationship).
- ▶ Look up the country code that is represented by different values in the two applications (static relationship).

The purpose of this test case is to verify the migration of a commonly used collaboration and static/dynamic relationships.

### **7.2.3 Connection pool**

The collaboration uses a connection pool to insert a row into a table in a DB2 database. The entry is based on the content of the business object for monitoring the events running through the collaboration. It is used to check the migration of connection pools and the API call using the connection pool.

### **7.2.4 Collaboration group**

The collaboration group consists of two simple passthrough collaborations. This group is used to check the migration of collaboration groups. The focus is to observe the behavior of the collaboration group after migration regarding failed flows and administrative commands (start, stop, pause, and change properties) compared to the previous version.

### **7.2.5 Polymorphic map**

A polymorphic map is used to determine which map is called based on the contents of the application-specific business object. This results in two different generic business objects calling different collaboration objects. The polymorphic map is used to check the polymorphic mechanism and the explicit binding of maps after migration.

## 7.2.6 Server Access Interface

The test case consists of two parts:

- ▶ Access Client implemented as custom code in a collaboration
- ▶ Access Client implemented in a standalone Java application

In both cases, the Access Client calls a simple passthrough collaboration. This scenario is used to verify the migration of the Server Access Interface API call. Of special interest were the code changes due to the change of the Object Request Broker (ORB) from VisiBroker ORB to IBM ORB.

## 7.2.7 Transactional collaboration

The scenario implements a typical transactional collaboration sending an event with verb *create* from a source adapter to two destination applications. If the second service call request fails for any reason, a delete is sent as compensation to the first destination application. An objective of this scenario is to check whether a transactional status of a collaboration can be resolved after migration.

## 7.2.8 Failed flows with different transports

The test case consists of three collaborations (one collaboration for each transport). The collaboration creates a failed flow based on the setting of a collaboration property. The source and destination adapters use three different transports (WebSphere MQ, IDL, and JMS). Failed flows were generated before migration and were resubmitted after migration using the Flow Manager.

## 7.2.9 Schedules

A schedule is used to stop a collaboration every two minutes. The objective of this test case is to verify that a schedule can be migrated.

## 7.2.10 Imported external code

The collaboration of this test case imports custom code from an external .jar file and calls a custom method to calculate the free memory within the JVM™. The resulting string of the method call is stored in an attribute of the destination business object. This scenario is used to test the migration of custom code used within a collaboration.

## 7.3 Migrated runtime data

The environment that must be migrated consists of both business logic (object and process definitions) and business content (runtime data). Therefore, the migration environment is prepared with runtime data, which includes:

- ▶ Incomplete flows
- ▶ Relationship table content
- ▶ Historical data for the Web-based System Monitor before migration

This data is described in the following sections.

### 7.3.1 Incomplete flows

Table 7-6 shows the prepared incomplete flows for each version before migration.

Table 7-6 Incomplete flows for migration

Scenario	Version (4.1.1/4.2.1/4.2.2)
Failed flow for IDL	All
Failed flow for MQ	All
Failed flow for JMS	All
Transactional collaboration in transactional status (compensation call failed)	All
Transactional collaboration failed flow (compensation call successful)	All
LLBP waiting asynchronous inbound call	4.2.1 / 4.2.2 only
LLBP failed flow	4.2.1 / 4.2.2 only
Collaboration group failed at Destination 1	All
Collaboration group failed at Destination 2	All

### 7.3.2 Relationship table content

The relationship tables contain sample entries for the dynamic and static relationships. These entries were migrated along with the relationship definitions in the in-place database migration and without in-place database migration scenarios. This data is used to determine whether relationship content can be migrated.

### 7.3.3 Historical data for the Web-based System Monitor

The following list describes the historical data that is collected with the Web-based System Monitor before migration:

- ▶ Business Object Probes
- ▶ A custom monitor
- ▶ Historical system information, including:
  - Flow information:
    - Successful flows
    - Failed flows
  - Component state changes:
    - WebSphere InterChange Server start
    - WebSphere InterChange Server stop
    - Connector Controller start
    - Connector Controller stop

This data is used to check if historical data of the Web-based System Monitor can be migrated.



# Migration procedures for WebSphere InterChange Server V4.1.1.9 to V4.3

This chapter provides instructions for migrating from an IBM WebSphere InterChange Server Version 4.1.1.9 (formally known as CrossWorlds) to an IBM WebSphere InterChange Server (WebSphere ICS) Version 4.3.0 environment.

This chapter provides four major sections for in-place database migration and without in-place database migration for both Microsoft Windows and IBM AIX operating system environments. For a detailed description of the issues related to in-place database migration and without in-place database migration, refer to Chapter 4, “General planning” on page 53.

Each major section is divided into subsections describing pre-migration procedures, migration procedures, and post-migration procedures.

## 8.1 In-place database migration of V4.1.1.9 to V4.3 on Windows

This section describes an in-place database migration scenario of WebSphere InterChange Server Version 4.1.1.9 to WebSphere InterChange Server Version 4.3.0. The complete repository from Version 4.1.1.9 is reused in this scenario. This includes component definitions, failed flows, and relationships.

For the full description of an in-place database migration, refer to Chapter 4, “General planning” on page 53 and the Installation Guide for Windows.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 8.1.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system.

#### **Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least Version 4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 development guide for each component. For example, business object names and attributes can only be used with US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67.

For more information, view the relevant publications at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

#### **Put the environment in a quiescent state**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.

- ▶ All data is consistent between the integrated applications.

To put WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator.
2. Resolve transactional status of transactional collaborations by right-clicking the collaboration object in the CrossWorlds System Manager (CSM). Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details. Transactional collaborations that are in transactional status can be migrated; however, it is recommended that they be resolved in this pre-migration step.
3. Resubmit failed events or discard the events by selecting **Administer Unresolved Flows** from the **Server** menu in the CSM. Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details. Failed flows can be migrated, but it is recommended that they be resolved in this pre-migration step.
4. After all events are processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in the CSM. Make sure the in-process and pending event counter is 0.

## Backing up and shutting down components

This section describes the steps needed to shut down and take a backup. It is a recommended practice to take a backup that can allow the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3.

Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Administration** → **PDFs** and click **System administration** in the right-side pane.

There are two types of data in the WebSphere InterChange Server system: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors

- Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- Relationship database tables
- Work-in-progress tables and transaction database tables
- WebSphere MQ message data

Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is WIN411 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository411.txt. This file is created in the directory where the **repos\_copy** command is run.

```
repos_copy -sWIN411 -uadmin -pnull -oRepository411.txt
```

**Note:** **repos\_copy** is not used during an in-place migration; however, it is still a recommended practice for backing up the current definitions of the WebSphere InterChange Server repository.

2. Shut down the InterChange Server. In the CrossWorlds System Manager, right-click the ICS name and select **Shutdown Gracefully**.
3. Back up the product directory indicated by the %CROSSWORLDS% environment variable. By default, this is C:\CrossWorlds.
4. Arrange with the database administrator to back up the database. This must be a complete backup, including schema information and stored procedures.
5. Shut down the (CW) VisiBroker Service.
6. Arrange for a WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.
7. Save the environment settings. In particular, the PATH, CLASSPATH, and CROSSWORLDS environment variables may be modified during the installation of WebSphere InterChange Server 4.3.0. The current environment settings can be viewed with the **set** command. The following example saves the environment settings to a file called env.txt:

```
set > env.txt
```

## 8.1.2 Migration steps

This section describes the main steps required for the migration, including:

- ▶ Upgrading related software to the required level
- ▶ Installing and configuring WebSphere InterChange Server 4.3
- ▶ Completing the upgrades of WebSphere InterChange Server components

### Preparing the database system

The database used for the repository in WebSphere InterChange Server 4.1.1.9 may not meet the minimum database level required by Version 4.3. If upgrading the database to the supported level is required, ensure afterward that it has the proper configuration for WebSphere InterChange Server 4.3.

- ▶ Check the *IBM WebSphere InterChange Server Installation Guide for Windows* for the proper database settings for V4.3. For example, when using DB2, a change in the database settings is that the `applheapsz` parameter must be set to at least 4096.

Read or download the *IBM WebSphere InterChange Server System Installation Guide for Windows* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **Windows** in the right-side pane.

- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To test that the C-compiler is installed and configured, use the `db2` command line processor to connect to any database and run the following command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been successfully created. After the stored procedure has been successfully created, it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 Fix Pack 7. If failed flows that involve WebSphere MQ Delivery Transport are migrated, the queue messages must be preserved when upgrading WebSphere MQ. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Uninstall software

This section describes the uninstall procedures for the WebSphere InterChange Server and related software.

1. Uninstall WebSphere InterChange Server and WebSphere Business Integration Adapters windows services (if any). Uninstall Windows services using the Add/Remove Programs control panel. Some of these services may still remain after the Add/Remove process.

Use the following command line to manually uninstall any remaining Windows Services.

```
cwservice -xr -sServiceName
```

2. Uninstall the VisiBroker Object Request Broker using the Add/Remove Programs control panel
3. Uninstall Java Development Kit (JDK) using the Add/Remove Programs control panel, only if it is not required by other products. For this release, the JDK is typically Java 2 SDK v.1.3.1\_02
4. Uninstall WebSphere InterChange Server. During the uninstall process, all files can be removed safely because of the product directory backup you made. After the uninstaller has completed, the product directory can be deleted.

## Installation

At this point, install WebSphere InterChange Server 4.3.0. Refer to Chapter 6, "Installation of WebSphere InterChange Server 4.3" on page 99 for instructions.

**Note:** During the installation process, use the same WebSphere InterChange Server name as in the previous version. Remember, the WebSphere InterChange Server name must not contain dots (.).

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is prompted to configure the WebSphere InterChange Server, as we demonstrate in the

following figures and examples. The WebSphere InterChange Server Configuration wizard can be run after the installation has finished, by invoking:

```
%CROSSWORLDS%\bin\ICSConfig.bat
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all configuration settings must be entered in the Options.txt file that is delivered with the installer. For detailed information, refer to the Installation Guide for Windows.

The following figures show the configuration values used by the redbook team during migration:

1. Figure 8-1 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

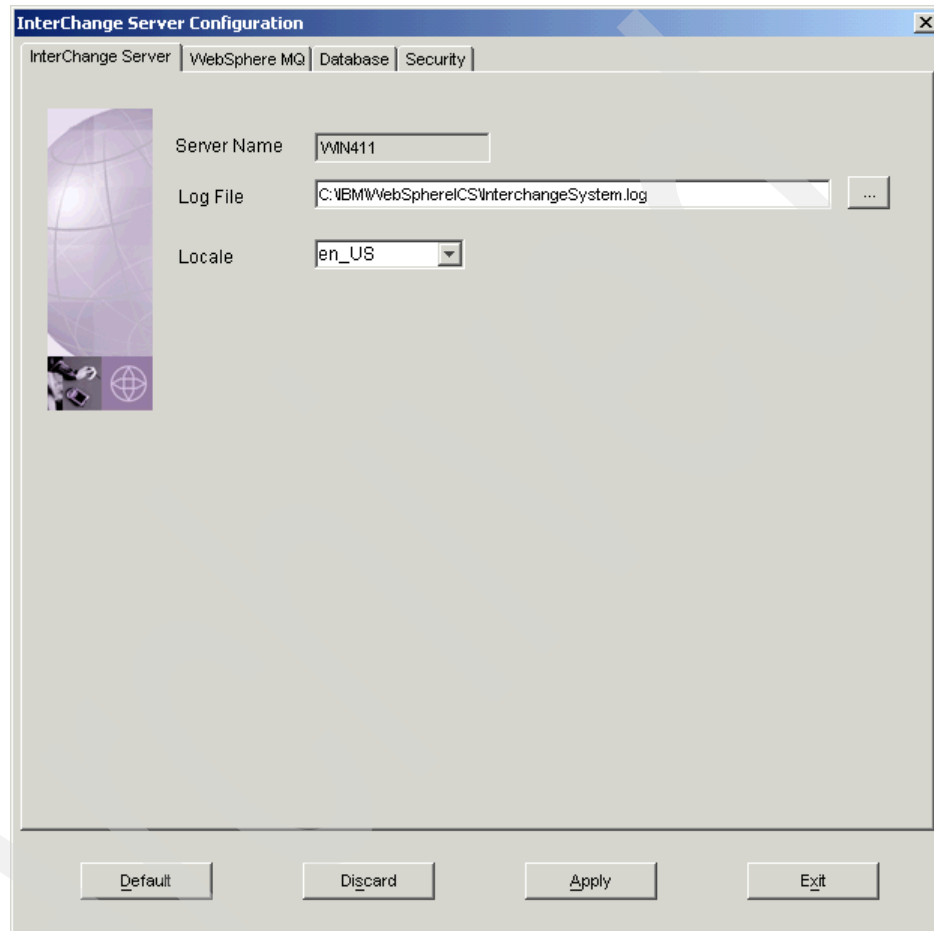
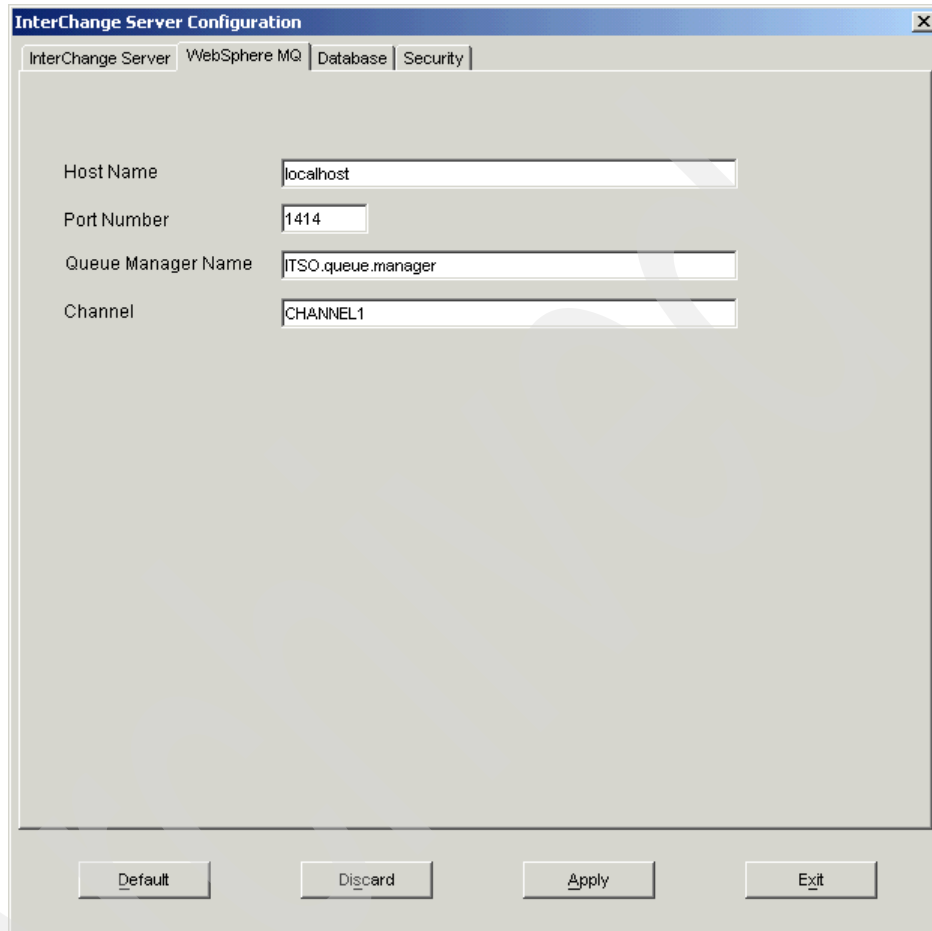


Figure 8-1 InterChange Server Configuration: InterChange Server tab



2. Figure 8-2 shows the WebSphere MQ configuration tab. Make sure the settings for WebSphere InterChange Server 4.1.1 are entered here.



The image shows a dialog box titled "InterChange Server Configuration". It has four tabs: "InterChange Server", "WebSphere MQ", "Database", and "Security". The "WebSphere MQ" tab is selected. The dialog contains four text input fields with the following labels and values:

Field Label	Value
Host Name	localhost
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

At the bottom of the dialog, there are four buttons: "Default", "Discard", "Apply", and "Exit".

Figure 8-2 InterChange Server Configuration: WebSphere MQ tab

3. The third tab is for database configuration. Enter the connection information from WebSphere InterChange Server 4.1.1 and apply the following settings:
  - Max Connections: unlimited, if possible. For the purposes of this book, the scenario was successfully migrated with this parameter set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Database' tab selected. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab contains several sections for configuring database connectivity and monitoring.

**Database Connectivity:**

- Database Driver: DB2 (dropdown)
- Max Connections: [ ] ☒ Unlimited
- Max Pools: 50 (text box)
- Idle Timeout: 2 (text box)

**Event Management:**

- Host Name: [ ]
- Database: cwrepos (text box)
- Max Connections: [ ] ☒ Unlimited
- Login: db2admin (text box)
- Password: [ ] (password field)
- Port Number: [ ]
- Change button

**Transactions:**

- Host Name: [ ]
- Database: cwrepos (text box)
- Max Connections: [ ] ☒ Unlimited
- Login: db2admin (text box)
- Password: [ ] (password field)
- Port Number: [ ]
- Change button

**Repository:**

- Host Name: [ ]
- Database: cwrepos (text box)
- Max Connections: [ ] ☒ Unlimited
- Login: db2admin (text box)
- Password: [ ] (password field)
- Port Number: [ ]
- Change button

**Flow Monitoring:**

- Host Name: [ ]
- Database: cwrepos (text box)
- Schema Name: [ ]
- Max Connections: [ ] ☒ Unlimited
- Login: db2admin (text box)
- Password: [ ] (password field)
- Port Number: [ ]
- Change button

At the bottom of the dialog are four buttons: Default, Discard, Apply, and Exit.

Figure 8-3 InterChange Server Configuration: Database tab

4. The fourth tab is for security configuration (). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Security' tab selected. The 'User Registry Type' is set to 'Repository'. The 'Repository' section contains the following fields: 'Host Name' (empty), 'Login' (clb2admin), 'Database' (cwrepos), 'Password' (masked with asterisks), 'Max Connections' (empty), 'Unlimited' (checked), and 'Port Number' (empty). A 'Change' button is next to the password field. At the bottom are buttons for 'Default', 'Discard', 'Apply', and 'Exit'.

User Registry Type	
User Registry	Repository

Repository	
Host Name	
Database	cwrepos
Max Connections	<input checked="" type="checkbox"/> Unlimited
Login	clb2admin
Password	*****
Port Number	

Figure 8-4 InterChange Server Configuration: Security tab

If the WebSphere InterChange Server is installed as a Windows Service, restart Windows as instructed.

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMS and collaboration directories from WebSphere InterChange Server Version 4.1.1 to the new installation directory for Version 4.3.0. Be sure to include all subdirectories.
2. Apply any custom configuration settings to %CROSSWORLDS%\InterchangeSystem.cfg.

For example, to disable DEADLOCK\_DETECTOR\_CHECK for collaborations, edit the config file code shown in Example 8-1 using a suitable editor for XML.

### Example 8-1 Disable DEADLOCK\_DETECTOR\_CHECK

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>>false</tns:reposController>
<tns:reposAgent>>false</tns:reposAgent>
<tns:localConfig>>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

In the line starting with <tns:value, change true to false.

3. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, they must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
4. Include external code and .jar files in CLASSPATH. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in %CROSSWORLDS%\bin\start\_server.bat. The external classes are first added to a new variable EXT\_LIB. Then the EXT\_LIB is appended to the CWCLASSES variable. Refer to the following example snippet from start\_server.bat:

```
REM external classes added here
set EXT_LIB=C:\Sample.jar
CWCLASSES="%CROSSWORLDS%\lib\crossworlds.jar;%CROSSWORLDS%\lib\cworio
n.jar;%CROSSWORLDS%\lib\CodeGeneration.jar;%EXT_LIB%
```

5. Modify and verify the environment variables as documented. Check that the following three environment variables are set.

**CROSSWORLDS** Contains the location of the 4.3.0 installation directory.

**MQ\_LIB** Contains the location of WebSphere MQ Java /lib directory.

**PATH** Contains the /bin directory of the IBM Java Development Kit 1.4.2. Ensure that the bin directory for JDK 1.4.2 precedes any other installed JDKs. Check that the compiler is installed by invoking `javac` from the command line. The usage instructions for `javac` are returned.

## Starting the new server

When the server is started for the first time after installation, WebSphere InterChange Server repository tables are upgraded or created. This section describes the steps to prepare and start the server for the first time.

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database
2. Run the `PeristentNameServer.bat` file, located in `%CROSSWORLDS%\bin\`.

**Restriction:** As we write this book, a product defect caused the WebSphere InterChange Server to freeze during first startup. This defect is applicable only for Version 4.1.1 when using DB2 as the repository database. This defect has been reported and assigned tracking number CR60364. Check the support site to see if a fix for this problem number is available.

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

In the scenario, a DDL script is used to circumvent the problem. This workaround manually creates five database tables and indices. Use the DB2 Command Center or command line processor to load the DDL script shown in Appendix B, “Sample code” on page 465. Change the database name, user, and password to match the environment. This workaround was reviewed and approved by the WebSphere InterChange Server Development team.

3. Start the WebSphere InterChange Server in design mode: Edit the Properties of the IBM WebSphere InterChange Server shortcut in the Start menu. Add the `-design` option to the Target field. Run the shortcut after modification.

WebSphere InterChange Server upgrades existing tables, creates new tables for new features, and logs messages to the `InterChangeSystem.log` file

located in the product directory. The message Interchange Server is Ready indicates that the WebSphere InterChange Server has started successfully.

## Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. To complete this task:

1. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Connect to the WebSphere InterChange Server.
  - c. Create a new integration component library and import all components from the server. During the import, maps and collaboration templates are upgraded to the compatible 4.3.0 format. Refer to the *Implementation Guide for WebSphere InterChange Server* for details about importing components using the Import Components wizard of System Manager:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Development** → **PDFs** and click **System implementation** in the right-side pane.

**Note:** If the repository has an unusually large number of components, it is recommended that you import the ICS components in pieces. Importing the entire repository at one time may be slow and might cause memory errors in the System Manager. The recommended order of component import is:

1. Business objects
2. Maps
3. Collaboration templates and objects
4. Connectors
5. Relationships and connection pools

6. Include any custom .jar files in the classpath for the WebSphere Business Integration Toolset. Select **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the classpath for the Toolset. Add the location of the custom .jar files to this list as shown in Figure 8-5.

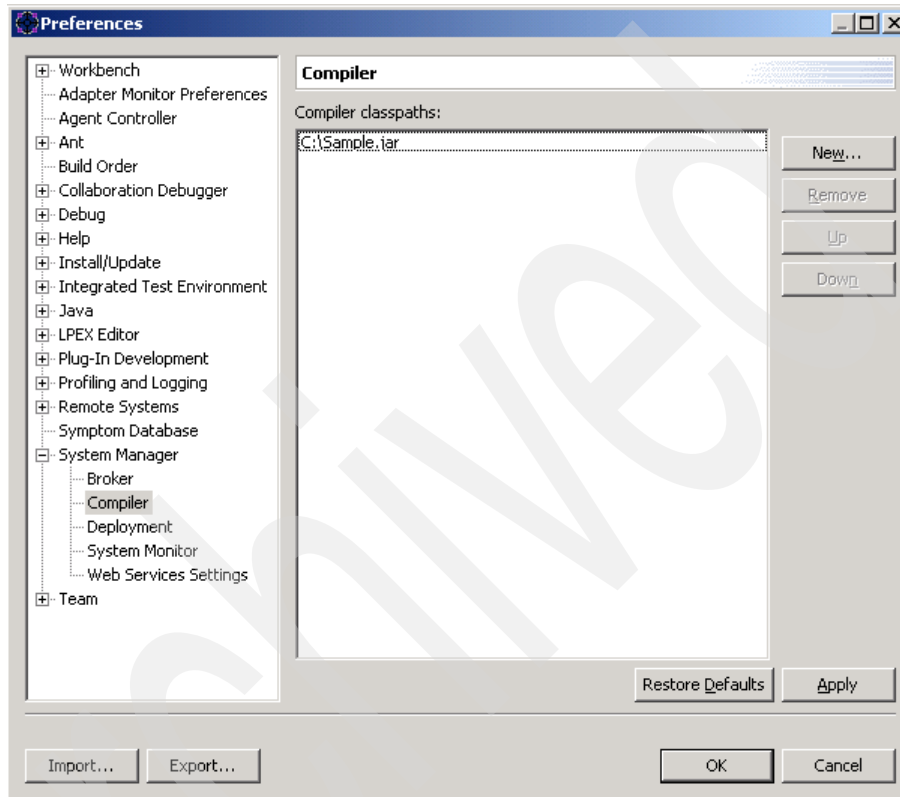


Figure 8-5 Setting the classpath for the Toolset

7. Set the compiler compliance to Java 1.4: Navigate to **Java** → **Compiler**. Under the Compliance and classfiles tab, change the compiler compliance level to **1.4** (Figure 8-6). A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to *not* rebuild files at this time.

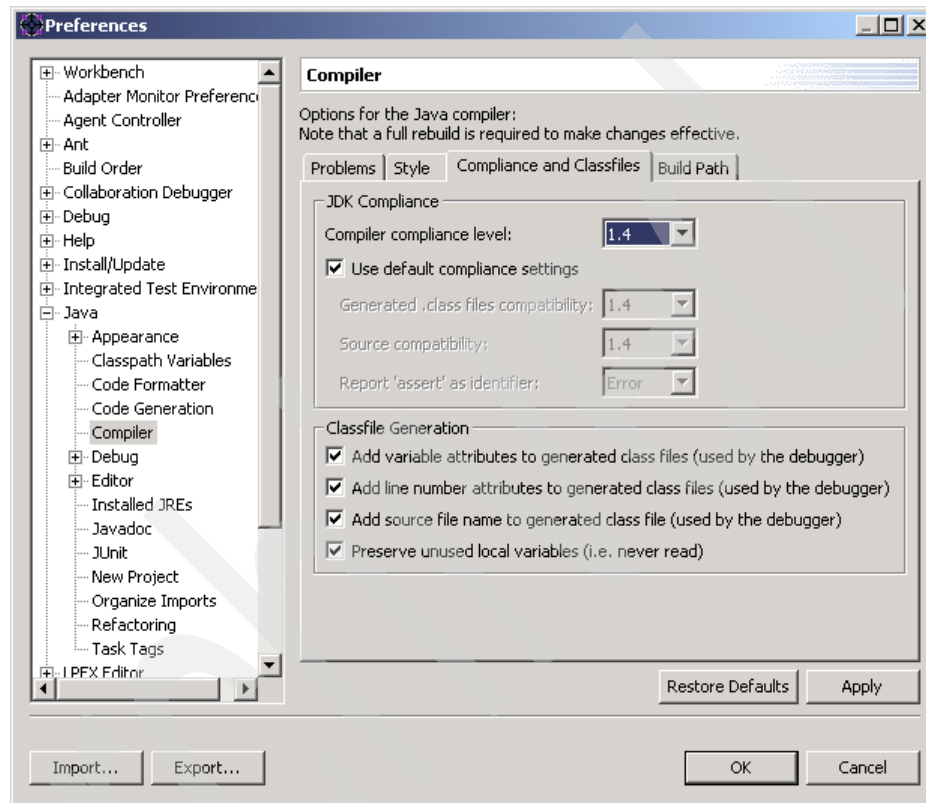


Figure 8-6 Setting the compiler compliance level

8. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is use of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in the previous step.
9. Compile all of the maps and templates in the integration component library and examine the console for any error messages.



10. The queue conventions for JMS delivery transport changed between V4.1.1 and V4.3.0. The configuration for all connectors that use JMS delivery transport must be examined and modified:
  - a. Edit the connector definition by double-clicking the connector name in the System Manager. This launches the Connector Configurator. The default queue names for JMS delivery transport, which are displayed in the Standard Properties tab, can be used as is or modified if necessary.
  - b. Create new queues in the queue manager with the appropriate queue names specified in the previous step.
  - c. In the Connector Configurator, save the configuration file to the project, and save a copy to file for use by the adapter agent.
11. Reconfigure any of the other components that require modification:
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more details.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
12. Create a new user project and populate it with the component from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for details.
13. Resolve any collaborations that are transactional status, because transactional status is lost after redeployment of the collaboration. This can be done from the Flow Manager. Refer to the *WebSphere InterChange Server System Administration Guide* for details.

If the collaboration cannot be brought to an active state to resolve the transactional status, deploy only the updated components that are required to activate the integration, then resolve the transactional status.

**Note:** To make use of all administrative features for maps and collaborations, the maps and templates must be deleted and redeployed. Typically, each component must be placed in an inactive state before it can be overwritten. To simplify the process and avoid any unresolved dependencies, the whole repository is deleted in step 14 and redeployed in step 15. Additionally, recompilation ensures that the code is optimized and Java 1.4 compliant.

However, redeploying the maps and templates is not required, because the Java Runtime Environment is backward-compatible and poses no problems executing map and template class files compiled in WebSphere InterChange Server 4.1.1.

If the decision is made to reuse the previous maps and collaborations, only deploy the components that have been updated.

14. In the System Manager, delete the repository by right-clicking the WebSphere InterChange Server name in the InterChange Server component management view and select **Delete the repository**.

15. Deploy the user project:

- a. Right-click the user project and select **Deploy User Project**.
- b. Click **Select the Destination Server** and choose the server name.
- c. Select the components to be deployed.
- d. Select **Perform Compilation on Interchange Server machine**.
- e. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This re-creates the stored procedures and does not overwrite existing data.

16. Restart WebSphere InterChange Server if prompted.

### 8.1.3 Post-migration steps

This section describes the steps that are taken after migration and after the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the password previously assigned to the administrator role. The following steps and figures show how to do this:

1. In the System Manager, connect to the WebSphere InterChange Server.
2. Right-click the server name in the InterChange Component management view and select **Users and roles** (Figure 8-7).

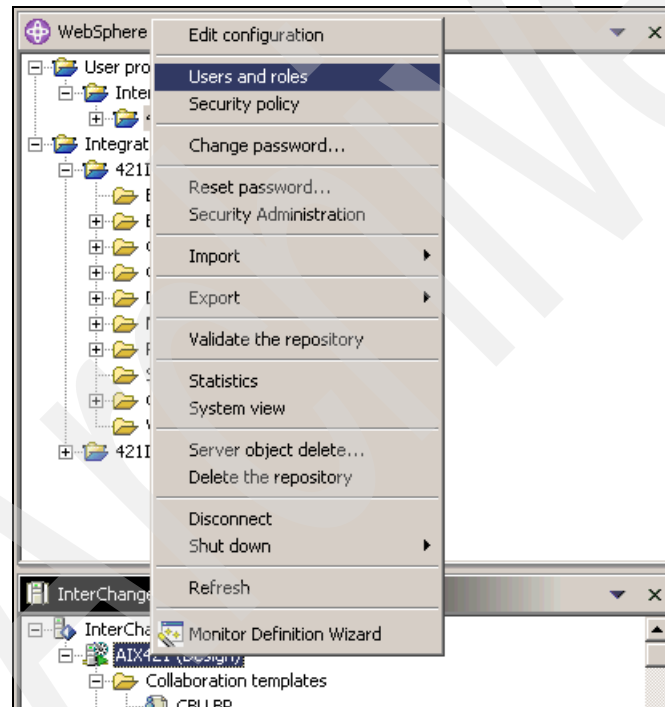


Figure 8-7 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User** as shown in Figure 8-8.

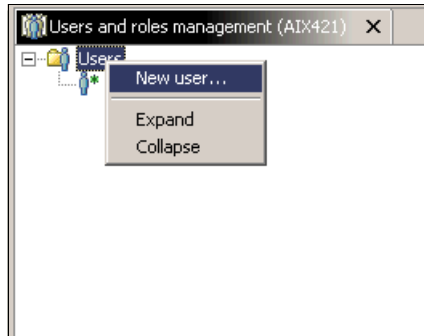


Figure 8-8 Selecting New user

4. In the Create user window (Figure 8-9), enter the name admin and the previously used password. Add additional information if needed and click **OK**.

A screenshot of a "Create user" dialog box. It has a title bar with a green icon and the text "Create user". The dialog contains several text input fields: "User name" with the text "admin", "Password" with "\*\*\*\*", "Confirm password" with "\*\*\*\*", "Distinguished name", "Common name", and "Surname". At the bottom right, there are two buttons: "OK" and "Cancel".

Figure 8-9 Creating a user

5. The new user is added to the User and Roles management window (Figure 8-10).

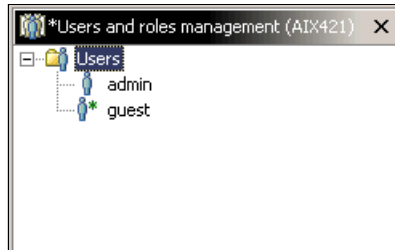


Figure 8-10 The newly created admin user

6. Right-click the new admin user and select **Add roles** (Figure 8-11).

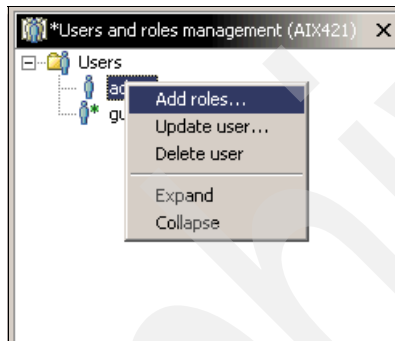


Figure 8-11 Add roles

7. The Role list window opens (Figure 8-12).

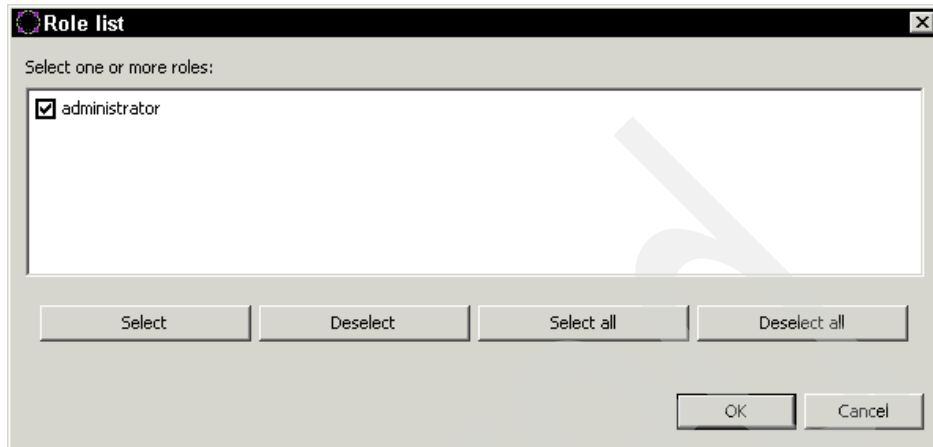


Figure 8-12 Role list

8. Select **administrator** and click **OK**. Close the Users and roles management window (Figure 8-13).

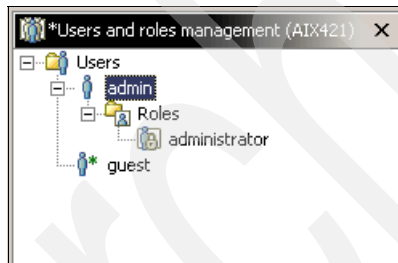


Figure 8-13 User admin with administrator role

9. Right-click the server again and choose **Edit Configuration**. This opens an editor for configuring the server.

10. Click the **Security - RBAC** tab (Figure 8-14) and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Set **Server start user** to admin.
  - c. For **Server start password** enter the password for the admin user.
  - d. Check all other settings in the tab and modify if needed.
  - e. Save the configuration to the server by clicking the save icon or selecting **File → Save**.

The screenshot shows the 'Security - RBAC' configuration tab in the WebSphere InterChange Server console. The interface is divided into several sections:

- RBAC setting:**
  - ☒ Enable RBAC
  - User registry: Repository (dropdown)
  - Server start user: admin (text field)
  - Server start password: \*\*\*\* (password field)
- Audit setting:**
  - ☒ Enable audit
  - Audit log directory: C:\IBM\WebSphereICS\logs\Audit (text field)
  - Audit log frequency: WEEKLY (dropdown)
  - Audit file size (MB): 1 (text field)
- LDAP setting:**
  - LDAP URL: (text field)
  - User name DN: (text field)
  - Password: (password field)
  - Userbase DN: (text field)
  - User name attribute: (text field)
  - Search criteria: (text field)
  - Maximum number of search returns: (text field)
  - ☐ SSL
- Repository details:**
  - Host name: (text field)
  - Database: cwrepos (text field)
  - Maximum number of connections: (text field) ☒ Unlimited
  - Max connect retries: 3 (text field)
  - Login: db2admin (text field)
  - Password: \*\*\*\*\* (password field) with a 'Change' button
  - Port number: 0 (text field)
  - Connect retry interval: 60 (text field) Second(s)

At the bottom, there is a navigation bar with tabs: General, Database, Tracing levels, Trace and log files, E-mail, CORBA, Miscellaneous, Environment properties, WebSphere MQ, Security - RBAC (selected), and Privacy.

Figure 8-14 Server Configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** Adapter Agents default to the guest account if RBAC is activated for WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, a WebSphere Business Integration Adapter Version 2.6 must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. It includes the Visual Test

Connector. For more detailed instructions about installing adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the migrated InterChange Server system from development to production it is recommended to perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 8-1 shows the scenarios that were tested in this migration and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be migrated successfully if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 8-1 Test scenarios results

Test scenario	Results
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated

A failed flow is considered to be migrated successfully if the event can be resubmitted and completed after migration. Table 8-2 on page 143 shows the results of the migration of failed flows.



Table 8-2 Migrated failed flows results

Failed flows tested	Results
Failed flow for IDL	Successfully migrated
Failed flow for MQ	Successfully migrated
Failed flow for JMS	Successfully migrated
Transactional collaboration in transactional status (compensation call failed)	Successfully migrated
Transactional collaboration failed flow (compensation call successful)	Successfully migrated
Collaboration group failed at Destination 1	Successfully migrated
Collaboration group failed at Destination 2	Successfully migrated

**Note:** After all system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is restarted in Production mode.

## 8.2 Without in-place database migration of V4.1.1.9 to V4.3 on Windows

This section describes a without in-place database migration scenario of WebSphere InterChange Server Version 4.1.1 to Version 4.3.0. WebSphere InterChange Server 4.3.0 is started with a new and empty repository. Repository content (component definitions) are exported from the original InterChange Server 4.1.1 and imported into WebSphere InterChange Server 4.3.

For the full description of an without in-place database migration refer to Chapter 4, “General planning” on page 53 and the Installation Guide for Windows.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 8.2.1 Pre-migration steps

The environment must be put into a quiescent state before migration is attempted. It is also necessary to take a backup of the system.

**Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

**Put the environment in a quiescent state**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator tool.
2. Resolve transactional status of transactional collaborations by right-clicking the collaboration object in the CrossWorlds System Manager (CSM). Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details. All status information is lost during the migration.
3. Resubmit failed events or discard the events by selecting **Administer Unresolved Flows** from the **Server** menu in CSM. Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details. Failed flows are lost during the migration.
4. When all events are processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics. Make sure the in-process and pending event counter is 0.

**Backing up and shutting down components**

This section describes the steps for shutting down and taking a backup. It is a recommended practice to take a backup that will enable the recovery of any data

that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3.0.

Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*. The WebSphere InterChange Server system has two types of data: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current WebSphere InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is WIN411 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository411.txt. This file is created in the directory where the **repos\_copy** command is run.

```
repos_copy -sWIN411 -uadmin -pnull -oRepository411.txt
```

2. Shut down the WebSphere InterChange Server. In the CrossWorlds System Manager, right-click the ICS name and select **Shutdown Gracefully**.
3. Back up the product directory indicated by the %CROSSWORLDS% environment variable. By default, this is C:\CrossWorlds
4. Arrange with the database administrator to back up the database. This must be a complete backup, including schema information and stored procedures.
5. Shut down the (CW) VisiBroker Service.
6. Arrange for a WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.

7. Clear WebSphere MQ queues of any outstanding messages if the same queue manager is reused for Version 4.3. Old messages in the queues may be seen as new events, which could cause data inconsistencies. Refer to the WebSphere MQ documentation for more information.
8. Save the environment settings. In particular, the PATH, CLASSPATH, and CROSSWORLDS environment variables may be modified during the installation of WebSphere InterChange Server 4.3. The current environment settings can be viewed with the **set** command. The following example saves the environment settings to a file called env.txt:

```
set > env.txt
```

## 8.2.2 Migration steps

This section describes the main steps required for the migration including:

- ▶ Upgrading related software to the required level.
- ▶ Installing and configuring WebSphere InterChange Server 4.3.
- ▶ Completing the upgrades of WebSphere InterChange Server components.

### Preparing the database system

Prepare an empty database for the 4.3.0 repository: Create a new database following the steps in the *IBM WebSphere InterChange Server System Installation Guide*, and ensure that the database has the proper configuration for WebSphere InterChange Server 4.3:

- ▶ Check the installation guide for the proper database settings for 4.3. For example, when using DB2, one change in the database settings is that the applheapsz parameter must be set to at least 4096.
- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be at Version 5.3.0.2 Fix Pack 7. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Uninstall software

This section describes the uninstall procedures for the WebSphere InterChange Server and related software:

1. Uninstall WebSphere InterChange Server and WebSphere Business Integration Adapters Windows Services (if any) using Add/Remove Programs in the Windows Control Panel.

Some of these services may remain after the Add/Remove process. Use this command line to manually uninstall any remaining Windows Services:

```
cwservice -xr -sServiceName
```

2. Uninstall the VisiBroker Object Request Broker using Add/Remove Programs in the Windows Control Panel.
3. Uninstall the Java Development Kit (JDK) using Add/Remove Programs in the Windows Control Panel—only if it is not required by other products. For this release, the JDK is typically Java 2 SDK 1.3.1\_02.
4. Uninstall WebSphere InterChange Server. During the uninstall process, all files can be removed safely because of the backup of the product directory that was taken earlier. After the uninstaller has completed, the product directory can be deleted.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 for details.

**Note:** During the installation process, the WebSphere InterChange Server name may be changed. If so, the adapters and queues may have to be configured. You should use the same WebSphere InterChange Server name if possible.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following steps explain how to configure the WebSphere InterChange Server. The IBM WebSphere

InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
%CROSSWORLDS%\bin\ICSConfig.bat
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all configuration settings must be entered in the file Options.txt, which is delivered with the installer. For detailed information, refer to the Installation Guide for Windows.

The following figures show the configuration values used by the redbook team during migration:

1. Figure 8-15 shows the InterChange Server tab. In this tab, the location and name of the log file are specified as well as the locale setting for WebSphere InterChange Server.

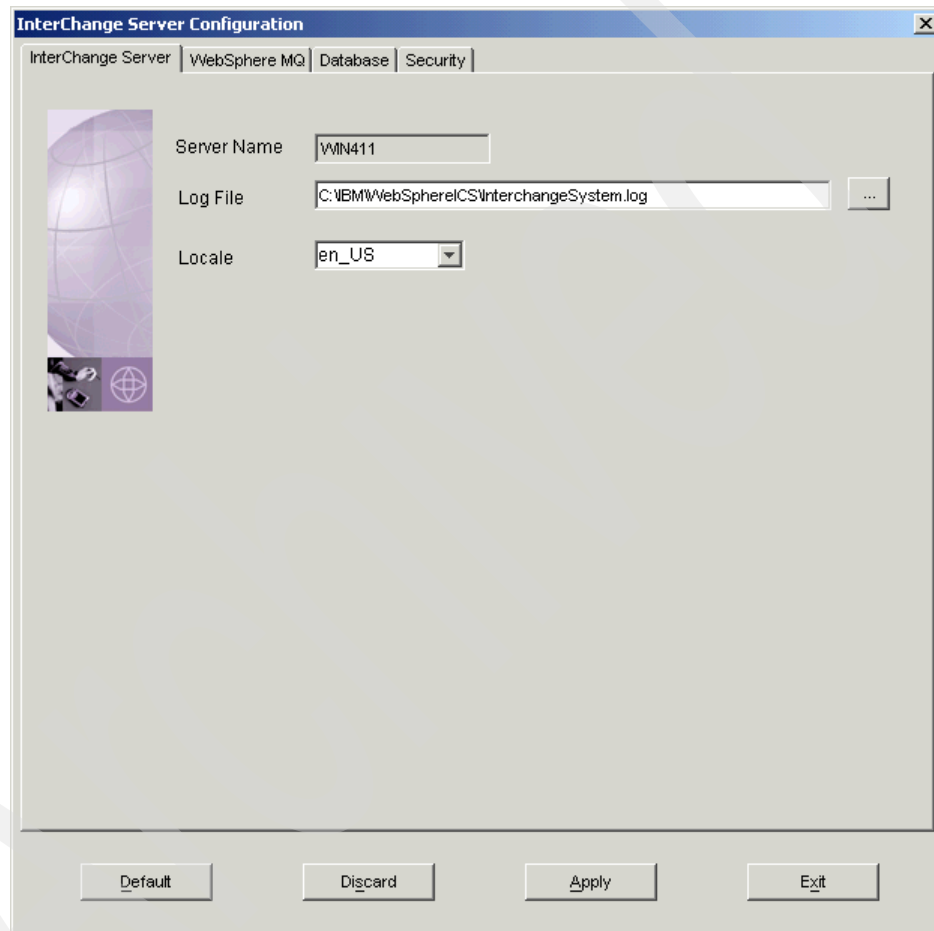
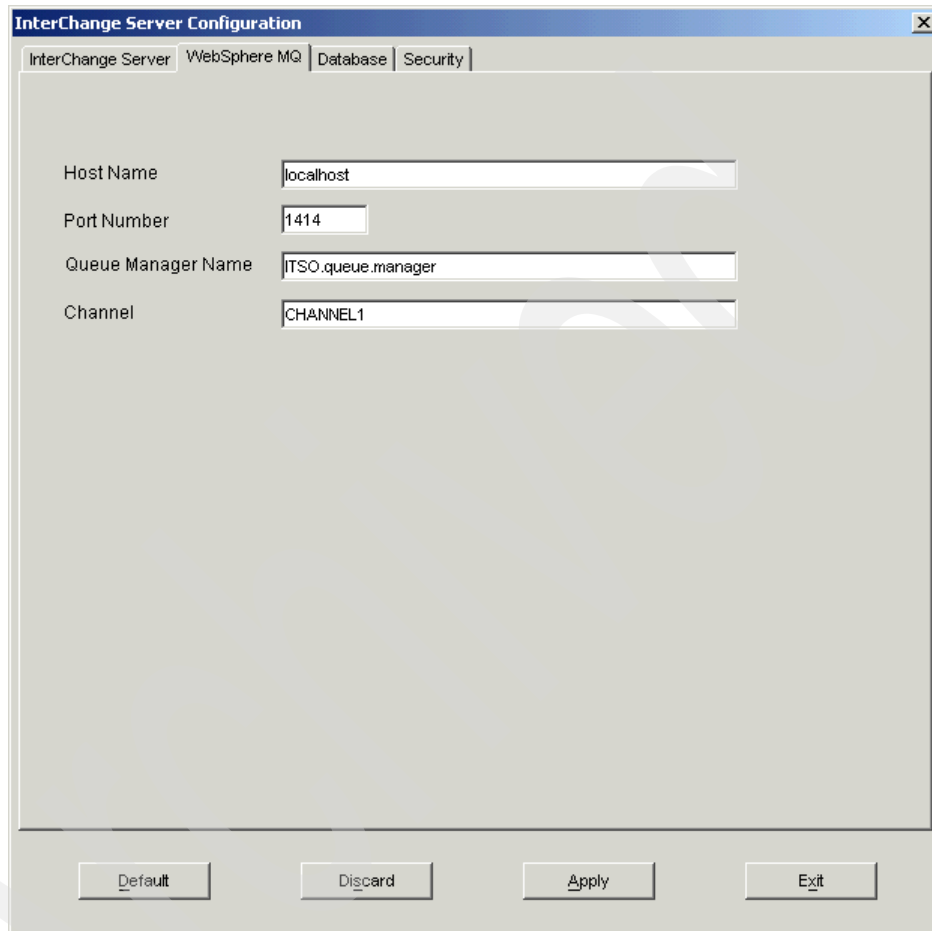


Figure 8-15 InterChange Server Configuration: InterChange Server tab

2. Figure 8-16 shows the WebSphere MQ configuration tab. Make sure the WebSphere MQ settings for WebSphere InterChange Server 4.1.1 are entered here.



The image shows a dialog box titled "InterChange Server Configuration". It has four tabs: "InterChange Server", "WebSphere MQ", "Database", and "Security". The "WebSphere MQ" tab is selected. Inside the dialog, there are four labeled text input fields: "Host Name" with the value "localhost", "Port Number" with the value "1414", "Queue Manager Name" with the value "ITSO.queue.manager", and "Channel" with the value "CHANNEL1". At the bottom of the dialog, there are four buttons: "Default", "Discard", "Apply", and "Exit".

Field	Value
Host Name	localhost
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 8-16 InterChange Server Configuration: WebSphere MQ tab



3. The third tab is for database configuration (Figure 8-17). Make sure to enter the connection information from WebSphere InterChange Server 4.1.1 and apply the following settings:
  - Max Connections: unlimited, if possible. For the purposes of this book, the scenario was successfully migrated with the Max Connections parameter set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Database' tab selected. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab contains several sections for configuring database connectivity and monitoring.

**Database Connectivity**

- Database Driver: DB2 (dropdown)
- Max Connections: ☐ Unlimited (checked)
- Max Pools: 50 (text box)
- Idle Timeout: 2 (text box)

**Event Management**

- Host Name: (text box)
- Database: cwrepos (text box)
- Max Connections: ☐ Unlimited (checked)
- Login: db2admin (text box)
- Password: \*\*\*\*\* (text box) with a 'Change' button
- Port Number: (text box)

**Transactions**

- Host Name: (text box)
- Database: cwrepos (text box)
- Max Connections: ☐ Unlimited (checked)
- Login: db2admin (text box)
- Password: \*\*\*\*\* (text box) with a 'Change' button
- Port Number: (text box)

**Repository**

- Host Name: (text box)
- Database: cwrepos (text box)
- Max Connections: ☐ Unlimited (checked)
- Login: db2admin (text box)
- Password: \*\*\*\*\* (text box) with a 'Change' button
- Port Number: (text box)

**Flow Monitoring**

- Host Name: (text box)
- Database: cwrepos (text box)
- Schema Name: (text box)
- Max Connections: ☐ Unlimited (checked)
- Login: db2admin (text box)
- Password: \*\*\*\*\* (text box) with a 'Change' button
- Port Number: (text box)

At the bottom of the dialog are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 8-17 InterChange Server Configuration: Database tab

4. The fourth tab is for security configuration (Figure 8-4 on page 129). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.

The screenshot shows the 'InterChange Server Configuration' window with the 'Security' tab selected. The 'User Registry Type' is set to 'Repository'. The 'Repository' section contains the following fields: 'Host Name' (empty), 'Login' (clb2admin), 'Database' (cwrepos), 'Password' (masked with asterisks), 'Max Connections' (empty), 'Unlimited' (checked), and 'Port Number' (empty). A 'Change' button is next to the password field. At the bottom are buttons for 'Default', 'Discard', 'Apply', and 'Exit'.

User Registry Type	
User Registry	Repository

Repository	
Host Name	
Login	clb2admin
Database	cwrepos
Password	*****
Max Connections	<input checked="" type="checkbox"/> Unlimited
Port Number	

Figure 8-18 InterChange Server Configuration: Security tab

If the WebSphere InterChange Server is installed as a Windows Service, restart Windows as instructed.

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMS and collaboration directories from WebSphere InterChange Server Version 4.1.1 to the new installation directory for Version 4.3.0. Be sure to include all subdirectories.
2. Apply any custom configuration settings to `$CROSSWORLDS/InterchangeSystem.cfg`.

For example, to disable `DEADLOCK_DETECTOR_CHECK` for collaborations, edit the config file code shown in Example 8-2 using a suitable editor for XML.

### *Example 8-2 Disable DEADLOCK\_DETECTOR\_CHECK*

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>>false</tns:reposController>
<tns:reposAgent>>false</tns:reposAgent>
<tns:localConfig>>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change `true` to `false` in the line starting with `<tns:value`.

3. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, these must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
4. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in `%CROSSWORLDS\bin\start_server.bat`. The external classes are first added to a new variable `EXT_LIB`. Then the `EXT_LIB` is appended to the `CWCLASSES` variable. Refer to the following example snippet from `start_server.bat`:

```
REM external classes added here
set EXT_LIB=C:\Sample.jar
CWCLASSES="%CROSSWORLDS%\lib\crossworlds.jar;%CROSSWORLDS%\lib\cworio
n.jar;%CROSSWORLDS%\lib\CodeGeneration.jar;%EXT_LIB%
```

5. Modify and verify the environment variables as documented. Confirm that the following three environment variables are set properly:

<b>CROSSWORLDS</b>	Contains the location of the 4.3.0 installation directory.
<b>MQ_LIB</b>	Contains the location of WebSphere MQ Java /lib directory.
<b>PATH</b>	Contains the /bin directory of IBM Java Development Kit 1.4.2. Ensure that this bin directory precedes any other installed JDKs. Confirm that the compiler is installed by invoking <b>javac</b> from the command line. The usage instructions for <b>javac</b> should be returned.

## Starting the new server

When the WebSphere InterChange Server is started for the first time after installation, the WebSphere InterChange Server repository tables are created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database
2. Run the PeristentNameServer.bat file located in %CROSSWORLDS%\bin\.
3. The V4.1.1 repository file may have to be edited in a text editor before importing into Version 4.3.0. Save a copy of the original V4.1.1 repository file before making these changes:
  - Connection pools: If a connection pool is not accessible by the 4.3.0 ICS, delete the connection pool entry in the repository file. The connection pool must be recreated at a later step. Example 8-3 shows a ConnectionPool definition.

### *Example 8-3 A ConnectionPool definition*

---

```
[ConnectionPool]
Compressed = TRUE
Checksum = 1222416404
[BeginBlob]
UESDBBQACAAIAKKCVDEAAAAAAAAAAAAAAAAAAAAUnVudG1tZUVudG10ew2QTQ7CIBCf9z1Fw15RV8Z
QTNPowmhiTHsABKwYmEkAf44vNrFR62YI32Pe8IYtH87mN+DQSjIdDwhuQaJykBbkKZej+ZkyTN2uEI
OTq9SjUaH746kNOH7CgG0jA1XQp71gCbW8QThZFr0NuooG2/5JZ0LdZwtDqttWTP65p0HCKd5EoVyBh
jtEduLE07oFTcxIKP9ldHBpN/f7Bft/nIpyA7BRPQkT+lTShF7LJNCE1tdJCC/k9Lf5bEsydQSwcIHZ
kAhr4AAABeAQAAUESBAhQAFAAIAAgAooJUMR2ZAB+AAAAXgEAAA0AAAAAAAAAAAAAAAAAAAAAFJ1bn
RpbWVfbnRpdHTQSwUGAAAAAAEAAQA7AAAA+QAAAAA
[EndBlob]
[End]
```

---

- Relationship definitions: For each relationship definition that is not accessible by the V4.3.0 WebSphere InterChange Server, delete the entire lines for DatabaseURL, LoginName, and LoginPassword. The new relationship tables are configured and created at a later step. Example 8-4 shows a section of the relationship definition after the modification.

*Example 8-4 A RelationshipDefinition after modification*

---

```
[RelationshipDefinition]
Name = Customer
Version = 1.0.0
State = Active
Identity = true
Static = false
Cached = false
DatabaseType = 3

[Role]
Name = CwCust
CwManagedIds = true
BusinessObject = Customer
[...]
```

---

- Any schedule information in the 4.1.1 repository file must be deleted. Example 8-5 shows a sample Schedule Entry.

*Example 8-5 A Schedule Entry*

---

```
[ScheduleEntry]
ID = 1
ComponentType = 1
ComponentName = ScheduleCollaboration_App1_to_App2
ComponentVersion = 1.0.0
Status = 1
Action = 1
InitialDate = Friday, October 8, 2004 8:26:48 PM CEST
NextOccurrence = Tuesday, October 12, 2004 10:26:48 PM CEST
RecurrenceType = 2
RecurrencePeriod = 1
RecurrenceNumWeeks = 58080488
UserComments =
[End]
```

---

4. Start the WebSphere InterChange Server in design mode by editing the **Properties** of the IBM WebSphere InterChange Server shortcut in the Start

menu. Add the option `-design` to the **Target** field. Run the shortcut after modification.

WebSphere InterChange Server logs messages to the `InterChangeSystem.log` file located in the product directory. An Interchange Server is Ready message indicates that WebSphere InterChange Server has started successfully.

## Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Load the modified repository file into the WebSphere InterChange Server by using the **repos\_copy** command line utility:

```
repos_copy -sWICS_NAME -ilocation of Repository411.txt
```

If the relationship definition has been edited in the previous step to change the database location, add the option `-r*` to the **repos\_copy** command line. This option instructs the WebSphere InterChange Server to *not* create the relationship tables and stored procedures at this time.

```
repos_copy -sWICS_NAME -r* -ilocation of Repository411.txt
```

### Notes:

- ▶ If the relationship definition has been edited to remove the database location, use the `-r*` option during **repos\_copy**. Otherwise, the WebSphere InterChange Server creates the relationship tables and stored procedures using the default location of the repository database.
- ▶ If the **repos\_copy** has failed, submit the command **repos\_copy -d -sWICS\_NAME** to delete the repository and restart the WebSphere InterChange Server before subsequent attempts to load the repository file.
- ▶ The **repos\_copy** utility does not support the replace (`-ar -arp`) options for duplicates for Version 4.1.1 repository files. If an error message is received indicating that there are duplicate items in the repository file, they must be removed manually.

2. Restart the WebSphere InterChange Server after a successful **repos\_copy**.
3. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Connect to the ICS.
  - c. Create a new integration component library and import the components from the server. During the import, maps and collaboration templates are

upgraded to the compatible 4.3.0 format. Refer to the *Implementation Guide for WebSphere InterChange Server* for details about importing components using the Import components wizard of System Manager.

**Note:** If the repository contains an unusually large number of components, it is recommended that you import the WebSphere InterChange Server components in pieces. Importing the entire repository at one time may be slow and might cause memory errors in the System Manager. The recommended order of component import is:

1. Business objects
2. Maps
3. Collaboration templates and objects
4. Connectors
5. Relationships and connection pools

6. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH for the Toolset. Add the location of the custom .jar files to this list.

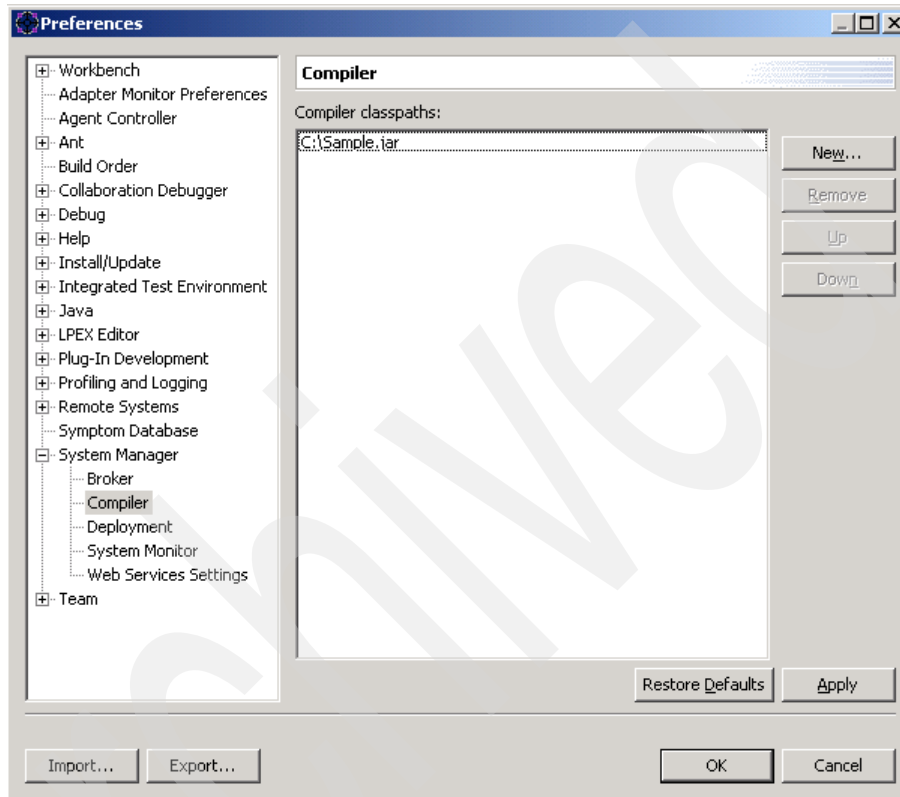


Figure 8-19 Setting the classpath for the Toolset



7. Set the compliance of the compiler to Java 1.4. Select **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag change the Compiler compliance level to **1.4**. A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to not rebuild files at this time.

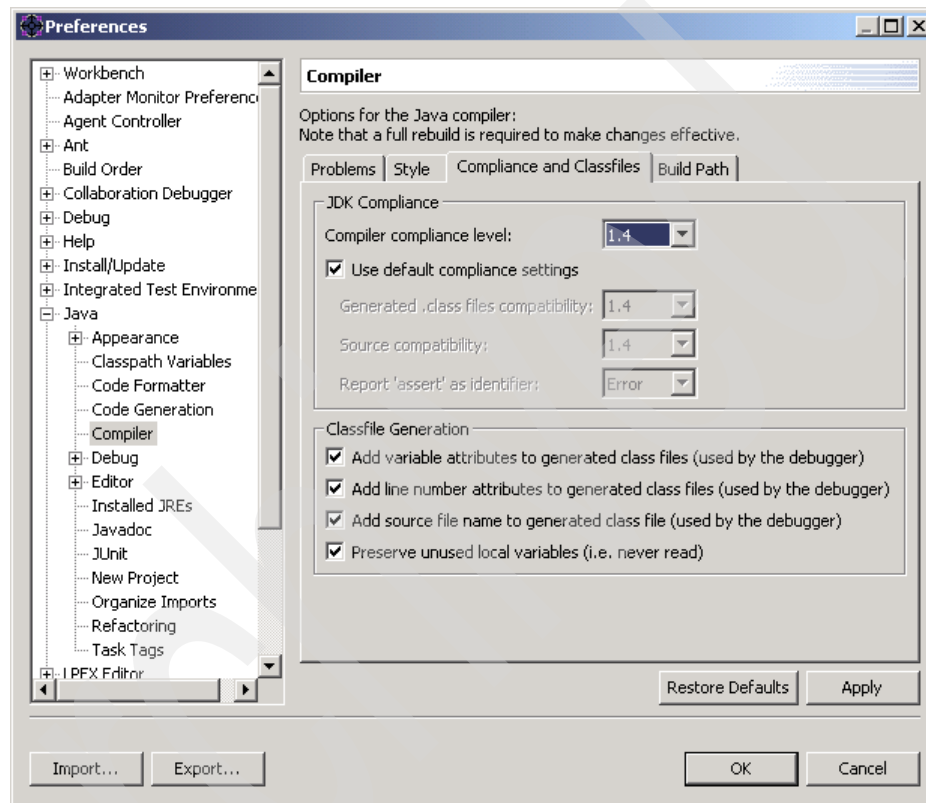


Figure 8-20 Setting the compiler compliance level

8. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the use of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in a previous step.
9. Compile all maps and templates in the integration component libraries and examine the console for any error messages.

10. The queue conventions for JMS delivery transport have changed between versions 4.1.1 and 4.3.0. The configuration for all connectors that use JMS delivery transport have been examined and modified.
  - a. Edit the connector definition by double-clicking the connector name in the System Manager. This launches the Connector Configurator. The default queue names for JMS delivery transport are displayed in the Standard Properties tab. The default queue names can be used as is, or they can be modified if needed.
  - b. Create new queues in the queue manager with the appropriate queue names specified in the previous step.
  - c. In the Connector Configurator, save the configuration file to the project, and save a copy to file for use by the adapter agent.
11. Reconfigure any of the other components that require modification:
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, "Technical impact of major design changes" on page 67 for more details.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
12. Create a new user project and populate it with the component from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for details.
13. In the System Manager, delete the repository by right-clicking the WebSphere InterChange Server name in the InterChange Server component management view and select **Delete the repository**.
14. Deploy the user project:
  - a. Right-click the user project and select **Deploy User Project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. If the location of the relationship tables has changed, select **Create Schema**.
  - f. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This recreates the stored procedures and does not overwrite existing data.

15. Restart the WebSphere InterChange Server if prompted.

16. Completing relationship migration:

If a new relationship location is used, the relationship data from the Version 4.1.1 relationship tables can be migrated in the following way.

- a. For both static and dynamic relationships, copy the data from each relationship table to the new relationship table. Consult with a database administrator for more details. The steps to complete this process using DB2 are:
  - i. In the Control Center, right-click the original relationship table and choose **Export** as shown in Figure 8-21.

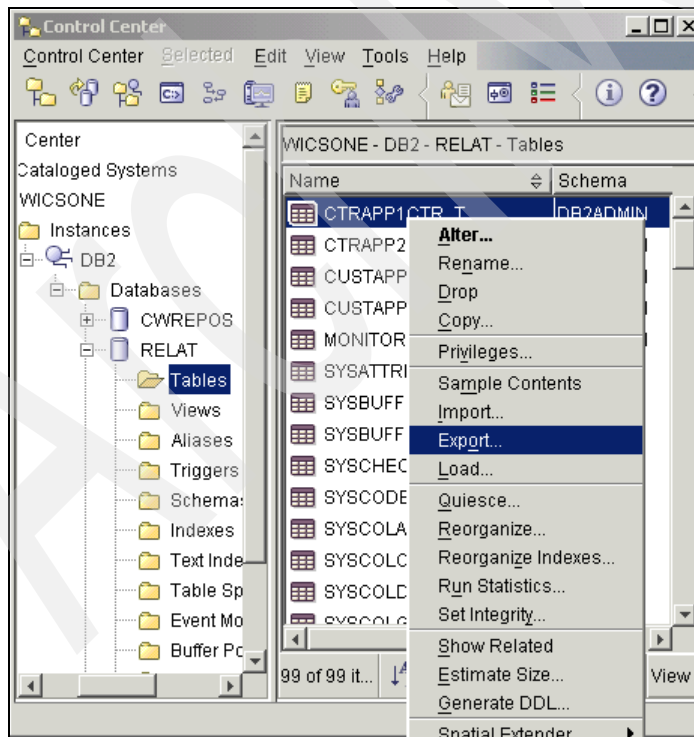


Figure 8-21 Exporting a table

- ii. Enter an output file location and a message file location and click **OK**. This is shown in Figure 8-22.

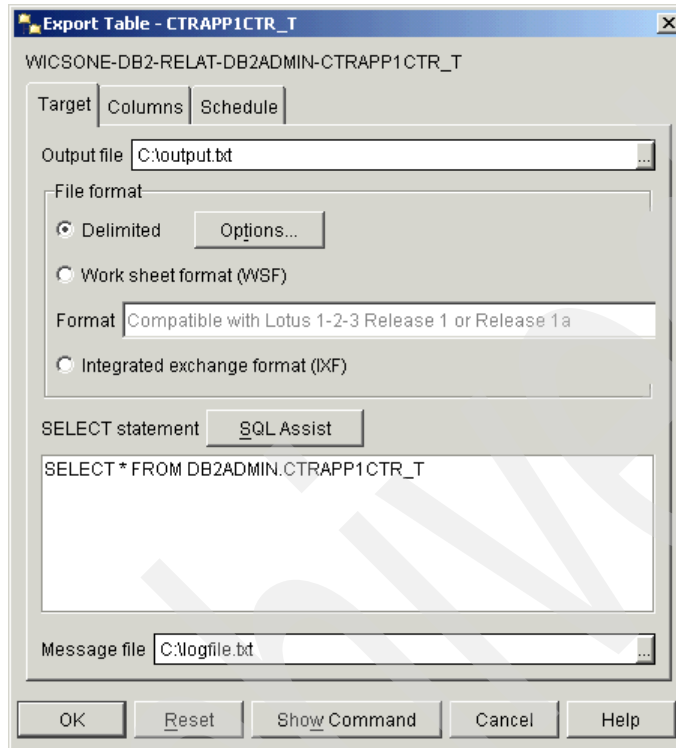


Figure 8-22 Completing the table export

**Tip:** If the OK button is not active, in the Schedule tab, select the option **Run now without saving task history**.

- iii. In the Control Center, right-click the newly created relationship table and choose **Import**.

- iv. Enter the location of the saved file as the import file and click **OK** as shown in Figure 8-23.

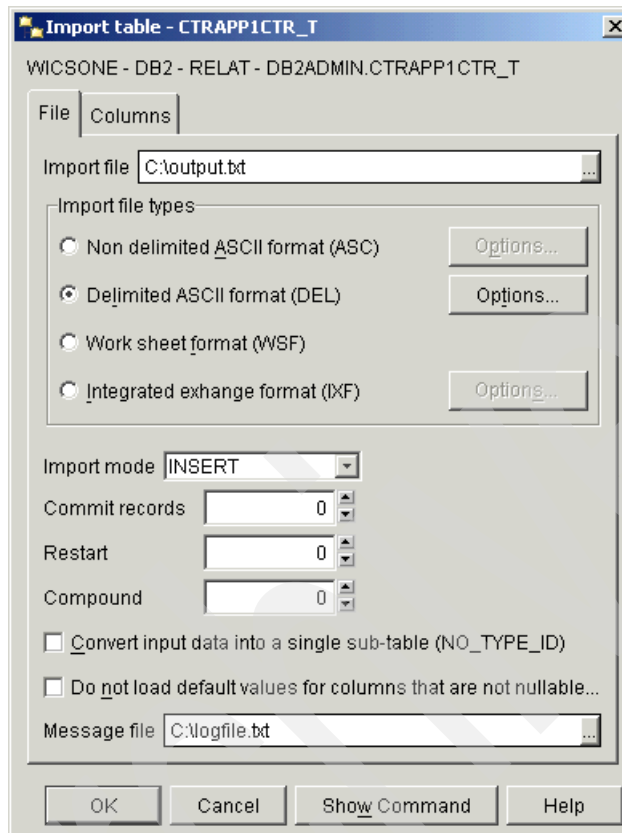


Figure 8-23 Importing the data

- b. Additional steps are needed to set the sequence number for each relationship. Here are the steps required to set the sequence number when the relationships are stored in DB2.
- Find the current highest ID for any relationship by examining the INSTANCEID column for any participant table in the relationship.  

```
SELECT MAX(INSTANCEID) FROM DB2ADMIN.CUSTAPP1CUST_T
```

In this example, the query has returned the value 41 as the highest INSTANCEID.
  - Find the sequence variable name. The following command generates a SQL script that provides information about the relationship table. This

information can also be obtained in the Control Center by right-clicking a database table and choosing **Generate DDL**.

```
db2look -dRELAT -tCUSTAPP1CUST_T -a -e -c -oOUTPUT.SQL
```

Examine the OUTPUT.SQL file to obtain the sequence variable name. Search for the text CREATE SEQUENCE.

```
CREATE SEQUENCE "DB2ADMIN"."CUSTOMER_1_0_0" AS INTEGER MINVALUE 1  
MAXVALUE 2147483647 START WITH 1 INCREMENT BY 1 CACHE 20 NO CYCLE  
NO ORDER;
```

In the previous example, the sequence variable name is Customer\_1\_0\_0. The sequence name is derived from the name of the relationship. In this case the relationship name is Customer.

- iii. Set the sequence variable to be at least one larger than the current number. In this example, the current highest INSTANCEID is 41 so the sequence is set to 42 with this command:

```
alter sequence Customer_1_0_0 restart with 42
```

- c. Verify that the relationship has been successfully moved to the new database by examining the relationship using Relationship Manager.

### 8.2.3 Post-migration steps

This section describes the steps to be taken after migration is complete and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the password previously assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.

2. Right-click the server name in the InterChange component management view and select **Users and Roles**. This is shown in Figure 8-24.

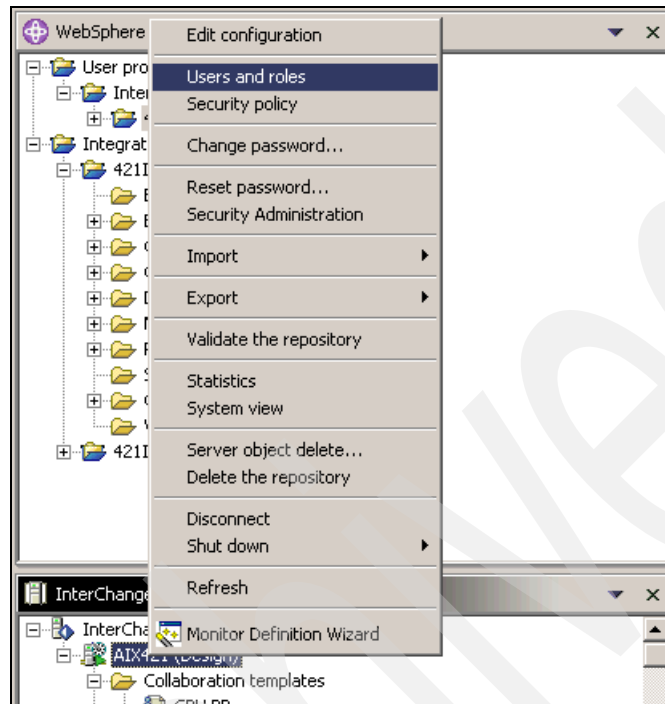


Figure 8-24 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User**. This is shown in Figure 8-25.

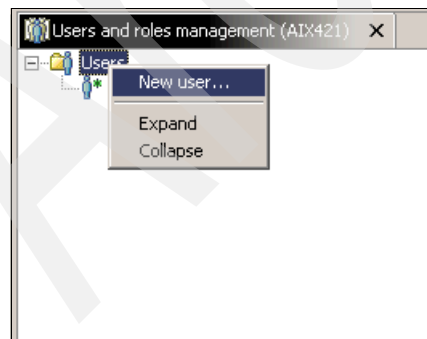


Figure 8-25 Selecting New user

4. The Create User window opens (Figure 8-26). Enter the name `admin` and the previous password, plus additional information if needed. Click **OK** when finished.

A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' (containing 'admin'), 'Password' (containing '\*\*\*\*'), 'Confirm password' (containing '\*\*\*\*'), 'Distinguished name' (empty), 'Common name' (empty), and 'Surname' (empty). At the bottom right are 'OK' and 'Cancel' buttons.

Create user

User name  
admin

Password  
\*\*\*\*

Confirm password  
\*\*\*\*

Distinguished name

Common name

Surname

OK Cancel

Figure 8-26 Creating a user

5. The new user is added to the Users and roles management window.

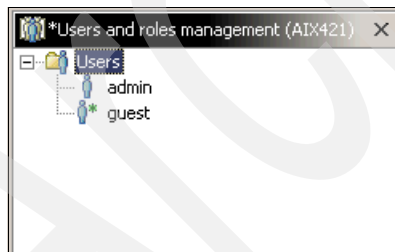


Figure 8-27 The newly created admin user



6. Right-click the newly created admin user and select **Add roles**. This is shown in Figure 8-28.

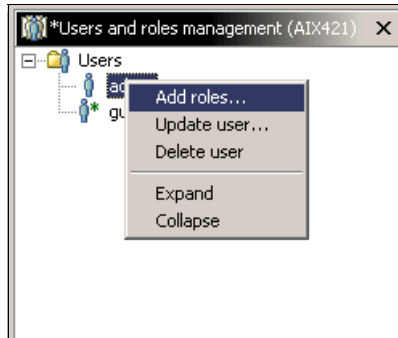


Figure 8-28 Add roles

7. The Role list window opens. Select **administrator** and click **OK**.

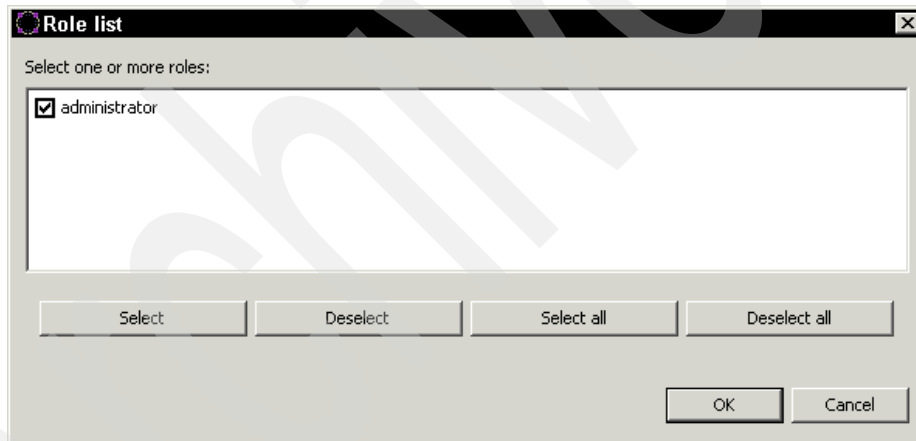


Figure 8-29 Role list

8. Close the Users and roles management window (Figure 8-30).



Figure 8-30 User admin with administrator role

9. Right-click the server again in the Components management view and choose **Edit Configuration**. This opens an editor for configuring the server.

10. Click the **Security-RBAC** tab as shown in Figure 8-31 and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Enter admin for **Server start user**.
  - c. Set **Server start password** to the password for the admin user.
  - d. Check all other settings in the tab and modify if needed.
  - e. Save the configuration to the server by clicking the save icon or selecting **File → Save**.

The screenshot shows the 'Security-RBAC' configuration tab in the WebSphere InterChange Server administration console. The window is titled '\*IC5430'. It contains several sections with settings:

- RBAC setting:**
  - ☒ Enable RBAC
  - User registry: Repository (dropdown)
  - Server start user: admin
  - Server start password: \*\*\*\*
- Audit setting:**
  - ☒ Enable audit
  - Audit log directory: C:\IBM\WebSphereICS\logs\Audit
  - Audit log frequency: WEEKLY (dropdown)
  - Audit file size (MB): 1
- LDAP setting:**
  - LDAP URL: (empty)
  - User name DN: (empty)
  - Password: (empty)
  - Userbase DN: (empty)
  - User name attribute: (empty)
  - Search criteria: (empty)
  - Maximum number of search returns: (empty)
  - ☐ SSL
- Repository details:**
  - Host name: (empty)
  - Database: cwrepos
  - Maximum number of connections: (empty) ☒ Unlimited
  - Max connect retries: 3
  - Login: db2admin
  - Password: \*\*\*\*\* Change
  - Port number: 0
  - Connect retry interval: 60 Second(s)

At the bottom, there is a tab bar with the following tabs: General, Database, Tracing levels, Trace and log files, E-mail, CORBA, Miscellaneous, Environment properties, WebSphere MQ, **Security - RBAC**, and Privacy.

Figure 8-31 Server configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** Adapter Agents default to the Guest account if RBAC is activated for the WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual

Test Connector. For more detailed instructions on how to install adapters, refer to the individual adapter guides.

## Server Access Interface

If any Access Clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the upgraded InterChange Server system from development to production, it is recommended perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information on testing a migrated system.

Table 8-3 shows the scenarios that were tested in this migration, and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 8-3 Test scenario results

Test scenario	Results
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated

**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is re-started in Production mode.

## 8.3 In-place database migration of V4.1.1.9 to V4.3 on AIX

This section describes an in-place database migration scenario of WebSphere InterChange Server Version 4.1.1.9 to WebSphere InterChange Server Version 4.3.0. The complete repository from 4.1.1.9 is reused in this scenario. This includes component definitions, failed flows, and relationships.

For the full description of an in-place database migration, refer to Chapter 4, “General planning” on page 53 and the Installation Guide for UNIX.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 8.3.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system.

#### **Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

#### **Put the environment in a quiescent state**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.

- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server in a quiescent state, perform the following steps:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator.
2. Resolve transactional status of transactional collaborations by right-clicking the collaboration object in the CrossWorlds System Manager (CSM). Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details (page 121). Transactional collaborations that are in transactional status can be migrated, however it is recommended that they are resolved in this pre-migration step.
3. Resubmit failed events or discard the events by selecting Administer Unresolved Flows from the Server menu in CSM. Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details (page 122). Failed flows can be migrated, however it is recommended that they are resolved in this pre-migration step.
4. Once all events are processed by the system, stop all collaborations. To check if all events are processed, check the collaboration statistics in the CSM. Make sure the in-process and pending event counter is 0.

## Backing up and shutting down components

This section describes the steps needed to shutdown and take a backup. It is a recommended practice to take a backup which allows the recovery of any data that might be inadvertently overwritten during the installation of ICS 4.3.0.

Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*. There are two types of data in the WebSphere InterChange Server system: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current WebSphere InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is WIN411 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository411.txt. This file is created in the directory where the **repos\_copy** command is run.

```
repos_copy -sWIN411 -uadmin -pnull -oRepository411.txt
```

**Note:** A **repos\_copy** is not used during an in-place migration; however, it is still a recommended practice in order to back up the current definitions of the WebSphere InterChange Server repository.

2. Shut down the WebSphere InterChange Server:

```
ics_manager -stop
```

3. Arrange with the database administrator to back up the database. This must be a complete backup, including schema information, and stored procedures.
4. Stop the Inprise Visibroker Agent. Run the following command to determine the process ID of the osagent:

```
ps -ef | grep osagent
```

This is a sample output of the command:

```
cwadm421 35012      1   0   Aug 09      -   1:18 osagent
```

To stop the osagent run the command:

```
kill -9 35012
```

(35012 is the process ID of the osagent process.)

5. Arrange with a WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.

## 8.3.2 Migration steps

This section describes the main steps required for the migration including:

- ▶ Upgrading related software to the required level
- ▶ Installing and configuring WebSphere InterChange Server 4.3
- ▶ Completing the upgrades of WebSphere InterChange Server components

### Preparing the database system

The database used for the repository in WebSphere InterChange Server 4.1.1.9 may not meet the minimum required database level required by Version 4.3, so it must be upgraded to the supported level. After upgrading the database, ensure that the database has the proper configuration for WebSphere InterChange Server 4.3.

- ▶ Check the *IBM WebSphere InterChange Server Installation Guide for UNIX* for the proper database settings for V4.3. For example, when using DB2, a change in the database settings is that the `applsheap` parameter must be set to at least 4096.

Read or download the *IBM WebSphere InterChange Server System Installation Guide for UNIX* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **UNIX** in the right-side pane.

- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

### Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 fix pack 7. If failed flows that involve WebSphere MQ Delivery Transport are being migrated, the queue messages



must be preserved when upgrading WebSphere MQ. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 for further instructions.

It is recommended that WebSphere InterChange Server Version 4.3.0 is installed with a different user ID and to a different directory than the previous version. The previous installation is not modified during the install process.

During the install process, use the same WebSphere InterChange Server name as in the previous version. As a reminder, the WebSphere InterChange Server name must not contain dots (.).

The WebSphere InterChange Server name can be changed after the installation process. Edit the file `$CROSSWORLDS/bin/ics_manager` and the file `$ADAPTER_RUNTIME/bin/connector_manager`. Change the parameter `SERVERNAME` to the desired ICS name.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following pictures and examples explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
$CROSSWORLDS/bin/ICSConfig.sh
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all the configuration settings have to be entered in the file `Options.txt` which is delivered with the installer. For detailed information, refer to the *IBM WebSphere InterChange Server System Installation Guide for UNIX*.

The following figures show the configuration values used by the redbook team during migration:

1. Figure 8-32 shows the InterChange Server tab. In this tab, the location and name of the log file are specified as well as the locale setting for WebSphere InterChange Server.

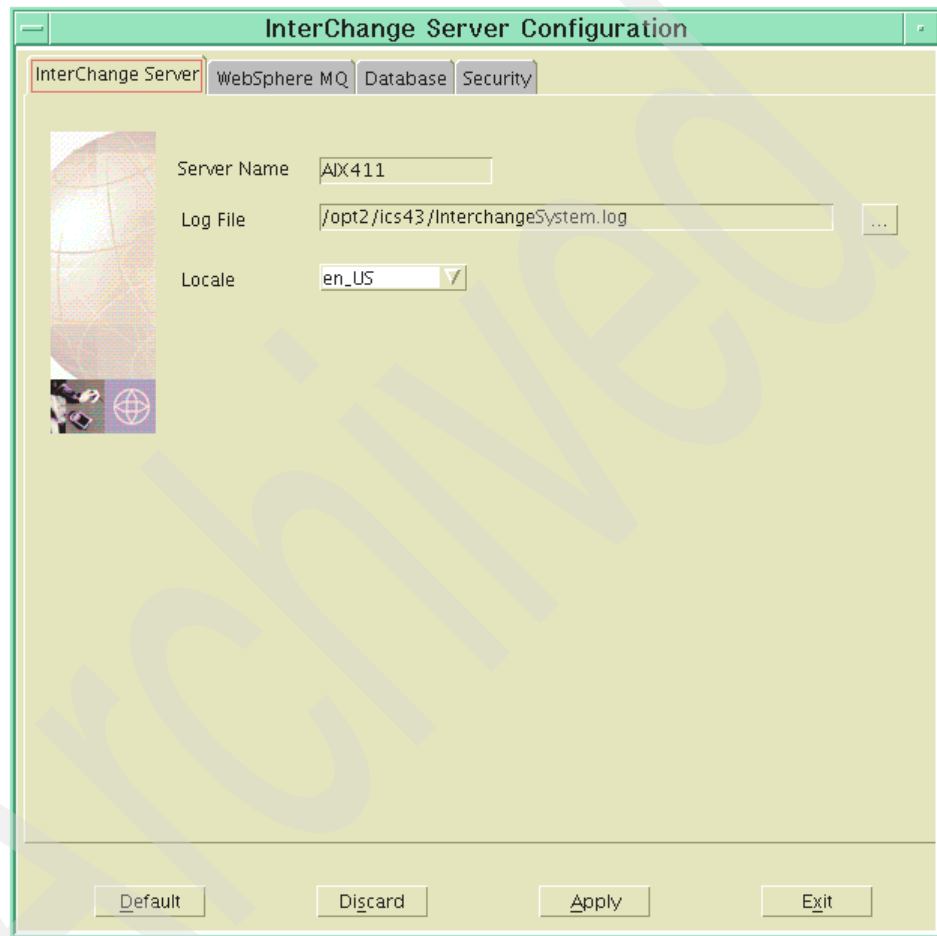
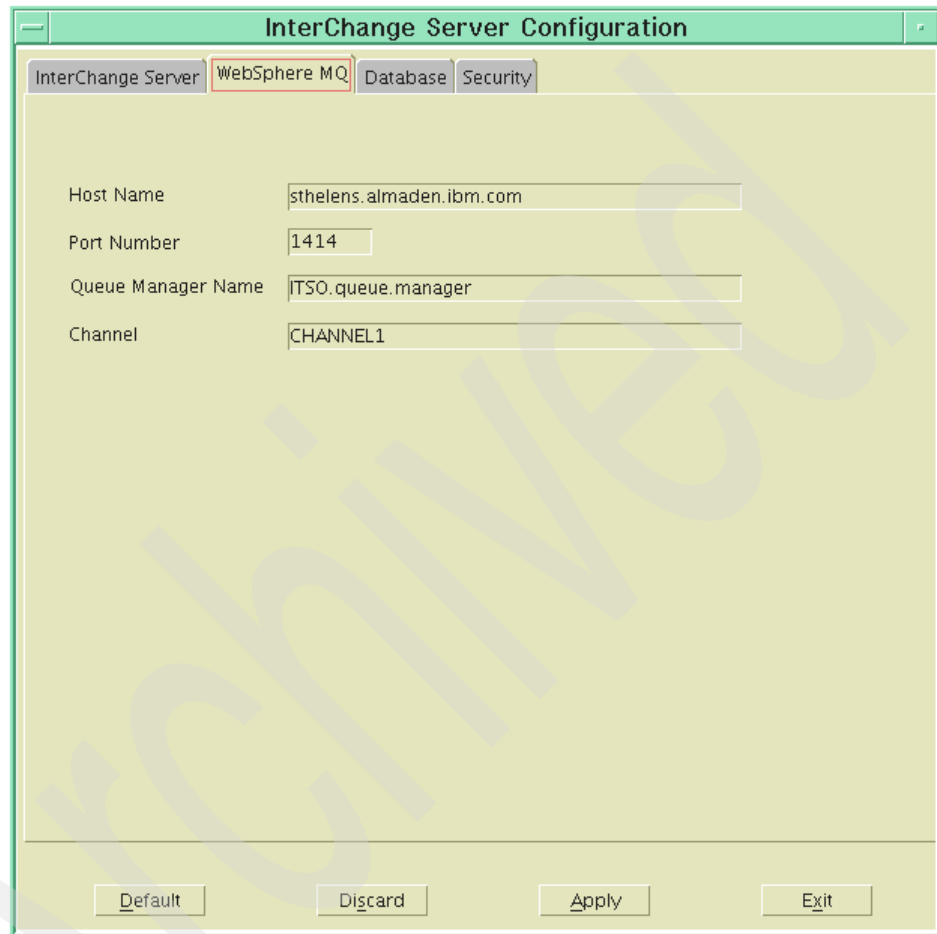


Figure 8-32 InterChange Server Configuration: InterChange Server tab

2. Figure 8-33 shows the WebSphere MQ configuration tab. Make sure the previous WebSphere MQ settings for IBM CrossWorlds InterChange Server 4.1.1.9 are entered here.



The image shows a window titled "InterChange Server Configuration" with four tabs: "InterChange Server", "WebSphere MQ", "Database", and "Security". The "WebSphere MQ" tab is selected and highlighted with a red border. The configuration fields are as follows:

Field	Value
Host Name	sthelens.almaden.ibm.com
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

At the bottom of the window, there are four buttons: "Default", "Discard", "Apply", and "Exit".

Figure 8-33 InterChange Server Configuration: WebSphere MQ tab

3. The third tab is for database configuration (Figure 8-34). Make sure to enter the connection information from WebSphere InterChange Server 4.1.1 and apply the following settings:
- Max Connections: unlimited, if possible. For the purposes of this book, the scenario was successfully migrated with the Max Connections parameter set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' window with the 'Database' tab selected. The window has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab is active and contains several sections for configuration:

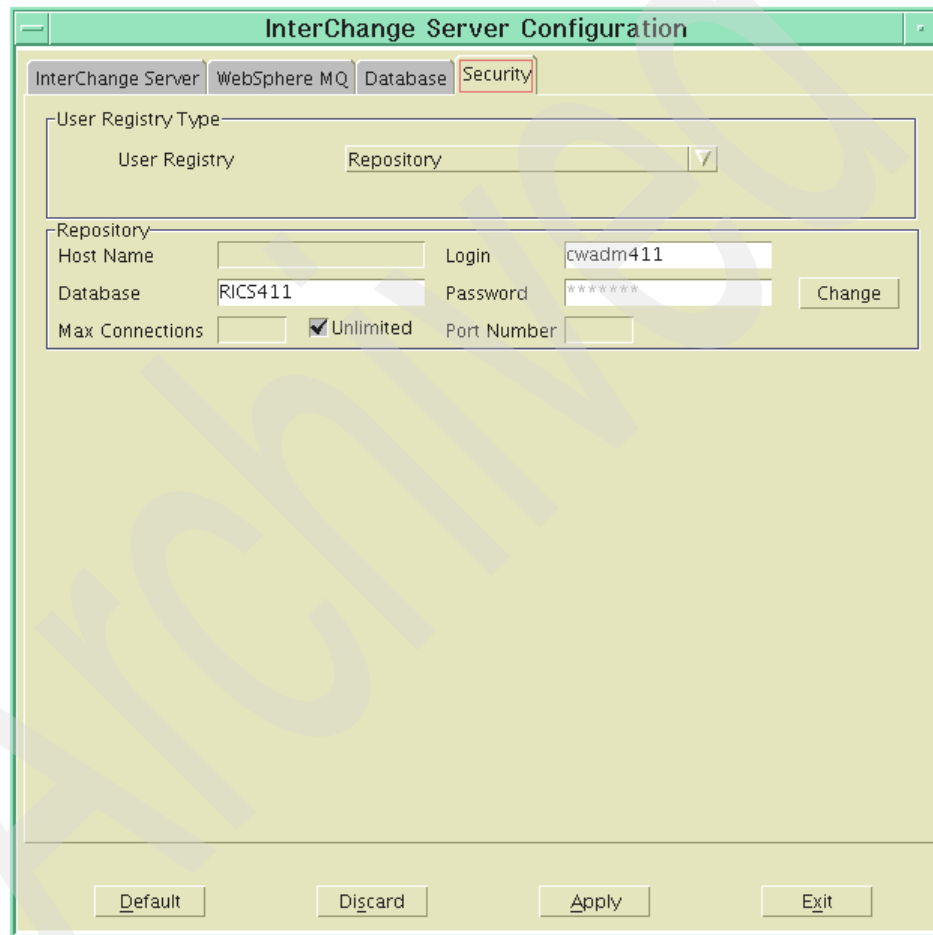
- Database Connectivity:** Includes a dropdown for 'Database Driver' set to 'DB2', and input fields for 'Max Connections' (set to 'Unlimited'), 'Max Pools' (set to '50'), and 'Idle Timeout' (set to '2').
- Event Management:** Includes input fields for 'Host Name', 'Database' (set to 'RICS411'), 'Login' (set to 'cwadm411'), and 'Password' (masked with '\*\*\*\*\*'). There is a 'Change' button and a 'Max Connections' dropdown set to 'Unlimited'.
- Transactions:** Similar to Event Management, with 'Host Name', 'Database' (set to 'RICS411'), 'Login' (set to 'cwadm411'), 'Password' (masked), 'Change' button, and 'Max Connections' set to 'Unlimited'.
- Repository:** Similar to the others, with 'Host Name', 'Database' (set to 'RICS411'), 'Login' (set to 'cwadm411'), 'Password' (masked), 'Change' button, and 'Max Connections' set to 'Unlimited'.
- Flow Monitoring:** Includes 'Host Name', 'Database' (set to 'RICS411'), 'Login' (set to 'cwadm411'), 'Password' (masked), 'Change' button, 'Schema Name', 'Port Number', and 'Max Connections' set to 'Unlimited'.

At the bottom of the window are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 8-34 InterChange Server Configuration: Database tab

4. The fourth tab is for security configuration (Figure 8-35). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.



The image shows a screenshot of the 'InterChange Server Configuration' window, specifically the 'Security' tab. The window has a title bar with the text 'InterChange Server Configuration'. Below the title bar, there are four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Security' tab is selected and highlighted. The main content area of the 'Security' tab is divided into two sections. The top section is labeled 'User Registry Type' and contains a dropdown menu with 'Repository' selected. The bottom section is labeled 'Repository' and contains several input fields: 'Host Name' (empty), 'Login' (containing 'cwadm411'), 'Database' (containing 'RICS411'), 'Password' (containing '\*\*\*\*\*'), 'Max Connections' (empty), 'Unlimited' (checked), and 'Port Number' (empty). There is a 'Change' button next to the 'Password' field. At the bottom of the window, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 8-35 InterChange Server Configuration: Security tab

To save the settings, click **Apply** button. To exit the configurator, click **Exit** button.

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMs and collaboration directories from WebSphere InterChange Server 4.1.1 to the new installation directory for Version 4.3.0. Be sure to include all subdirectories.

Example from this migration scenario:

```
cp -R /OLD INSTALLATION PATH/DLMs /NEW INSTALLATION PATH/
cp -R /OLD INSTALLATION PATH/collaborations /NEW INSTALLATION PATH/
```

2. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.

For example, to disable DEADLOCK\_DETECTOR\_CHECK for collaborations, edit the config file code shown in Example 8-6 using a suitable editor for XML.

### Example 8-6 Disable DEADLOCK\_DETECTOR\_CHECK

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>false</tns:reposController>
<tns:reposAgent>false</tns:reposAgent>
<tns:localConfig>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change true to false in the line starting with <tns:value.

3. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, these must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
4. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in %CROSSWORLDS/bin/CWSharedEnv.sh. The external classes are first added to a new variable EXT\_LIB. Then the EXT\_LIB is appended to the CWCLASSES variable. Refer to the following example from CWSharedEnv.sh:

```
# external classes added here
```

```
EXT_LIB=${CROSSWORLDS}/lib/Sample.jar
```

```
CWCLASSES=${WLICENSE}:${CORECW}:${DATAHANDLERS}:${DATADIRECT}:${DATAMANAGER}:${ITLM}:${JAVAMAIL}:${WEBSERVICES_JARS}:${EXT_LIB}
export CWCLASSES
```

5. Modify and verify the environment variables as documented in the *IBM WebSphere InterChange Server System Installation Guide for UNIX*.

For example, make sure the path to the Java 1.4.2 compiler precedes the Java 1.3.1 compiler if the 1.3.1 compiler is still in the path. The following example represents part of the \$HOME/.profile from one of the test servers:

```
#Start DB Environment
. /home/db2inst1/sqllib/db2profile
export EXTSHM=ON
# End DB Environment

# WBI environment
export PATH=/usr/java14/bin:$PATH
. /opt2/ics43/bin/CWSharedEnv.sh
#END WBI environment
```

Make sure the latest settings are in effect. To confirm that the right Java Compiler is activated, type **which javac**.

The following output is returned for AIX, which is the installation path for Java 1.4.2 JDK:

```
/usr/java14/bin/javac
```

6. Install the WebSphere Business Integration Toolset on a supported Windows client. If the tools are to be installed on a Windows client that currently hosts an older version of tools, the old tools must be uninstalled prior to the installation of the new tools. You should make a backup of the Windows client before uninstalling and installing the new tools.

During installation, a prompt for the location of the Object Request Broker is displayed. Enter the host name or IP address of the machine hosting the Persistent Name Server that WebSphere InterChange Server is using. Also enter the port number if it is different from the default, 14500. To change the settings after installation, edit %CROSSWORLDS%\bin\CWSharedEnv.bat:

```
set ORB_PORT=14500
set ORB_HOST=persian.almaden.ibm.com
```

## Starting the new server

When WebSphere InterChange Server is started for the first time after installation, the WebSphere InterChange Server repository tables are upgraded

or created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running
  - WebSphere MQ queue manager and Listener.
  - Database.
2. Start the Persistent Name Server. In a new console run the following command:

```
$CROSSWORLDS/bin/PersistentNameServer.sh
```

**Restriction:** At the time of this writing, a product defect causes the WebSphere InterChange Server to hang during the first startup. This defect is only applicable for Version 4.1.1.9 when using DB2 as the repository database. This defect has been reported and assigned tracking number CR60364. Check the support site below to see if a fix for this problem number is available.

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

In our scenario, a DDL script is used to circumvent the problem. This workaround manually creates five database tables and indices. Use the DB2 Command Center or command line processor to load the DDL script which can be found in “DDL code for WebSphere InterChange Server 4.1.1.9” on page 466. Change the database name, user and password to match the environment. This workaround is reviewed and approved by the WebSphere InterChange Server Development team.

3. Start the WebSphere InterChange Server in design mode.

```
$CROSSWORLDS/bin/ics_manager -start -design
```

The WebSphere InterChange Server upgrades existing tables and create new tables for new features.

The WebSphere InterChange Server logs messages to the InterChangeSystem.log file located in the product directory. An Interchange Server is Ready message indicates that the WebSphere InterChange Server has started successfully.



## Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Import the previous repository into the tooling.
  - a. Start the System Manager.
  - b. Connect to the WebSphere InterChange Server.
  - c. Create a new integration component library and import the components from the server. During the import, maps and collaboration templates are upgraded to the compatible 4.3.0 format. Refer to the *Implementation Guide for WebSphere InterChange Server* for WebSphere InterChange Server for details on how to import components using the Import components wizard of System Manager.

**Note:** If the repository contains an unusually large number of components, it is recommended to import the WebSphere InterChange Server components in pieces. Importing the entire repository at one time may be slow and might cause memory errors in the System Manager. The recommended order of component import is as follows.

1. Business objects
2. Maps
3. Collaboration templates and objects
4. Connectors
5. Relationships and connection pools

6. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH for the Toolset. Add the location of the custom .jar files to this list. Figure 8-36 shows an example.

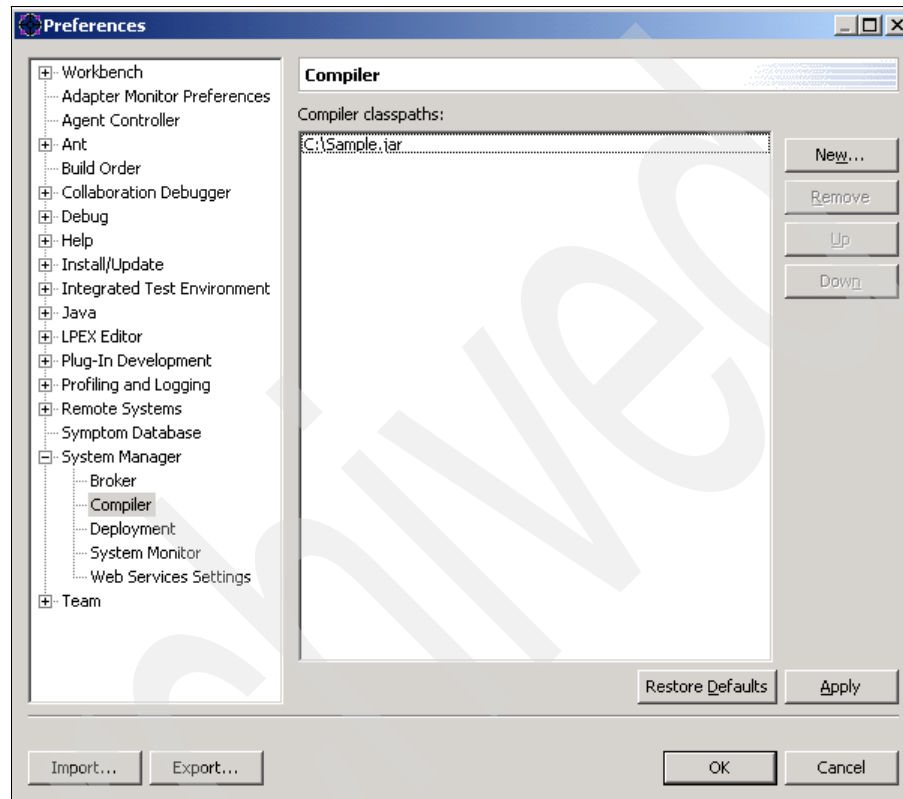


Figure 8-36 Setting the classpath for the Toolset

7. Set the compliance of the compiler to Java 1.4. Click **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag change the Compiler compliance level to 1.4. A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to not rebuild files at this time. Figure 8-37 shows an example.

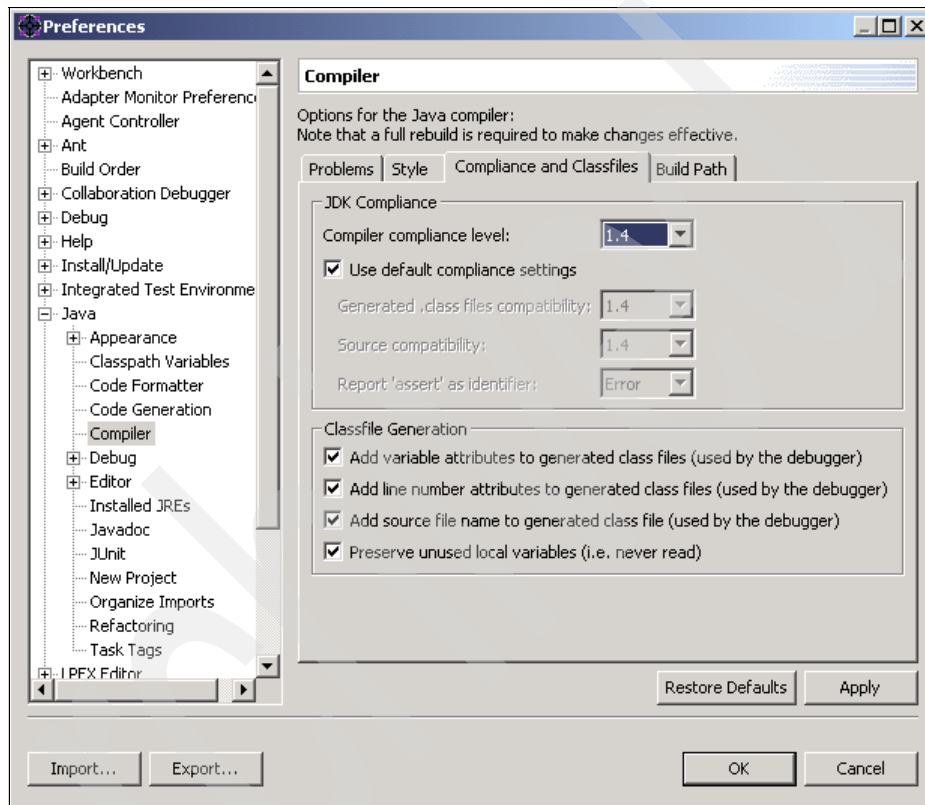


Figure 8-37 Setting the Compiler compliance level

8. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the usage of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level has been correctly set in previous step.
9. Compile all the maps and templates in the integration component libraries and examine the console for any error messages.

10. The queue conventions for JMS delivery transport has changed between 4.1.1.9 and 4.3.0. The configuration for all connectors that use JMS delivery transport need to be examined and modified.
  - a. Edit the connector definition by double clicking the connector name in the System Manager. This launches the Connector Configurator. The default queue names for JMS delivery transport are displayed in the Standard Properties tab. The default queue names can be used as is, or they can be modified if needed.
  - b. Create new queues in the queue manager with the appropriate queue names specified in the previous step.
  - c. In the Connector Configurator, save the configuration file to the project, and save a copy to file for use by the adapter agent.
11. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more details.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
12. Create a new user project and populate it the component from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for WebSphere InterChange Server for details.
13. Resolve any collaborations that are transactional status, because transactional status is lost after redeployment of the collaboration. This can be done from the Flow Manager. If the collaboration cannot be brought to an active state to resolve the transactional status, deploy only the updated components required to activate the integration, then resolve the transactional status.

**Note:** To make use of all administrative features for maps and collaborations, the maps and templates need to be deleted and redeployed. Typically, each component must be placed in an inactive state before it can be overwritten. To simplify the process and to avoid any unresolved dependencies, the whole repository is deleted in step 10 and redeployed in step 11. Additionally, recompilation ensures that the code is optimized and Java 1.4 compliant.

However, it is not required to redeploy the maps and templates, because the Java Runtime Environment is backwards compatible and poses no problems executing map and template class files compiled in WebSphere InterChange Server 4.1.1.

If the decision is made to reuse the previous maps and collaborations, only deploy the components which have been updated.

14. In the System Manager, delete the repository by right-clicking the WebSphere InterChange Server name in the InterChange Server component management view and select **Delete the repository**.

15. Deploy the user project:

- a. Right-click the user project and select **Deploy User Project**.
- b. Click **Select the Destination Server** and choose the server name.
- c. Select the components to be deployed.
- d. Select **Perform Compilation on Interchange Server machine**.
- e. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This recreates the stored procedures and does not overwrite existing data.

16. Restart the WebSphere InterChange Server if prompted.

### 8.3.3 Post-migration steps

This section describes the steps that need to be taken after migration and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

## Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user, with the previous password, assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.
2. Right-click the server name in the InterChange component management view and select **Users and Roles**. This is shown in Figure 8-38.

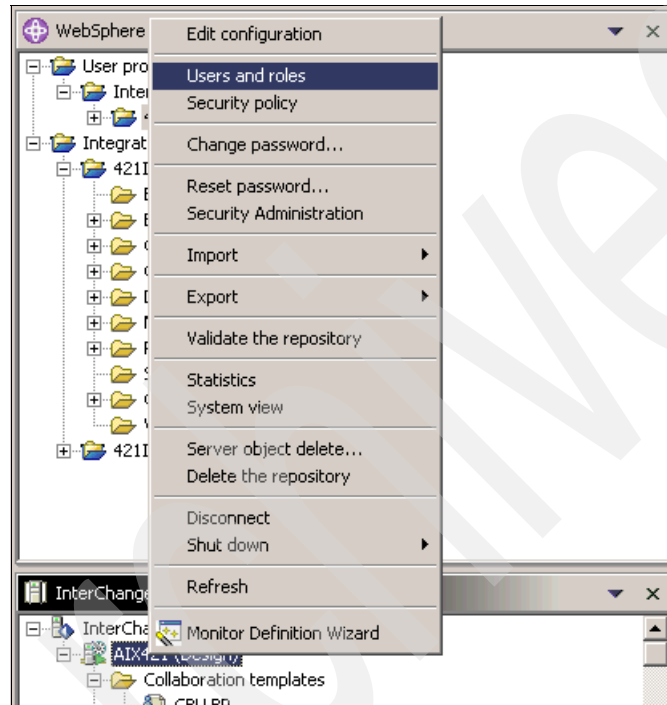


Figure 8-38 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click Users icon and select **New User**. This is shown in Figure 8-39.

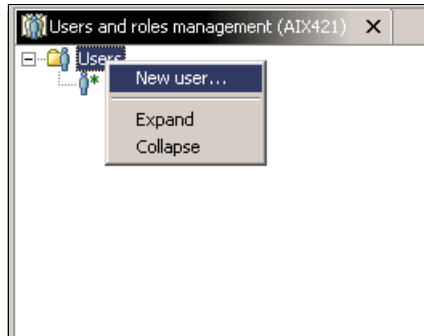


Figure 8-39 Selecting New user

4. The window Create User comes up. This is shown in Figure 8-40. Enter the name admin and select the previous password plus additional information if needed. Press **OK** when finished.



Figure 8-40 Creating a user

5. The new user is added to the User and Roles management window. This is shown in Figure 8-41.

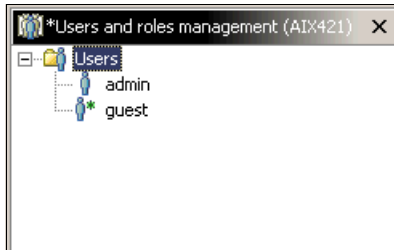


Figure 8-41 The newly created admin user

6. Right-click the newly created admin user and select **Add roles**. This is shown in Figure 8-42.

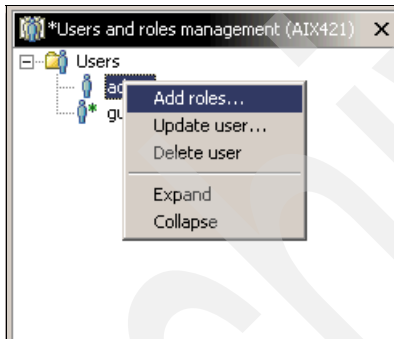


Figure 8-42 Add roles



7. The Role list window is displayed. This is shown in Figure 8-43.

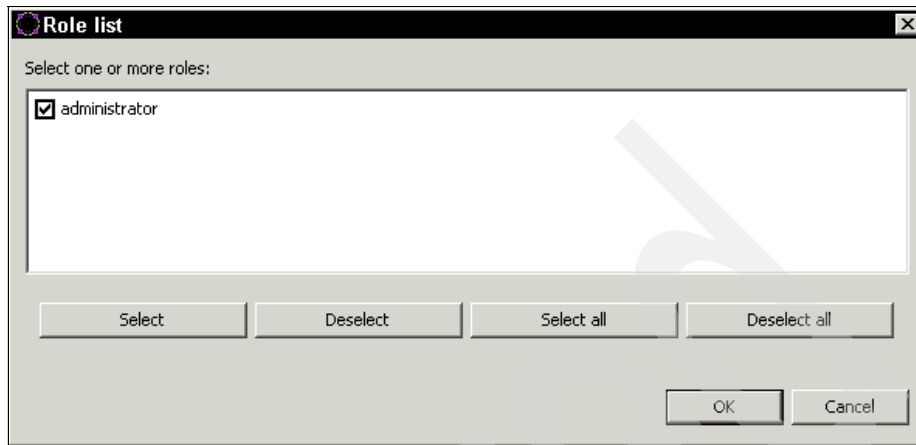


Figure 8-43 Role list

8. Select **administrator** and click **OK**.

9. Close the Users and roles management window. This window is shown in Figure 8-44.



Figure 8-44 User admin with administrator role

10. Right-click the server again in the InterChange Components management view and choose **Edit Configuration**. An editor to configure the server is displayed.

11. Click the **Security - RBAC** tab as shown in Figure 8-45.

The screenshot shows the 'Security - RBAC' configuration window for the \*AIX421 server. The window is divided into several sections: RBAC setting, Audit setting, LDAP setting, and Repository details. The RBAC setting section has 'Enable RBAC' checked, 'User registry' set to 'Repository', 'Server start user' set to 'admin', and 'Server start password' set to '\*\*\*\*'. The Audit setting section has 'Enable audit' checked, 'Audit log directory' set to '/opt2/ics431/logs/Audit', 'Audit log frequency' set to 'WEEKLY', and 'Audit file size (MB)' set to '1'. The LDAP setting section has fields for 'LDAP URL', 'User name DN', 'Password', 'Userbase DN', 'User name attribute', 'Search criteria', and 'Maximum number of search returns', all of which are currently empty. There is also an unchecked 'SSL' checkbox. The Repository details section has 'Host name' and 'Database' fields (the latter is set to 'RICS421'), 'Maximum number of connections' set to 'Unlimited', 'Max connect retries' set to '3', 'Login' set to 'cwadm421', 'Password' set to '\*\*\*\*\*' with a 'Change' button, 'Port number' set to '0', and 'Connect retry interval' set to '60' seconds. At the bottom, there is a tab bar with the following tabs: General, Database, Tracing levels, Trace and log files, E-mail, CORBA, Miscellaneous, Environment properties, WebSphere MQ, Security - RBAC (selected), and Privacy.

Figure 8-45 Server configuration: Security-RBAC tab

12. In the tab, apply the following settings:

- Select **Enable RBAC**.
- Set **Server start user** to admin.
- Set **Server start password** to be the password for the admin user.
- Check all other settings in the tab and modify if needed.
- Save the configuration to the server by pressing the save icon, or choosing **File → Save**.

13. Reboot WebSphere InterChange Server to activate security.

**Note:** The adapter agent default to the Guest account if RBAC is activated for the WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual

Test Connector. For more detailed instructions on how to install adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they need to be updated to work with WebSphere InterChange Server 4.3. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end and performance testing. Before moving the migrated InterChange Server system from development to production it is recommended to perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 8-4 shows the scenarios that were tested in this migration, and the results found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.

Table 8-4 Test scenarios results

Test scenario	Results
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated

A failed flow is considered to be successfully migrated if the event can be resubmitted and completed after migration. Table 8-6 shows the results of the migration of failed flows.

Table 8-5 Migrated failed flows results

Failed flows tested	Results
Failed flow for IDL	Successfully migrated
Failed flow for MQ	Successfully migrated
Failed flow for JMS	Successfully migrated
Transactional collaboration in transactional status (compensation call failed)	Successfully migrated
Transactional collaboration failed flow (compensation call successful)	Successfully migrated
Collaboration group failed at Destination 1	Successfully migrated
Collaboration group failed at Destination 2	Successfully migrated

**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that the ICS is re-started in Production mode.

### Uninstall (optional)

This section describes the steps to uninstall the previous version of WebSphere InterChange Server and the related software. The following steps list the order of uninstalling relevant software.

1. Uninstall the VisiBroker ORB using the uninstaller located in the VisiBroker product directory.
2. Uninstall JDK as the root user using `smi t`, only if it is not required by other products. For this release, the JDK is typically Java 2 SDK v.1.3.1. Contact the system administrator for further assistance.
3. Uninstall WebSphere InterChange Server 4.1.1 using the uninstaller located in the product directory once it is certain that the previous version is no longer needed. After the uninstaller has completed, the product directory can be deleted.

## 8.4 Without in-place database migration of V4.1.1.9 to V4.3.0 on AIX

This section describes a without in-place database migration scenario of WebSphere InterChange Server Version 4.1.1.9 to Version 4.3.0. The 4.3.0 WebSphere InterChange Server is started with a new and empty repository.

Repository content (component definitions) is exported from the original 4.1.1.9 InterChange Server and imported into the 4.3.0 WebSphere InterChange Server.

For the full description of an without in-place database migration refer to Chapter 4, “General planning” on page 53.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

## 8.4.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system.

### Important:

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The *IBM WebSphere InterChange Server System Installation Guide for UNIX* has more information about other UNIX platforms.

### Put the environment in a quiescent state

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator.
2. Resolve transactional status of transactional collaborations using the Flow Manager. Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details. All status information is lost during the migration.

3. Resubmit failed events or discard the events using Flow Manager. Refer to the *System Administration Guide for CrossWorlds 4.1.1* for details. Failed flows are lost during the migration.
4. When all events have been processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in CrossWorlds System Monitor (CSM). Make sure that the in-process and pending event counter is 0.

## Backing up and shutting down components

This section describes the steps for shutting down and taking a backup. It is a recommended practice to take a backup that will enable the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3.0.

Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*. The WebSphere InterChange Server system has two types of data: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current WebSphere InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is WIN411 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository411.txt. This file is created in the directory where the **repos\_copy** command is run.

```
repos_copy -sWIN411 -uadmin -pnull -oRepository411.txt
```

2. Shut down the WebSphere InterChange Server:

```
ics_manager -stop
```

3. Arrange with the database administrator to back up the database. This must be a complete backup, including schema information and stored procedures.
4. Stop the Inprise VisiBroker Agent. Run the following command to determine the process ID of the osagent:

```
ps -ef | grep osagent
```

This is a sample output of the command:

```
cwadm421 35012      1   0   Aug 09      -   1:18 osagent
```

To stop the osagent, run the command:

```
kill -9 35012
```

(35012 is the process ID of the osagent process.)

5. Clear WebSphere MQ queues of any outstanding messages if the same queue manager is reused for Version 4.3.0. Old messages in the queues could be seen as new events and may cause data inconsistencies. Refer to the WebSphere MQ documentation for more information.

## 8.4.2 Migration steps

This section describes the main steps required for the migration:

- ▶ Upgrading related software to the required level
- ▶ Installing and configuring WebSphere InterChange Server 4.3.0
- ▶ Completing the upgrades of WebSphere InterChange Server components

### Preparing the database system

Create an empty database for the V4.3 repository following the steps in the *IBM WebSphere InterChange Server System Installation Guide for UNIX*:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **UNIX** in the right-side pane.

Ensure that this database has the proper configuration for WebSphere InterChange Server 4.3:

- ▶ Check the installation guide for the proper database settings for 4.3. For example, when using DB2, one change in the database settings is that the `aplleapsz` parameter must be set to at least 4096.

- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

- ▶ For DB2, it is recommended that you catalog the database system to route using TCP/IP as documented in the System Installation Guide for UNIX.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be at Version 5.3.0.2 Fix Pack 7. For detailed instructions, refer to the WebSphere MQ documentation at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 for instructions.

It is recommended that WebSphere InterChange Server Version 4.3.0 be installed with a different user ID and to a different directory than the previous version. The previous installation is not modified during the install process.

During the install process, the WebSphere InterChange Server name may be changed. If so, the adapters and queues may have to be configured. Use the same WebSphere InterChange Server name if possible.

The WebSphere InterChange Server name can be changed after the installation process. Edit the file `$CROSSSWORLDS/bin/ics_manager` and the file `$ADAPTER_RUNTIME/bin/connector_manager`. Change the parameter `SERVERNAME` to the desired WebSphere InterChange Server name.



## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following steps explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
$CROSSWORLDS/bin/ICConfig.sh
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all configuration settings must be entered in the Options.txt file, which is delivered with the installer. For detailed information, refer to the Installation Guide for UNIX (page 69).

The following figures show the configuration values used by the redbook team during migration:

1. Example 8-46 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

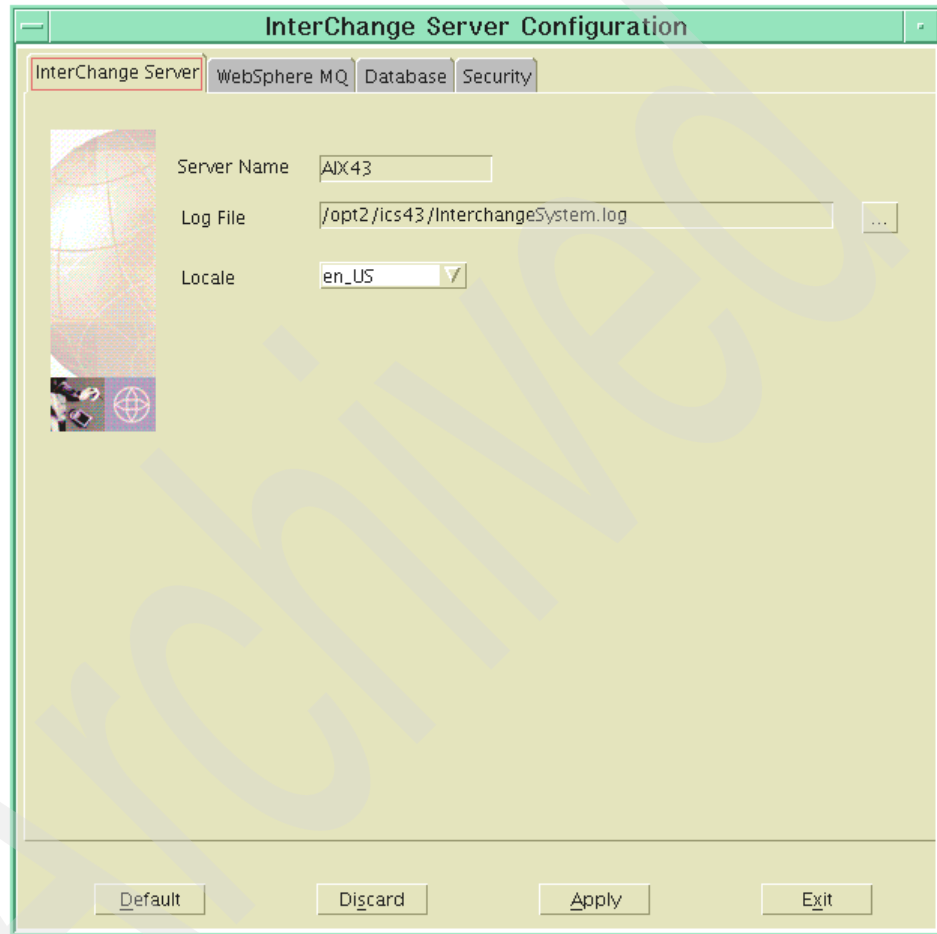
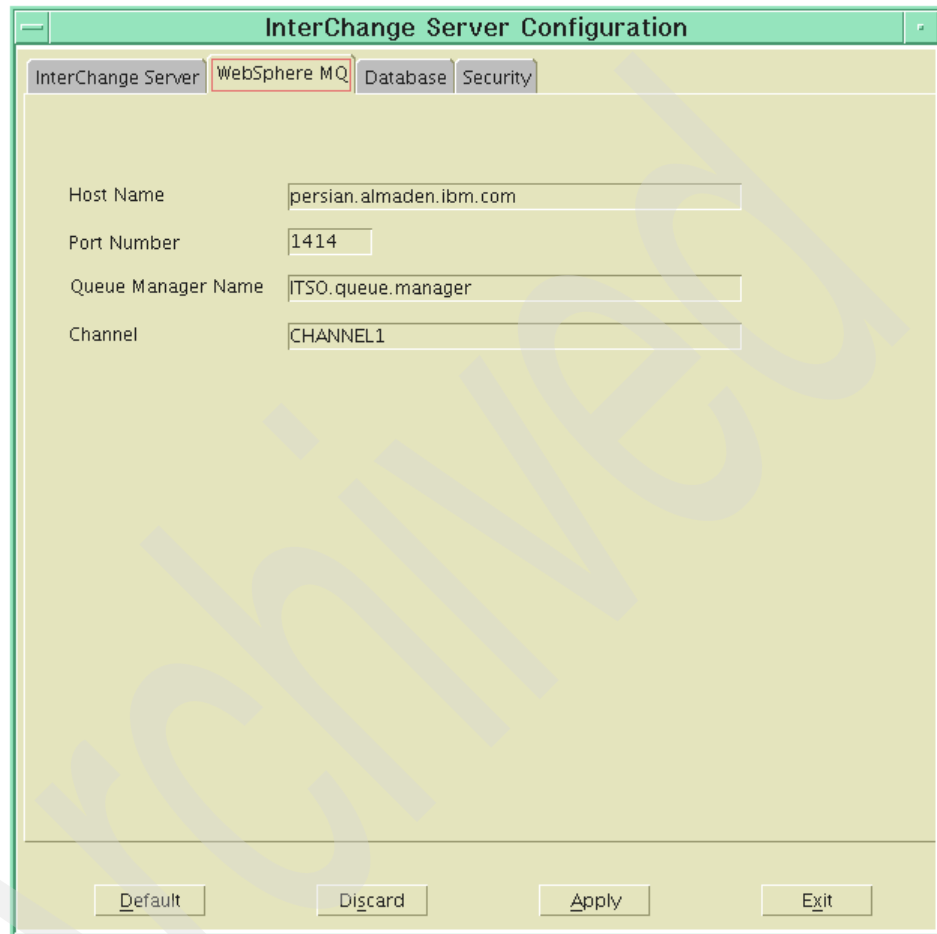


Figure 8-46 InterChange Server Configuration: InterChange Server tab

- Figure 8-47 shows the WebSphere MQ configuration tab. Make sure the previous WebSphere MQ settings for IBM CrossWorlds InterChange Server 4.1.1.9 are entered here.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'WebSphere MQ' tab. The dialog has a title bar with the text 'InterChange Server Configuration'. Below the title bar, there are four tabs: 'InterChange Server', 'WebSphere MQ' (which is selected and highlighted with a red border), 'Database', and 'Security'. The main area of the dialog contains four labeled text input fields: 'Host Name' with the value 'persian.almaden.ibm.com', 'Port Number' with the value '1414', 'Queue Manager Name' with the value 'ITSO.queue.manager', and 'Channel' with the value 'CHANNEL1'. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 8-47 InterChange Server Configuration: WebSphere MQ tab

3. The third tab is for database configuration (Figure 8-48). Make sure to enter the connection information from WebSphere InterChange Server 4.1.1 and apply the following settings:
- Max Connections: unlimited, if possible. For the purposes of this book, the scenario was successfully migrated with the Max Connections parameter set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' window with the 'Database' tab selected. The window has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab is active and contains several sections for configuration:

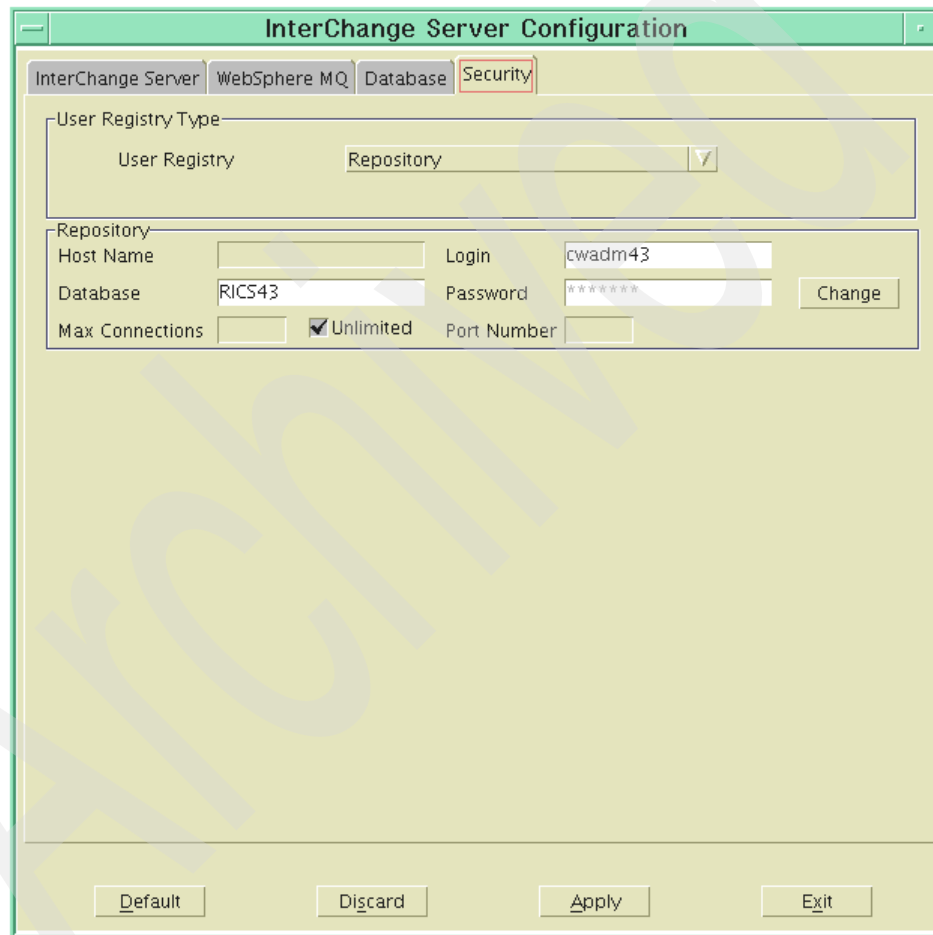
- Database Connectivity:** Includes a dropdown for 'Database Driver' set to 'DB2', a 'Max Connections' field with an 'Unlimited' checkbox checked, a 'Max Pools' field set to '50', and an 'Idle Timeout' field set to '2'.
- Event Management:** Includes fields for 'Host Name', 'Database' (set to 'RICS43'), 'Login' (set to 'cwadm43'), 'Password' (masked with asterisks), 'Max Connections' (with 'Unlimited' checked), and 'Port Number'. A 'Change' button is present.
- Transactions:** Includes fields for 'Host Name', 'Database' (set to 'RICS43'), 'Login' (set to 'cwadm43'), 'Password' (masked with asterisks), 'Max Connections' (with 'Unlimited' checked), and 'Port Number'. A 'Change' button is present.
- Repository:** Includes fields for 'Host Name', 'Database' (set to 'RICS43'), 'Login' (set to 'cwadm43'), 'Password' (masked with asterisks), 'Max Connections' (with 'Unlimited' checked), and 'Port Number'. A 'Change' button is present.
- Flow Monitoring:** Includes fields for 'Host Name', 'Database' (set to 'RICS43'), 'Login' (set to 'cwadm43'), 'Password' (masked with asterisks), 'Schema Name', 'Port Number', and 'Max Connections' (with 'Unlimited' checked).

At the bottom of the window are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 8-48 InterChange Server Configuration: Database tab

4. The fourth tab is for security configuration (Figure 8-49). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'Security' tab. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Security' tab is selected and highlighted with a red border. Inside the 'Security' tab, there is a section titled 'User Registry Type' with a dropdown menu set to 'Repository'. Below this, there is a 'Repository' section with several fields: 'Host Name' (empty), 'Login' (containing 'cwadm43'), 'Database' (containing 'RICS43'), 'Password' (containing '\*\*\*\*\*'), 'Max Connections' (empty), a checked 'Unlimited' checkbox, and 'Port Number' (empty). A 'Change' button is located to the right of the 'Password' field. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

InterChange Server Configuration			
InterChange Server WebSphere MQ Database <b>Security</b>			
User Registry Type			
User Registry		Repository	
Repository			
Host Name		Login	cwadm43
Database	RICS43	Password	*****
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number
Change			
Default Discard Apply Exit			

Figure 8-49 InterChange Server Configuration: Security tab

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMs and collaboration directories from WebSphere InterChange Server 4.1.1 to the new installation directory for V4.3.0. Be sure to include all subdirectories.

Example from this migration scenario:

```
cp -R /OLD INSTALLATION PATH/DLMs /NEW INSTALLATION PATH/  
cp -R /OLD INSTALLATION PATH/collaborations /NEW INSTALLATION PATH/
```

2. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.

For example, to disable DEADLOCK\_DETECTOR\_CHECK for collaborations, edit the config file excerpt in Example 8-7 using a suitable editor for XML.

*Example 8-7 Disable DEADLOCK\_DETECTOR\_CHECK*

---

```
<tns:property>  
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>  
<tns:value xml:space="preserve">true</tns:value>  
<tns:isEncrypted>>false</tns:isEncrypted>  
<tns:updateMethod>system restart</tns:updateMethod>  
<tns:location>  
<tns:reposController>>false</tns:reposController>  
<tns:reposAgent>>false</tns:reposAgent>  
<tns:localConfig>true</tns:localConfig>  
</tns:location>  
</tns:property>
```

---

Change true to false in the line starting with <tns:value.

3. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, these must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
4. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in %CROSSWORLDS/bin/CWSharedEnv.sh. The external classes are first added to a new variable, EXT\_LIB, which is then appended to the CWCLASSES variable. See the following excerpt from CWSharedEnv.sh:

```
# external classes added here  
EXT_LIB=${CROSSWORLDS}/lib/Sample.jar
```

```
CWCLASSES=${WLICENSE}:${CORECW}:${DATAHANDLERS}:${DATADIRECT}:${DATAMANAGER}:${ITLM}:${JAVAMAIL}:${WEBSERVICES_JARS}:${EXT_LIB}
export CWCLASSES
```

5. Modify and verify the environment variables as documented in the Installation Guide for UNIX.

For example, make sure the path to the JAVA 1.4.2 compiler precedes the JAVA 1.3.1 compiler if the JAVA 1.3.1 compiler is still in the path. This excerpt represents part of the \$HOME/.profile from one of our test servers:

```
#Start DB Environment
. /home/db2inst1/sqllib/db2profile
export EXTSHM=ON
# End DB Environment

# WBI environment
export PATH=/usr/java14/bin:$PATH
. /opt2/ics43/bin/CWSharedEnv.sh
#END WBI environment
```

Make sure the latest settings are in effect. To confirm that the right Java Compiler is activated, run the **which javac** command.

The following output is returned for AIX, which is the installation path for Java 1.4.2 JDK:

```
/usr/java14/bin/javac
```

6. Install the WebSphere Business Integration Toolset on a supported Windows client. If the tools are to be installed on a Windows client that currently hosts an older version of tools, the old tools must be uninstalled prior to the installation of the new tools. It is recommended to take a backup of the Windows client before uninstalling and installing the new tools.

During installation, a prompt for the location of the Object Request Broker is displayed. Enter the host name or IP address of the machine hosting the Persistent Name Server that the WebSphere InterChange Server is using. Also enter the port number if it is different from the default, 14500. To change settings after installation, edit %CROSSWORLDS%\bin\CWSharedEnv.bat.

```
set ORB_PORT=14500
set ORB_HOST=persian.almaden.ibm.com
```

## Starting the new server

When WebSphere InterChange Server is started for the first time after installation, the WebSphere InterChange Server repository tables are created. This section describes the steps to prepare and start the server for the first time.

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database
2. Start the Persistent Name Server. In a new console run the command:  
`$CROSSWORLDS/bin/PersistentNameServer.sh`
3. The V4.1.1.9 repository file may have to be edited in a text editor before importing into V4.3.0. Save a copy of the original 4.1.1.9 repository file before making these changes.
  - Connection pools: If a connection pool is not accessible by the 4.3.0 WebSphere InterChange Server, delete the connection pool entry in the repository file. The connection pool must be re-created at a later step. Example 8-8 shows a sample connection pool definition.

### *Example 8-8 A ConnectionPool definition*

---

```
[ConnectionPool]
Compressed = TRUE
Checksum = 1222416404
[BeginBlob]
UESDBBQACAAIAKKCVDEAAAAAAAAAAAAAAAAAAAAUnVudG1tZUVudG10eW2QTQ7CIBCF9z1Fw15RV8Z
QTNPowmhiTHsABKwYmEkAf44vNrFR62YI32Pe8IYtH87mN+DQSjIdDwhuQaJykBbkKZeJ+ZkyTN2uEI
OTq9SjUaH746kNOH7CgG0jA1XQp71gCbW8QThZFr0NuooG2/5JZOLdZwtDqttWTP65p0HCKd5EoVyBh
jtEduLE07oFTcxIKP9ldHBpN/f7Bft/nIpyA7BRPQkT+ltShF7LJNCE1tdJCC/k9Lf5bEsydQSwcIHZ
kAhr4AAABeAQAUEsBAhQAFAAIAAgAooJUMR2ZAB+AAAAXgEAAA0AAAAAAAAAAAAAAAAAAAAAFJ1bn
RpbWVfbnRpdH1QSwUGAAAAAAEAAQA7AAAA+QAAAAAA
[EndBlob]
[End]
```

---

- Relationship definition: For each relationship definition that is not accessible by the 4.3.0 WebSphere InterChange Server, delete the entire lines for DatabaseURL, LoginName, LoginPassword. The new relationship tables are configured and created at a later step. Example 8-9 shows a section of the Relationship definition after the modification.

### *Example 8-9 A Relationship definition after modification*

---

```
[RelationshipDefinition]
Name = Customer
Version = 1.0.0
State = Active
Identity = true
```



```
Static = false
Cached = false
DatabaseType = 3

[Role]
Name = CwCust
CwManagedIds = true
BusinessObject = Customer
[...]
```

---

- Any schedule information in the 4.1.1 repository file must be deleted.  
Example 8-10 shows a sample Schedule Entry.

*Example 8-10 A Schedule Entry*

---

```
[ScheduleEntry]
ID = 1
ComponentType = 1
ComponentName = ScheduleCollaboration_App1_to_App2
ComponentVersion = 1.0.0
Status = 1
Action = 1
InitialDate = Friday, October 8, 2004 8:26:48 PM CEST
NextOccurrence = Tuesday, October 12, 2004 10:26:48 PM CEST
RecurrenceType = 2
RecurrencePeriod = 1
RecurrenceNumWeeks = 58080488
UserComments =
[End]
```

---

4. Start the WebSphere InterChange Server in design mode:

```
$CROSSWORLDS/bin/ics_manager -start -design
```

The WebSphere InterChange Server logs messages to the `InterChangeSystem.log` file located in the product directory. An Interchange Server is Ready message indicates that the WebSphere InterChange Server has started successfully.

### **Migrating repository content to WebSphere ICS 4.3.0**

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Load the modified repository file into the WebSphere InterChange Server by using the **repos\_copy** command line utility:

```
repos_copy -sWICS_NAME -ilocation of Repository411.txt
```

If the relationship definition has been edited in the previous step to change the database location, add the option `-r*` to the **repos\_copy** command line. This option instructs WebSphere InterChange Server to *not* create the relationship tables and stored procedures at this time.

```
repos_copy -sWICS_NAME -r* -ilocation of Repository411.txt
```

**Note:**

- ▶ If the relationship definition has been edited to remove the database location, use the `-r*` option during **repos\_copy**. Otherwise, the WebSphere InterChange Server creates the relationship tables and stored procedures using the default location of the repository database.
- ▶ If the **repos\_copy** fails, use the command **repos\_copy -d -sWICS\_NAME** to delete the repository and restart the WebSphere InterChange Server before subsequent attempts to load the repository file.
- ▶ The **repos\_copy** utility does not support the replace (`-ar -arp`) options for duplicates for Version 4.1.1 repository files. If an error message is received indicating that there are duplicate items in the repository file, they must be removed manually.

2. Restart the WebSphere InterChange Server after a successful **repos\_copy**.
3. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Connect to the WebSphere InterChange Server.
  - c. Create a new integration component library and import the components from the server. During the import, maps and collaboration templates are upgraded to the compatible 4.3.0 format. Refer to the *Implementation Guide for WebSphere InterChange Server* for details about importing components using the Import components wizard of System Manager.

**Note:** If the repository holds an unusually large number of components, it is recommended that you import the ICS components in pieces. Importing the entire repository at one time may be slow and might cause memory errors in the System Manager. The recommended order for importing components is:

1. Business objects
2. Maps
3. Collaboration templates and objects
4. Connectors
5. Relationships and connection pools

6. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH for the toolset. Add the location of the custom .jar files to this list (Figure 8-50).

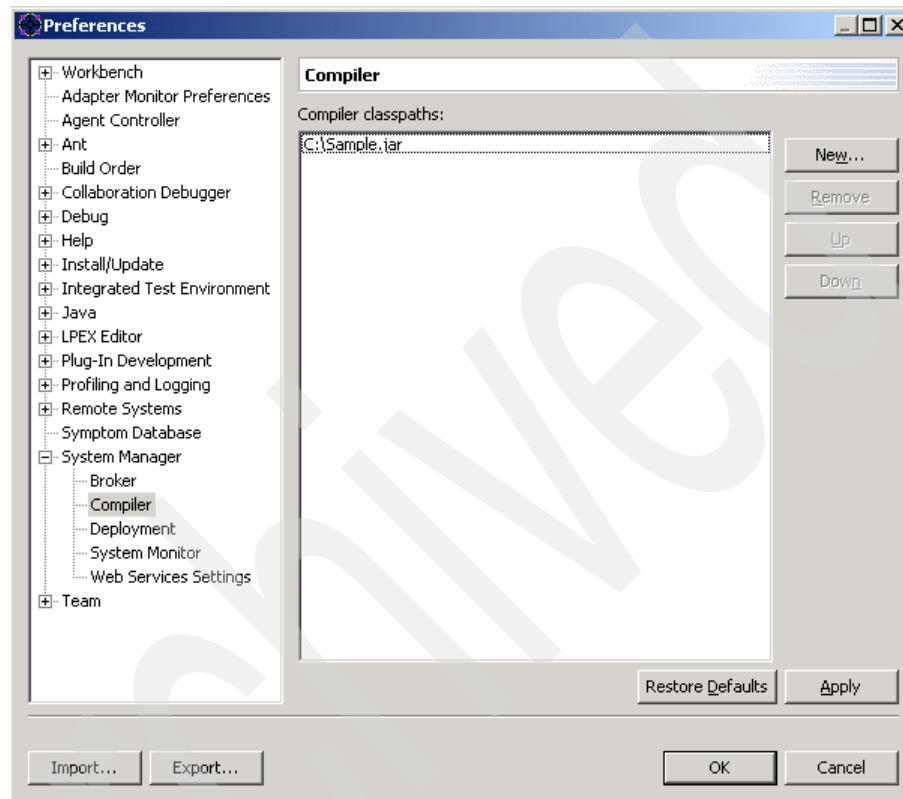


Figure 8-50 Setting the classpath for the toolset

7. Set the compliance of the compiler to Java 1.4. Click **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag change the Compiler compliance level to **1.4**. A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to not rebuild files at this time. Figure 8-51 shows an example.

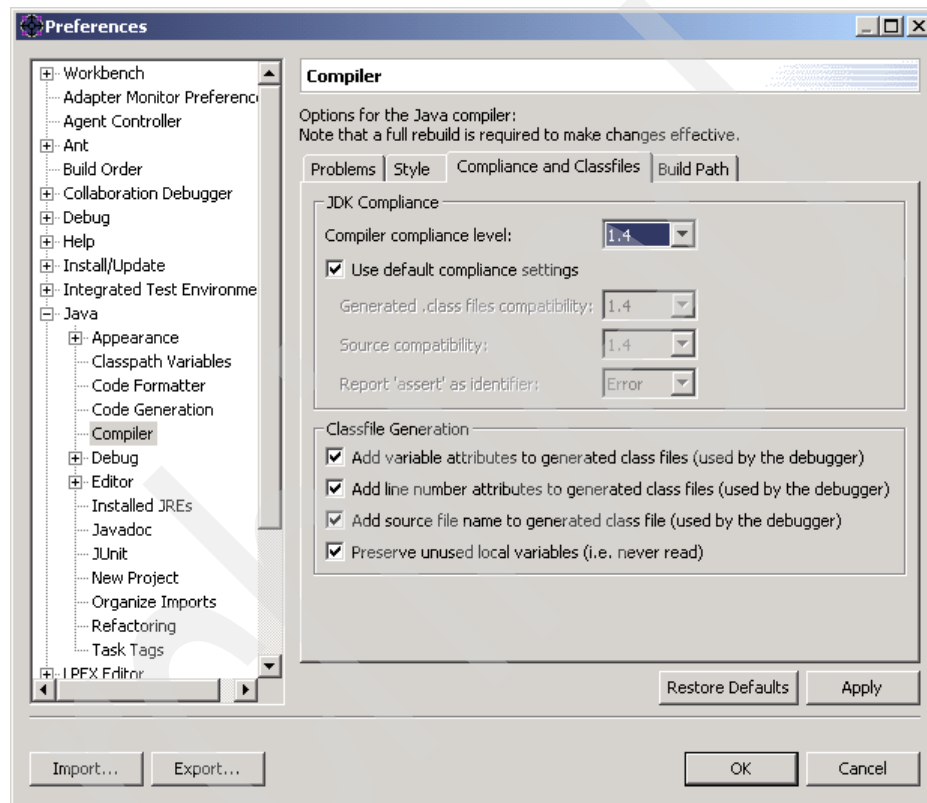


Figure 8-51 Setting the Compiler compliance level

8. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the usage of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in the previous step.
9. Compile all maps and templates in the integration component libraries and examine the console for any error messages.

10. The queue conventions for JMS delivery transport changed between 4.1.1 and 4.3.0. The configuration for all connectors that use JMS delivery transport must be examined and modified.
  - a. Edit the connector definition by double-clicking the connector name in the System Manager. This launches the Connector Configurator. The default queue names for JMS delivery transport are displayed in the Standard Properties tab. The default queue names can be used as is, or they can be modified if needed.
  - b. Create new queues in the queue manager with the appropriate queue names specified in the previous step.
  - c. In the Connector Configurator, save the configuration file to the project, and save a copy to file for use by the adapter agent.
11. Reconfigure any other components that require modification:
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more details.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
12. Create a new user project and populate it with the component from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for details.
13. Deploy the user project.
  - a. Right-click the user project and select **Deploy User Project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. If the location of the relationship tables has changed, select **Create Schema**.
  - f. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This recreates the stored procedures and does not overwrite existing data.

14. Restart the WebSphere InterChange Server if prompted.

15. Completing relationship migration: If a new relationship location is used, the the relationship data from the 4.1.1 relationship tables can be migrated as so:
  - a. For both static and dynamic relationships, copy the data from each relationship table to the new relationship table. (Consult with the database administrator.) The steps to complete this process using DB2 are:
    - i. In the Control Center, right-click the original relationship table and choose **Export**. This is shown in Figure 8-52.

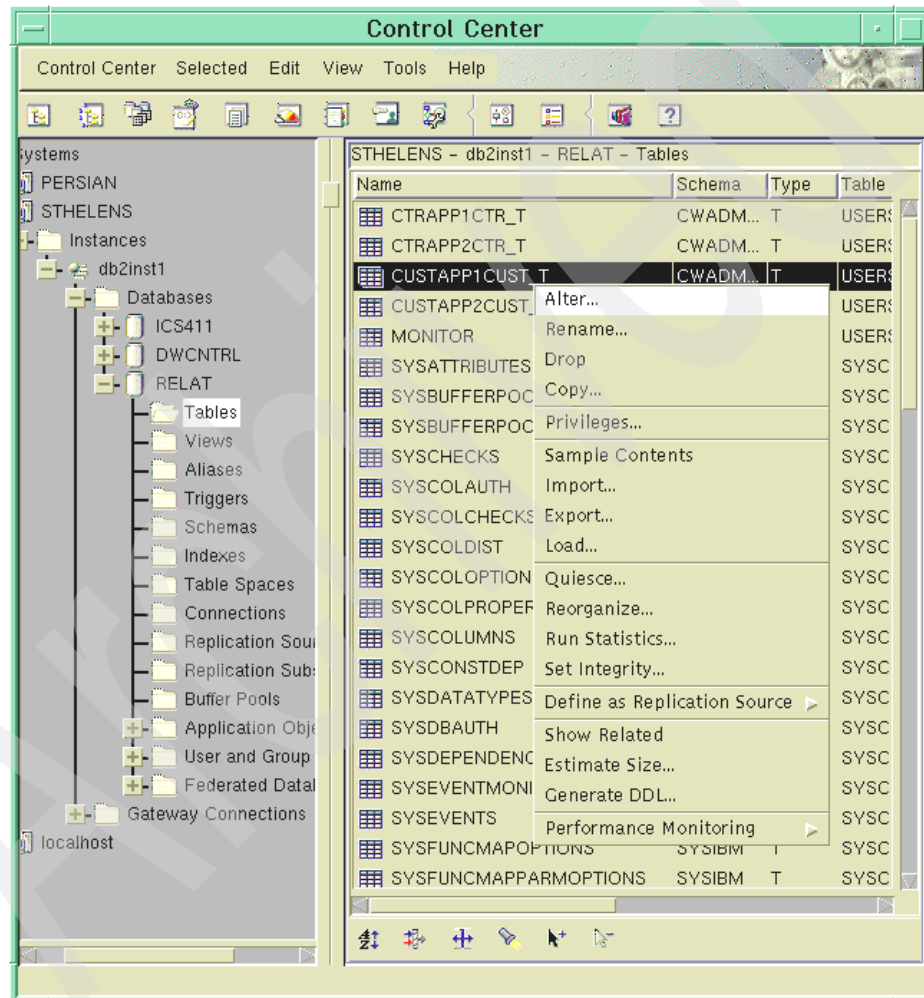


Figure 8-52 Exporting a table

- ii. Enter an output file location and a message file location and click **OK**.

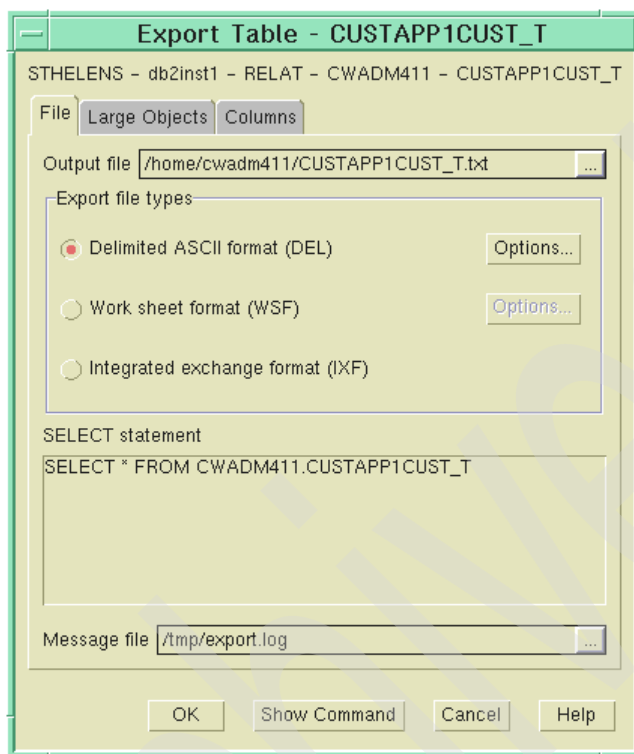


Figure 8-53 Completing the table export

- iii. In the Control Center, right-click the newly created relationship table and choose **Import**. Enter the location of the saved file as the import file and click **OK**.

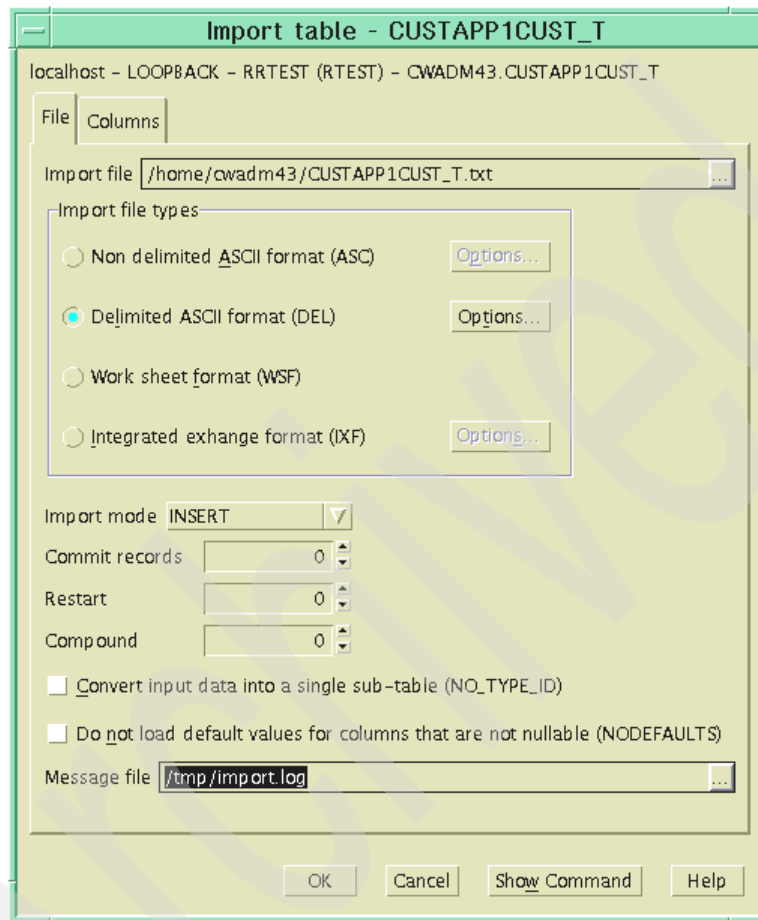


Figure 8-54 Importing the data

- b. Additional steps are needed to set the sequence number for each relationship. Here are the steps to set the sequence number when the relationships are stored in DB2:
  - i. Find the current highest ID for any relationship by examining the INSTANCEID column for any participant table in the relationship.

```
SELECT MAX(INSTANCEID) FROM DB2ADMIN.CUSTAPP1CUST_T
```

In this example, the query has returned the value 41 as the highest INSTANCEID.



- ii. Find the sequence variable name. The following command generates a SQL script that provides information about the relationship table. This information can also be obtained in the Control Center by right-clicking a database table and choosing **Generate DDL**.

```
db2look -dRELAT -tCUSTAPP1CUST_T -a -e -c -oOUTPUT.SQL
```

Examine the *OUTPUT.SQL* file to obtain the sequence variable name. Search for the text CREATE SEQUENCE.

```
CREATE SEQUENCE "DB2ADMIN"."CUSTOMER_1_0_0" AS INTEGER  
MINVALUE 1 MAXVALUE 2147483647 START WITH 1 INCREMENT BY 1  
CACHE 20 NO CYCLE NO ORDER;
```

In the previous example, the sequence variable name is CUSTOMER\_1\_0\_0. The sequence name is derived from the name of the relationship. In this case the relationship name is CUSTOMER.

- iii. Set the sequence variable to be at least one larger than the current number. In this example, the current highest INSTANCEID is 41 so the sequence is set to 42 with the following command.

```
alter sequence Customer_1_0_0 restart with 42
```

- c. Verify that the relationship has been moved successfully to the new database by examining the relationship using Relationship Manager.

### 8.4.3 Post-migration steps

This section describes the steps to take after migration completes and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the previously used password, assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.

2. Right-click the server name in the InterChange component management view and select **Users and roles**. This is shown in Figure 8-55.

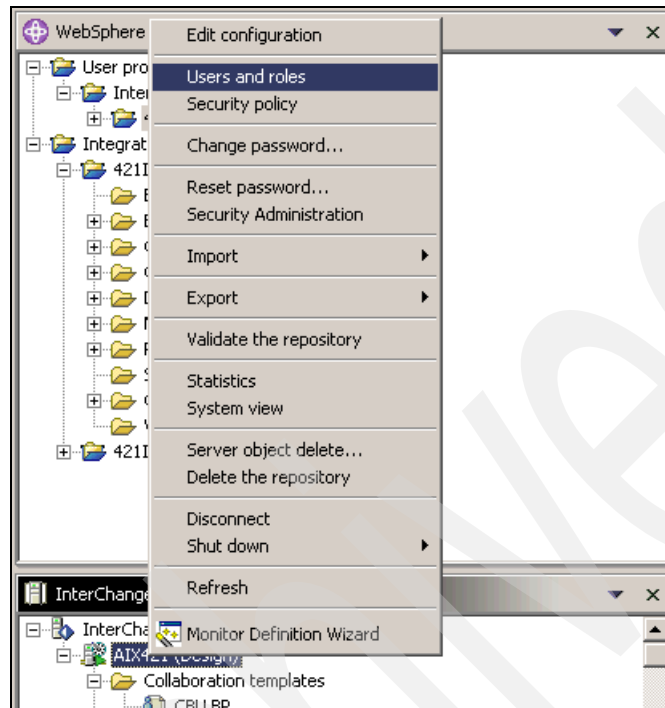


Figure 8-55 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User**.

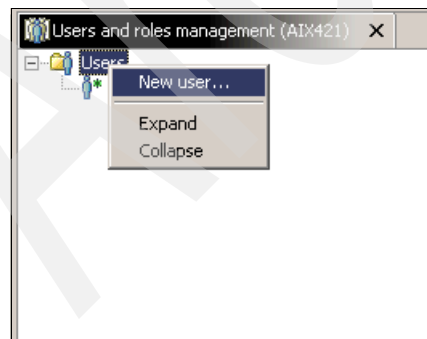


Figure 8-56 Selecting new user

4. The Create user window opens. Enter the name admin and the previous password, plus additional information if needed. Click **OK**.

A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' with 'admin' entered, 'Password' with '\*\*\*\*', 'Confirm password' with '\*\*\*\*', 'Distinguished name' (empty), 'Common name' (empty), and 'Surname' (empty). At the bottom right are 'OK' and 'Cancel' buttons.

Create user

User name  
admin

Password  
\*\*\*\*

Confirm password  
\*\*\*\*

Distinguished name

Common name

Surname

OK Cancel

Figure 8-57 Creating a user

5. The new user is added to the Users and roles management window.

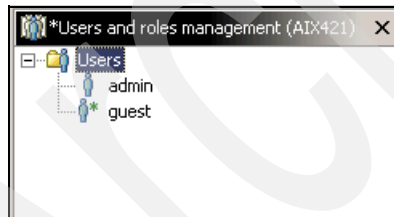


Figure 8-58 The newly created admin user

6. Right-click the newly created **admin** user and select **Add roles**.

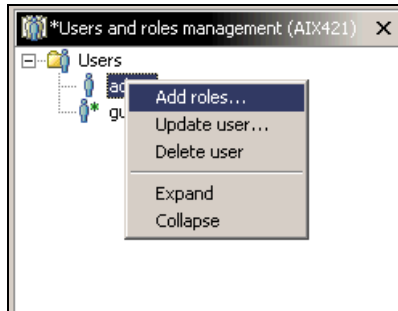


Figure 8-59 Add roles

7. The Role list window is displayed. Select **administrator** and click **OK**.

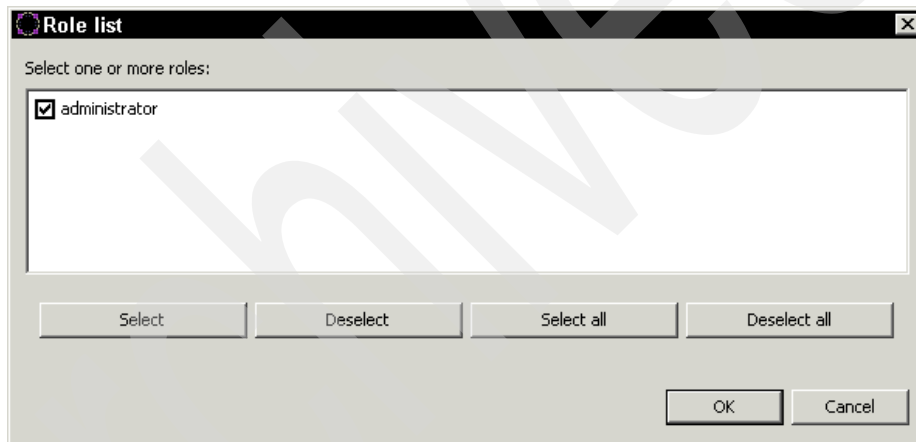


Figure 8-60 Role list

8. Close the Users and roles management window.

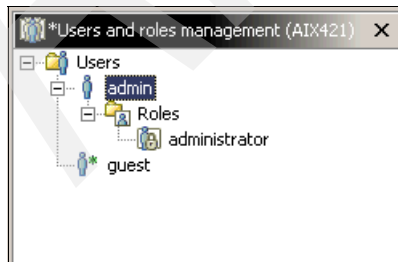


Figure 8-61 User admin with administrator role

9. Right-click the server again in the InterChange Component management view and select **Edit Configuration**. An editor opens to configure the server.
10. Click the **Security-RBAC** tab and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Set **Server start user** to admin.
  - c. Set **Server start password** to the password for the admin user.
  - d. Check all other settings and modify if necessary.
  - e. Save the configuration to the server by clicking the save icon or selecting **File → Save**.

The screenshot shows the 'Security-RBAC' configuration tab for a WebSphere InterChange server. The window is titled '\*AIX421'. It contains several sections:

- RBAC setting:**
  - ☒ Enable RBAC
  - User registry: Repository (dropdown)
  - Server start user: admin
  - Server start password: \*\*\*\*
- Audit setting:**
  - ☒ Enable audit
  - Audit log directory: /opt2/ics431/logs/Audit
  - Audit log frequency: WEEKLY (dropdown)
  - Audit file size (MB): 1
- LDAP setting:**
  - LDAP URL: (empty)
  - User name DN: (empty)
  - Password: (empty)
  - Userbase DN: (empty)
  - User name attribute: (empty)
  - Search criteria: (empty)
  - Maximum number of search returns: (empty)
  - ☐ SSL
- Repository details:**
  - Host name: (empty)
  - Database: RICS421
  - Maximum number of connections: (empty) ☒ Unlimited
  - Max connect retries: 3
  - Login: cwadm421
  - Password: \*\*\*\*\*
  - Port number: 0
  - Connect retry interval: 60 Second(s)

At the bottom, there is a tab bar with the following tabs: General, Database, Tracing levels, Trace and log files, E-mail, CORBA, Miscellaneous, Environment properties, WebSphere MQ, Security - RBAC (selected), and Privacy.

Figure 8-62 Server configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** Adapter agents default to the guest account if RBAC is activated for the WebSphere InterChange Server and no user name/password is provided.

### Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with WebSphere InterChange Server 4.3.0, Version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be

installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions about installing adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the upgraded WebSphere InterChange Server system from development to production, you should perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 8-6 shows the scenarios that were tested in this migration and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

*Table 8-6 Test scenarios results*

Test scenario	Results
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated

**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that WebSphere InterChange Server is re-started in Production mode.

### Uninstall (optional)

This section lists the steps to uninstall the previous version of WebSphere InterChange Server and the related software. The following steps list the order of uninstalling relevant software:

1. Uninstall the VisiBroker ORB using the uninstaller located in the VisiBroker product directory.
2. Uninstall JDK as the root user using **smit**, only if it is not required by other products. For this release, the JDK is typically Java 2 SDK V.1.3.1. Contact the system administrator for further assistance.
3. Uninstall the IBM CrossWorlds InterChange Server using the uninstaller located in the product directory when it is certain that the previous version is no longer needed. After the uninstaller has completed, the product directory can be deleted.





# Migration procedures for WebSphere InterChange Server V4.2.1 to V4.3

This chapter provides instructions for migrating from an IBM WebSphere InterChange Server Version 4.2.1 to IBM WebSphere InterChange Server (WebSphere ICS) Version 4.3 environment.

This chapter provides four major sections for in-place database migration and without in-place database migration for both IBM AIX and Microsoft Windows operating system environments. For a detailed description of the issues related to in-place database migration and without in-place database migration, refer to Chapter 4, “General planning” on page 53.

Each major section is divided into subsections describing pre-migration procedures, migration procedures, and post-migration procedures.

## 9.1 In-place database migration: V4.2.1 to V4.3 on Windows

This section describes an in-place database migration scenario of WebSphere InterChange Server Version 4.2.1 to WebSphere InterChange Server Version 4.3.0. The complete repository from Version 4.2.1 is reused in this scenario. This includes component definitions, failed flows, and relationships.

For the full description of an in-place database migration, refer to Chapter 4, “General planning” on page 53.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 9.1.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system.

#### Important:

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least Version 4.2.1.7.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can only be used with US English character set (umlauts can not be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67.

For more information, view the relevant publications at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

#### Put the environment in a quiescent state

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using Connector Configurator.
2. Resolve transactional status of transactional collaborations using the Flow Manager of the WebSphere Business Integration Toolset. Refer to the Administration Guide for Version 4.2.1 for information about using the Flow Manager. Transactional collaborations that are in transactional status can be migrated but it is recommended that they be resolved in this pre-migration step.
3. Resubmit failed events or discard the events using the Flow Manager of the WebSphere Business Integration Toolset. Refer to the System Administration Guide for Version 4.2.1 for information about using the Flow Manager. Failed flows can be migrated, but it is recommended that they be resolved in this pre-migration step.
4. After all events are processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in System Manager. Ensure that the in-process and pending event counter is 0.

## Backing up and shutting down the components

This section describes the steps needed to shut down and take a backup. It is recommended that you take a backup that enables the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3. Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Administration** → **PDFs** and click **System administration** in the right-side pane.

There are two types of data in the WebSphere InterChange Server system: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is WIN421 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository421.jar. This file is created in the directory where the **repos\_copy** command was run.

```
repos_copy -sWIN421 -uadmin -pnull -oRepository421.jar
```

**Note:** A **repos\_copy** may not be required during an in-place migration; however, it is still a recommended practice for backing up the current definitions of the WebSphere InterChange Server repository.

2. To export existing user projects, connect System Manager to the running WebSphere InterChange Server instance and follow the steps in System Manager:
  - a. Right-click **User Projects** and select **Export Solution**.
  - b. Select all user projects that are to be exported and enter a destination path.

**Restriction:** As we write this book, a product defect causes the Export Solution utility to not work properly. This defect has been reported and assigned tracking number CR60456. Visit the support site to see if a fix for this problem number is available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and choose **Export to Repository File**. User projects must be re-created from the integration component library after the migration.

The *Implementation Guide for WebSphere InterChange Server* has more information about this process. Go to:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Development** → **PDFs** and click **System implementation** in the right-side pane.

3. In System Manager, right-click the WebSphere InterChange Server name and choose **Shutdown Gracefully**.
4. Back up the product directory as indicated by the %CROSSWORLDS% environment variable. By default this is C:\IBM\WebSphere\ICS.
5. Arrange for a database administrator to back up the database. This must be a complete backup, including the schema information and stored procedures.
6. Shut down the (CW) VisiBroker Service.
7. Arrange for a WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.
8. Save the environment settings. In particular, the PATH, CLASSPATH, and CROSSWORLDS environment variables may be modified during the installation of WebSphere InterChange Server 4.3. The current environment settings can be viewed with the **set** command. The following example saves the environment settings to a file called env.txt.

```
set > env.txt
```

### 9.1.2 Migration steps

This section describes the main steps required for the migration, including:

- ▶ Upgrading the related software to the required level
- ▶ Installing and configuring the WebSphere InterChange Server 4.3

- Completing the upgrades of WebSphere InterChange Server components

## Preparing the database system

The database used for the repository with WebSphere InterChange Server 4.2.1 may not meet the minimum database level required by Version 4.3. If upgrading the database to the supported level is necessary, ensure afterward that it has the proper configuration for WebSphere InterChange Server 4.3.

- Check the *IBM WebSphere InterChange Server Installation Guide for Windows* for the proper database settings for V4.3. For example, when using DB2, a change in the database settings is that the `applheapsz` parameter must be set to at least 4096.

Read or download the *IBM WebSphere InterChange Server System Installation Guide for Windows* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **Windows** in the right-side pane.

- Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To test that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run the following command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been successfully created. After the stored procedure has been successfully created, it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 Fix Pack 7 (formerly known as CSD 07). If failed flows are being migrating that involve MQ Delivery Transport, the queue messages must be preserved when upgrading WebSphere MQ. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Prepare the Web-based System Monitor

If WebSphere Application Server is at Version 5.0.2.4 or 5.1 before running the WebSphere InterChange Server installer, the installer program automatically installs and configures System Monitor and uninstalls the previous version of the ICSMonitor application. This feature is available on Windows only when WebSphere Application Server is located on the same machine as the WebSphere InterChange Server. The WebSphere Application Server must at the supported level before installing WebSphere InterChange Server. The System Monitor can be installed after installing WebSphere InterChange Server.

## Uninstall software

This section describes the uninstall procedures for the WebSphere InterChange Server and related software.

1. Uninstall WebSphere InterChange Server and WebSphere Business Integration Adapters Windows services (if any) using the following command:

```
cwservice -xr -sServiceName
```

2. Uninstall the VisiBroker Object Request Broker using the Add/Remove Programs utility.
3. Uninstall the Java Development Kit (JDK) using the Add/Remove Programs utility, but only if it is not required by other products. For this release, the JDK is typically Java 2 SDK V.1.3.1.
4. Uninstall WebSphere InterChange Server. During the uninstall process, all files can be removed safely because of the product directory backup you made. After the uninstaller has completed, the product directory can be deleted.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, "Installation of WebSphere InterChange Server 4.3" on page 99 for details.

**Note:** During the installation process, use the same WebSphere InterChange Server name as in the previous version. Remember, the WebSphere InterChange Server name must not contain dots (.).

If a decision was made to have the installer automatically install System Monitor, confirm that the System Monitor has been installed successfully. Examine the file %CROSSWORLDS%/WBSM/configlog and search for the message Application ICSMonitor installed successfully.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is prompted to configure the WebSphere InterChange Server, as we demonstrate in the following pictures and examples. The IBM WebSphere InterChange Server Configuration wizard can be run after the installation has finished by invoking:

```
%CROSSWORLDS%\bin\ICConfig.bat
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all configuration settings must be entered in the Options.txt file that is delivered with the installer. For detailed information, refer to the Installation Guide for Windows.



The following figures show the configuration values used by the redbook team during migration:

1. Figure 9-1 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

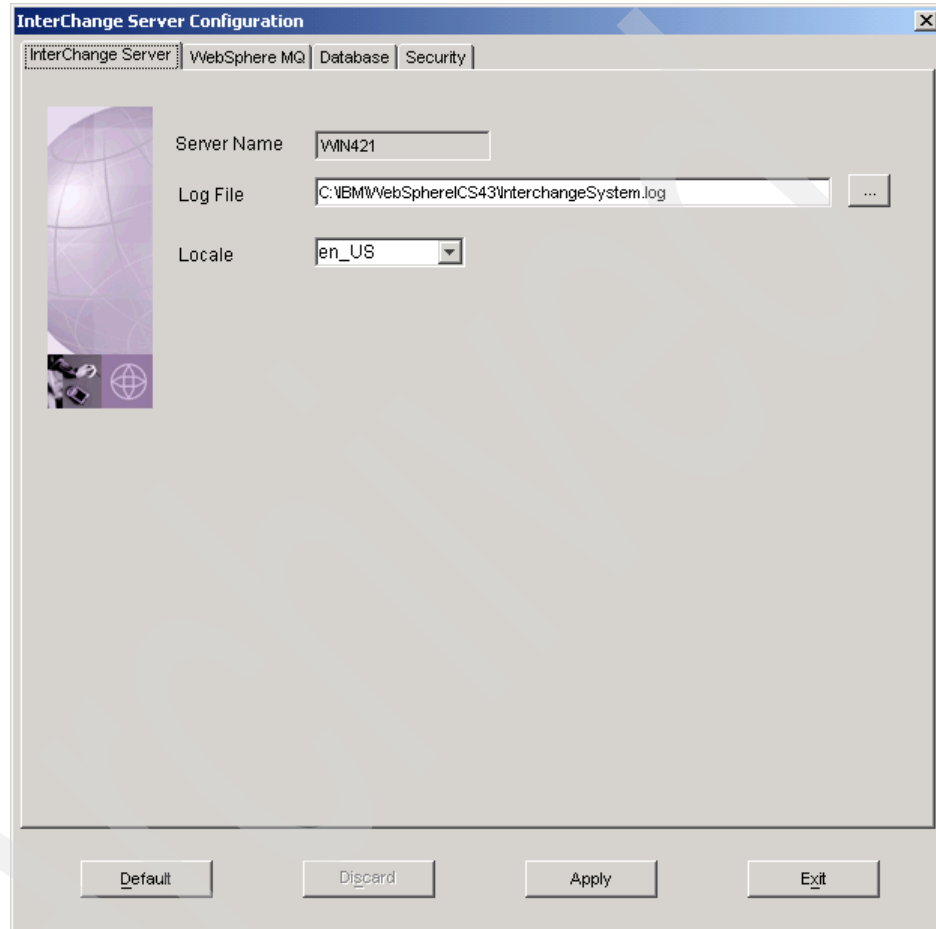
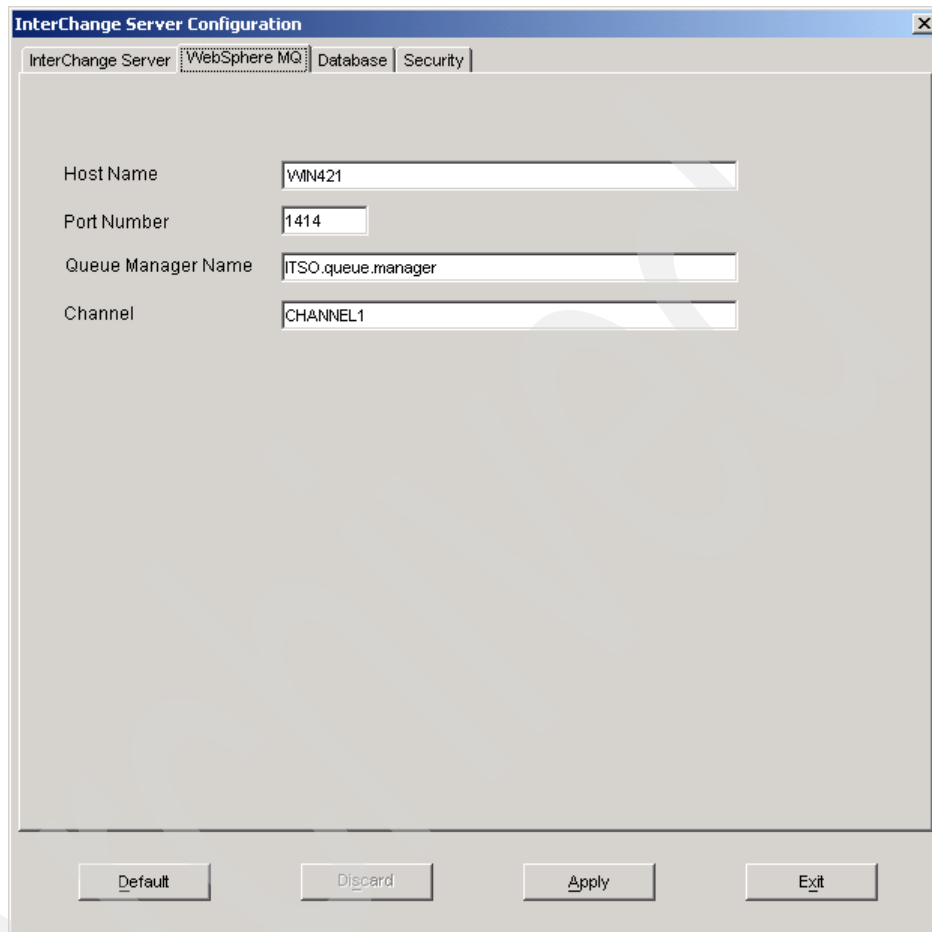


Figure 9-1 InterChange Server configuration: InterChange Server tab

2. Figure 9-2 shows the WebSphere MQ configuration tab. Make sure the settings for WebSphere InterChange Server 4.2.1 are entered here.



The image shows a dialog box titled "InterChange Server Configuration". It has four tabs: "InterChange Server", "WebSphere MQ", "Database", and "Security". The "WebSphere MQ" tab is selected. Inside the dialog, there are four text input fields with labels to their left: "Host Name" with the value "WIN421", "Port Number" with the value "1414", "Queue Manager Name" with the value "ITSO.queue.manager", and "Channel" with the value "CHANNEL1". At the bottom of the dialog, there are four buttons: "Default", "Discard", "Apply", and "Exit".

Field	Value
Host Name	WIN421
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 9-2 InterChange Server configuration: WebSphere MQ tab

3. The third tab is for database configuration. Enter the connection information from WebSphere InterChange Server 4.2.1 and apply the following settings:
  - Max Connections: unlimited, if possible. For the purposes of this book, the Max Connections parameter was set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Database' tab selected. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab contains several sections for configuring database connectivity and monitoring.

**Database Connectivity**

- Database Driver: DB2 (selected in a dropdown)
- Max Connections: 75 (with an 'Unlimited' checkbox that is unchecked)
- Max Pools: 50
- Idle Timeout: 2

**Event Management**

- Host Name: (empty text field)
- Login: db2admin
- Database: cwrepos
- Password: (masked with asterisks)
- Max Connections: (empty text field) with an 'Unlimited' checkbox that is checked
- Port Number: (empty text field)
- Change button

**Transactions**

- Host Name: (empty text field)
- Login: db2admin
- Database: cwrepos
- Password: (masked with asterisks)
- Max Connections: (empty text field) with an 'Unlimited' checkbox that is checked
- Port Number: (empty text field)
- Change button

**Repository**

- Host Name: (empty text field)
- Login: db2admin
- Database: cwrepos
- Password: (masked with asterisks)
- Max Connections: (empty text field) with an 'Unlimited' checkbox that is checked
- Port Number: (empty text field)
- Change button

**Flow Monitoring**

- Host Name: (empty text field)
- Login: db2admin
- Database: cwrepos
- Schema Name: (empty text field)
- Password: (masked with asterisks)
- Max Connections: (empty text field) with an 'Unlimited' checkbox that is checked
- Port Number: (empty text field)
- Change button

At the bottom of the dialog are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 9-3 InterChange Server configuration: Database tab

4. The fourth tab is for security configuration (Figure 9-4). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information. For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the database and connection information from the Database tab for the Security database.

The screenshot shows the 'InterChange Server Configuration' window with the 'Security' tab selected. The 'User Registry Type' is set to 'Repository'. Under the 'Repository' section, the 'Host Name' is empty, 'Login' is 'db2admin', 'Database' is 'CWREPOS', 'Password' is masked with asterisks, and 'Max Connections' is set to 'Unlimited'. A 'Change' button is next to the password field. At the bottom are 'Default', 'Discard', 'Apply', and 'Exit' buttons.

InterChange Server Configuration				
InterChange Server		WebSphere MQ	Database	Security
User Registry Type				
User Registry		Repository		
Repository				
Host Name		Login	db2admin	
Database	CWREPOS	Password	*****	Change
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number	

Figure 9-4 InterChange Server configuration: Security tab

To save the settings, click **Apply**. To exit the configurator, click **Exit**.

If the WebSphere InterChange Server is installed as a Windows Service, restart Windows as instructed.

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMS and collaboration directories from WebSphere InterChange Server Version 4.2.1 to the new installation directory for Version 4.3.0. Be sure to include all subdirectories.
2. Apply any custom configuration settings to `$CROSSWORLDS/InterchangeSystem.cfg`.

For example, to disable `DEADLOCK_DETECTOR_CHECK` for collaborations, edit the config file code shown in Example 9-1 using a suitable editor for XML.

### Example 9-1 Disable `DEADLOCK_DETECTOR_CHECK`

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>>false</tns:reposController>
<tns:reposAgent>>false</tns:reposAgent>
<tns:localConfig>>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change `true` to `false` in the line starting with `<tns:value`.

3. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, these must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
4. Include any external code and .jar files in the `CLASSPATH` environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in `%CROSSWORLDS\bin\start_server.bat`. The external classes are first added to a new variable `EXT_LIB`. Then the `EXT_LIB` is appended to the `CWCLASSES` variable. Refer to the following example snippet from `start_server.bat`:

```
REM external classes added here
set EXT_LIB=C:\Sample.jar
CWCLASSES="%CROSSWORLDS%\lib\crossworlds.jar;%CROSSWORLDS%\lib\cworio
n.jar;%CROSSWORLDS%\lib\CodeGeneration.jar;%EXT_LIB%
```

5. Modify and verify the environment variables as documented. Confirm that the following three environment variables are set properly:

<b>CROSSWORLDS</b>	Contains the location of the 4.3.0 installation directory.
<b>MQ_LIB</b>	Contains the location of WebSphere MQ Java /lib directory.
<b>PATH</b>	Contains the /bin directory of IBM Java Development Kit 1.4.2. Ensure that this bin directory precedes any other installed JDKs. Confirm that the compiler is installed by invoking <b>javac</b> from the command line. The usage instructions for <b>javac</b> should be returned.

### Starting the new server

When the WebSphere InterChange Server is started for the first time after the installation, the WebSphere InterChange Server repository tables are upgraded or created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database server
2. Run the PersistentNameServer.bat file, located in %CROSSWORLDS%\bin\.
3. Start the WebSphere InterChange Server in design mode: Edit the Properties of the IBM WebSphere InterChange Server shortcut in the Start menu. Add the -design option to the Target field. Run the shortcut after modification.

WebSphere InterChange Server upgrades existing tables, creates new tables for new features, and logs messages to the InterChangeSystem.log file located in the product directory. The message Interchange Server is Ready indicates that WebSphere InterChange Server has started successfully.

### Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. To complete this task.

1. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, import the component libraries using exported integration component library files:

1. Create a new integration component library.
2. Right-click the newly created integration component library and select **Import Repository from File**. Choose the repository file created during the backup procedure. Repeat these steps if there are multiple libraries to import.
3. The *Implementation Guide for WebSphere InterChange Server* contains further information about this process. View or download it at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left-side pane, expand **Interchange Server and Toolset** → **Development** → **PDFs**, and click **System Implementation** in the right-side pane.

4. Include any custom .jar files in the CLASSPATH environment variable for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH setting for the toolset (Figure 9-5). Add the location of any custom .jar files to this list.

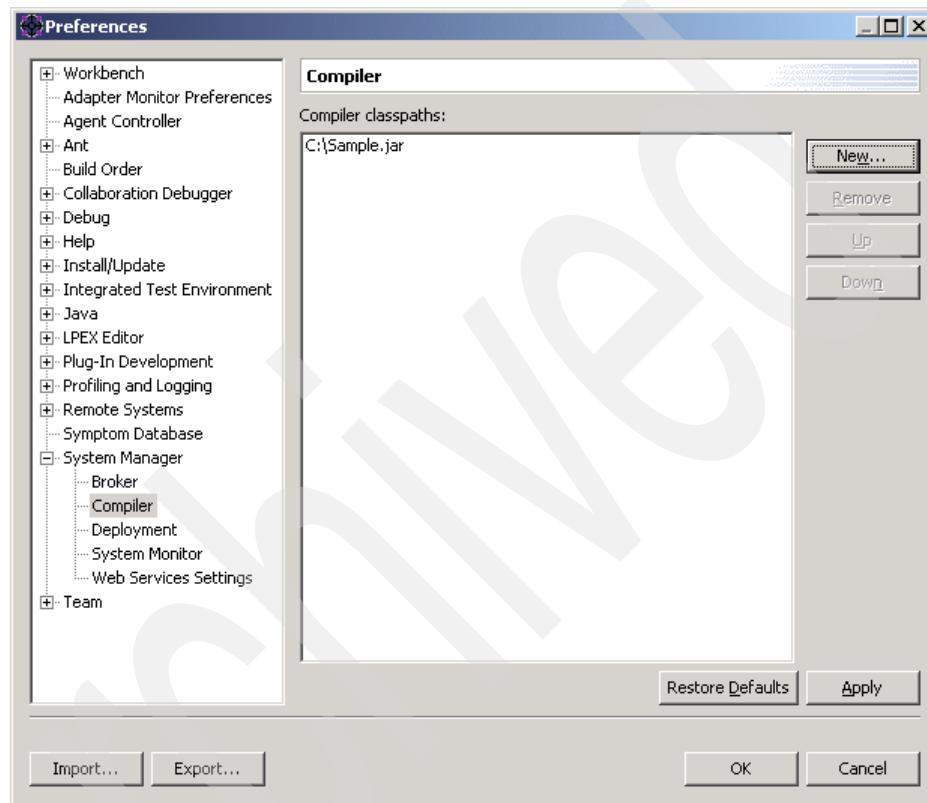


Figure 9-5 Setting classpath for the Toolset



5. Set the compiler compliance to Java 1.4: Navigate to **Java** → **Compiler**. Under the Compliance and classfiles tab, change the compiler compliance level to **1.4** (Figure 9-6). If a dialog box indicating that the compiler settings have changed is displayed, choose **No** to *not* rebuild files at this time.

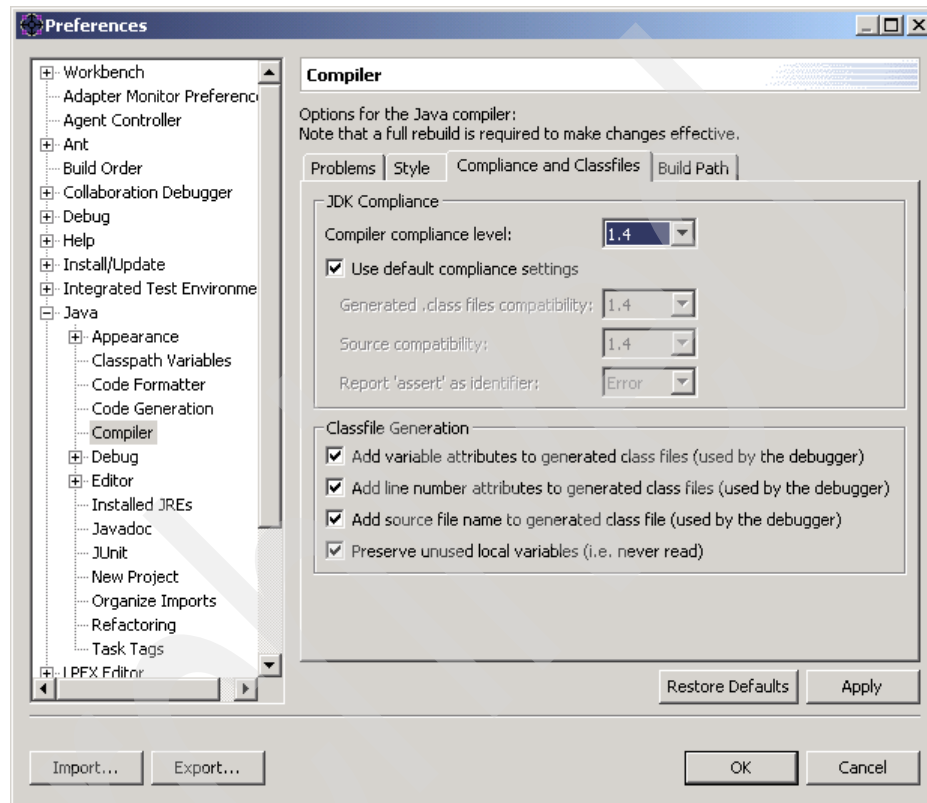


Figure 9-6 Setting the compiler compliance level

6. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the use of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in the previous step.
7. Compile all of the maps and templates in the integration component library and examine the console for any error messages.
8. Reconfigure any of the other components that require modification:
  - a. Reconfigure or create any connection pools and relationship definitions.

- b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more details.
  - c. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions, examine the Port and Triggering Events tab. For the BO Type that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to **<None>** for all verbs.
  - d. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
9. Create a new user project and populate the component from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for details.

**Note:** To make use of all administrative features for collaborations and maps, the maps and templates must be deleted and redeployed. Typically, each component must be placed in an inactive state before it can be overwritten. To simplify the process and avoid unresolved dependencies, the whole repository is deleted in step 10 and redeployed in step 11. Additionally, recompilation ensures that the code is optimized and Java 1.4 compliant.

However, redeploying the maps and templates is not required, because the Java Runtime Environment is backward-compatible and poses no problems executing map and template class files compiled in WebSphere InterChange Server 4.2.1.

If a decision is made to reuse the previous maps and collaborations, only deploy the components that have been updated.

10. In the System Manager, delete the repository by right-clicking the WebSphere InterChange Server name in the InterChange Server component management view and select **Delete the repository**.
11. Deploy the user project:
  - a. Right-click the user project and select **Deploy user project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This re-creates the stored procedures and does not overwrite existing data.

### 9.1.3 Post-migration steps

This section describes the steps that are taken after migration and after the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the previous password assigned to the administrator role. The following steps and figures show how to do this:

1. In the System Manager, connect to the WebSphere InterChange Server.
2. Right-click the server name in the InterChange Server component management view and select **Users and roles** (Figure 9-7).

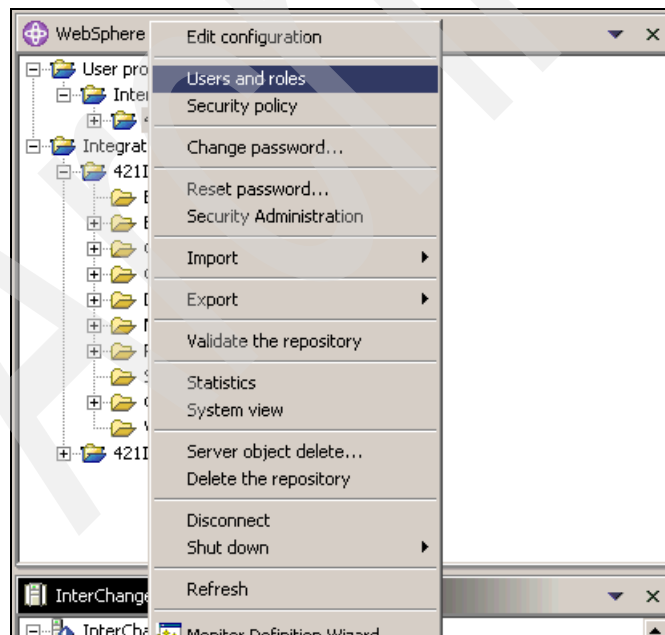


Figure 9-7 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User** (Figure 9-8).

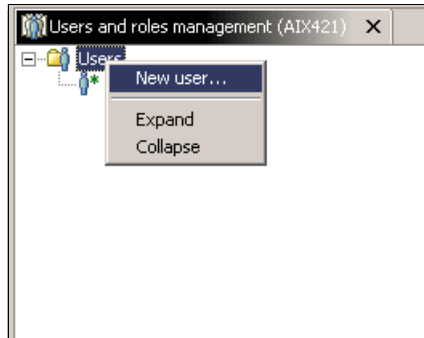


Figure 9-8 Select New user

4. In the Create user window (Figure 9-9), enter the name admin and the previously used password. Add additional information if needed and click **OK**.

A screenshot of a "Create user" dialog box. It has a title bar with a gear icon and the text "Create user". The dialog contains several text input fields: "User name" (containing "admin"), "Password" (containing "\*\*\*\*"), "Confirm password" (containing "\*\*\*\*"), "Distinguished name" (empty), "Common name" (empty), and "Surname" (empty). At the bottom right, there are two buttons: "OK" and "Cancel".

Figure 9-9 Creating a user

5. The new user is added to the User and Roles management window (Figure 9-10).

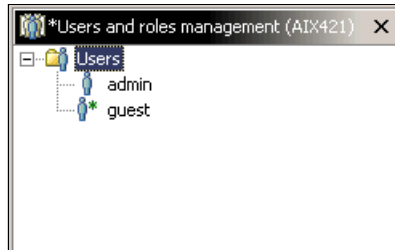


Figure 9-10 The newly created admin user

6. Right-click the new admin user and select **Add roles** (Figure 9-11).

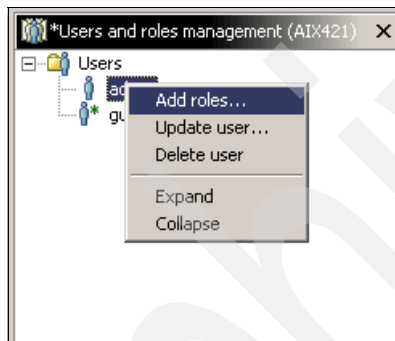


Figure 9-11 Add roles

7. The Role list window is displayed (Figure 9-12).

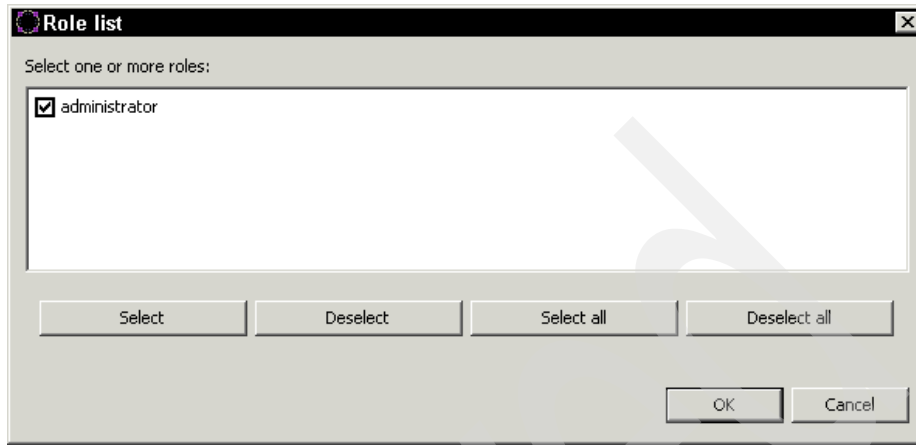


Figure 9-12 Role list

8. Select **administrator** and click **OK**. Close the Users and roles management window.

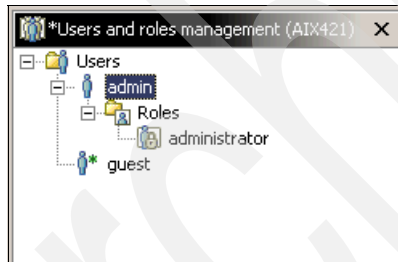


Figure 9-13 User admin with administrator role

9. Right-click the server again and choose **Edit Configuration**. This opens an editor for configuring the server.

10. Click the **Security-RBAC** tab (Figure 9-14) and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Enter admin for **Server start user**.
  - c. For **Server start password** enter the password for the admin user.
  - d. Check all other settings in the tab and modify if needed.
  - e. Save the configuration to the server by clicking the save icon or selecting **File** → **Save**.

Figure 9-14 Server Configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** Adapter agents default to the guest account if the RBAC is activated for WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3, a WebSphere Business Integration Adapter V2.6 must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter and includes the Visual Test Connector. For more about installing adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the migrated InterChange Server system from development to production, it is recommended that you perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 9-1 shows the scenarios that were tested in this migration and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be migrated successfully if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

*Table 9-1 Test scenarios results*

Test Scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated
Web-based System Monitor	Successfully migrated

A failed flow is considered to be migrated successfully if the event can be resubmitted and completed after migration. Table 9-2 on page 247 shows the results of the migration of failed flows.



Table 9-2 Migrated failed flows results

Failed flows tested	Results
Failed flow for IDL	Successfully migrated
Failed flow for MQ	Successfully migrated
Failed flow for JMS	Successfully migrated
Transactional Collaboration in transactional status (compensation call failed)	Successfully migrated
Transactional Collaboration failed flow (compensation call successful)	Successfully migrated
LLBP waiting asynchronous inbound call	Successfully migrated
LLBP failed flow	Successfully migrated
Collaboration group failed at Destination 1	Successfully migrated
Collaboration group failed at Destination 2	Successfully migrated

**Note:** After all system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is restarted in Production mode.

## 9.2 Without in-place database migration of V4.2.1 to V4.3 on Windows

This section describes a without in-place database migration scenario of WebSphere InterChange Server Version 4.2.1 to Version 4.3.0. WebSphere InterChange Server 4.3.0 is started with a new and empty repository. Repository content (component definitions) is exported from the original V4.2.1 and deployed in the Version 4.3.0 environment.

For the full description of a without in-place database migration refer to Chapter 4, “General planning” on page 53.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

## 9.2.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system.

### Important:

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.2.1.7.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

### Put the environment in a quiescent state

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All the collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator.
2. Resolve transactional status of transactional collaborations using the Flow Manager tool of WebSphere Business Integration Toolset. Refer to the *Administration Guide* Version 4.2.1 for information about using the Flow Manager. All the status information is lost during the migration.
3. Resubmit failed events or discard the events using the Flow Manager tool of WebSphere Business Integration Toolset. Refer to the *Administration Guide* Version 4.2.1 for information about using the Flow Manager. All failed flows are lost after migration.
4. When all events are processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in System Manager. Make sure the in-process and pending event counter is 0.

## Backing up and shutting down components

This section describes the steps for shutting down and taking a backup. It is a recommended practice to take a backup that will enable the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3.0.

Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Administration** → **PDFs** and click **System administration** in the right-side pane.

The WebSphere InterChange Server system has two types of data: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current WebSphere InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is WIN421 and has the default login of `admin` and the password of `null`. The following **repos\_copy** command creates a backup of the repository objects in a file called `Repository421.txt`. This file is created in the directory where the **repos\_copy** command is run.

```
repos_copy -sWIN421 -uadmin -pnull -oRepository421.jar
```

2. To export existing user projects, connect System Manager running on a connected Windows machine to the existing InterChange Server instance and take the following steps in System Manager:
  - a. Right-click **User Projects** and select **Export Solution**.
  - b. Select all user projects that to be exported and enter a destination path.

**Restriction:** As we write this book, a product defect prevents the Export Solution utility from working properly. This defect has been reported and assigned tracking number CR60456. Check the support site below to see whether a fix for this problem number is available.

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and select **Export to Repository File**. The *Implementation Guide for WebSphere InterChange Server* has more information about this process. The user projects must be re-created from the integration component library after the migration.

3. Shut down the WebSphere InterChange Server. In the System Manager, right-click the WebSphere InterChange Server name and choose **Shutdown Gracefully**.
4. Back up the product directory indicated by the %CROSSWORLDS% environment variable. By default this is C:\IBM\WebSphere\ICS.
5. Arrange with the database administrator to back up the database. This must be a complete backup, including schema information and stored procedures.
6. Arrange for a WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.
7. Clear WebSphere MQ queues of any outstanding messages if the same queue manager is reused for Version 4.3.0. Old messages in the queues could be seen as new events, which could cause data inconsistencies. Refer to the WebSphere MQ documentation for more information.
8. Save the environment settings. In particular, the PATH, CLASSPATH, and CROSSWORLDS environment variables may be modified during the installation of WebSphere InterChange Server 4.3.0. The current environment settings can be viewed with the **set** command. The following example saves the environment settings to a file called env.txt:

```
set > env.txt
```

## 9.2.2 Migration steps

This section describes the main steps required for the migration including:

- ▶ Upgrading related software to the required level.
- ▶ Installing and configuring WebSphere InterChange Server 4.3.0.
- ▶ Completing the upgrades of WebSphere InterChange Server components.

### Preparing the database system

Create an empty database for the V4.3 repository following the steps in the *IBM WebSphere InterChange Server System Installation Guide*, and ensure that the database has the proper configuration for WebSphere InterChange Server 4.3.

- ▶ Check the installation guide for the proper database settings for 4.3. For example, when using DB2, one change in the database settings is that the `applsheap` parameter must be set to at least 4096.
- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

### Upgrade WebSphere MQ

WebSphere MQ must be at Version 5.3.0.2 Fix Pack 7 (formerly known as CSD 07). Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Uninstall software

This section describes the uninstallation procedures of the WebSphere InterChange Server and related software:

1. Uninstall WebSphere InterChange Server and WebSphere Business Integration Adapters Windows Services (if any) using this command:  

```
cwservice -xr -sServiceName
```
2. Uninstall the Object Request Broker (VisiBroker) using the Add/Remove Programs utility.
3. Uninstall the Java Development Kit (JDK) using Add/Remove Programs in the Windows Control Panel—only if it is not required by other products. For this release, the JDK is typically Java 2 SDK 1.3.1.
4. Uninstall WebSphere InterChange Server. During the uninstall process, all files can be removed safely because of the backup of the product directory that was taken earlier. After the uninstaller has completed, the product directory can be deleted.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 for details.

**Note:** During the installation process, the WebSphere InterChange Server name may be changed. If so, the adapters and queues may have to be configured. You should use the same WebSphere InterChange Server name if possible.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following steps explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
%CROSSWORLDS%\bin\ICSSConfig.bat
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all configuration settings must be entered in the file Options.txt, which is delivered with the installer. For detailed information, refer to the Installation Guide for Windows.

The following figures show the configuration values used by the redbook team during migration:

1. Figure 9-15 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

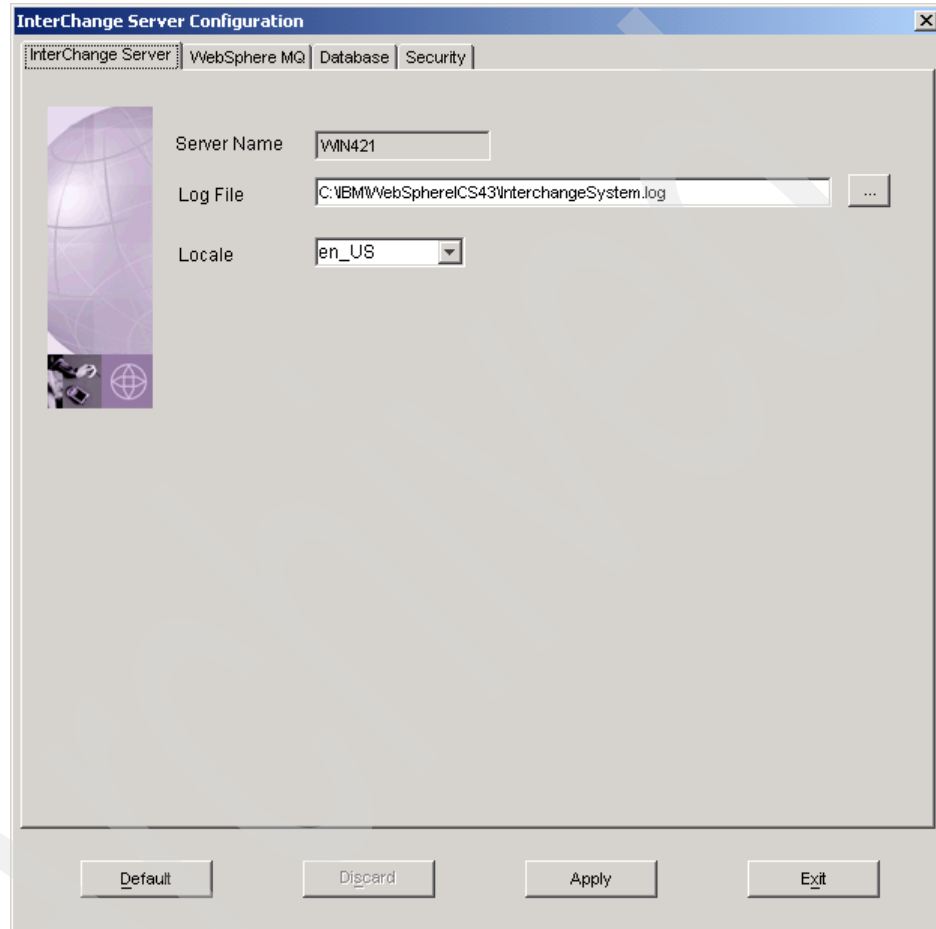
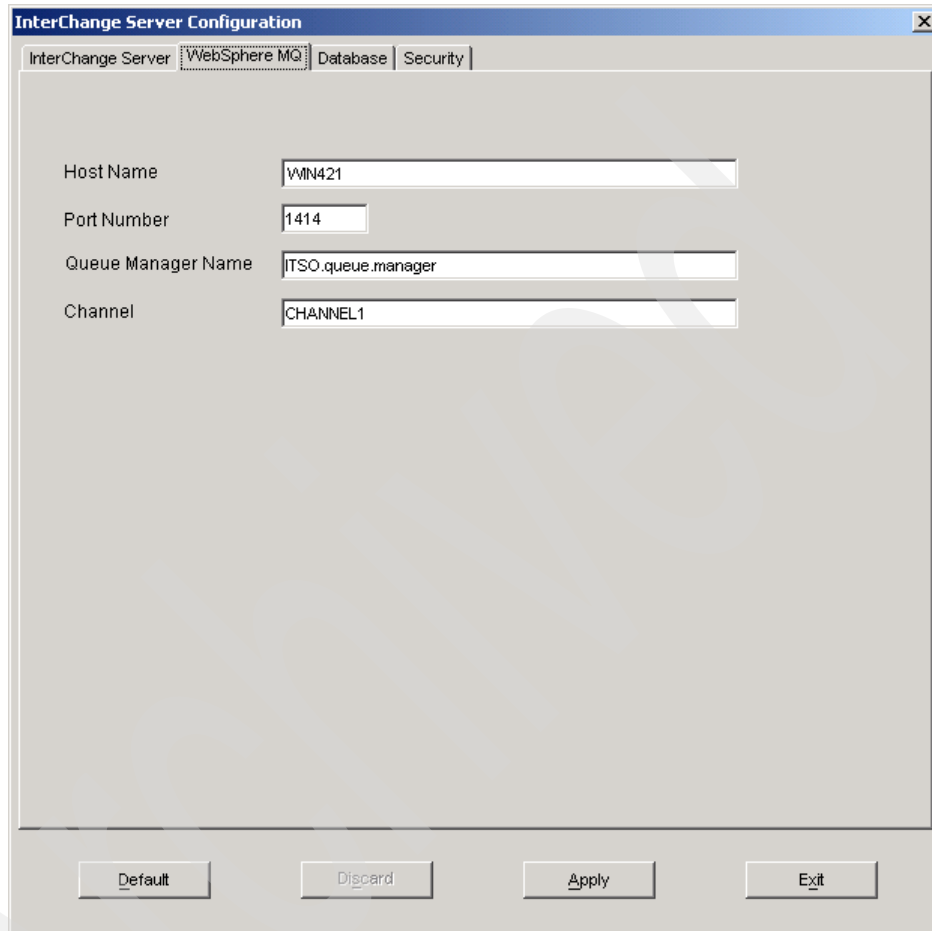


Figure 9-15 InterChange Server Configuration: InterChange Server tab

2. Figure 9-16 shows the WebSphere MQ configuration tab. Ensure that the settings for WebSphere MQ are entered here.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'WebSphere MQ' tab. The dialog box has a title bar with the text 'InterChange Server Configuration' and a close button. Below the title bar are four tabs: 'InterChange Server', 'WebSphere MQ' (which is selected), 'Database', and 'Security'. The main area of the dialog box contains four labeled text input fields: 'Host Name' with the value 'WIN421', 'Port Number' with the value '1414', 'Queue Manager Name' with the value 'ITSO.queue.manager', and 'Channel' with the value 'CHANNEL1'. At the bottom of the dialog box are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Field	Value
Host Name	WIN421
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 9-16 InterChange Server Configuration: WebSphere MQ tab



3. The third tab is for database configuration (Figure 9-17). Make sure to enter the connection information for the empty database created in the previous step and apply the following settings:
  - Max Connections: unlimited, if possible. For the purposes of this book, the Max Connections parameter was set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Database' tab selected. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab contains several sections for configuring database connectivity and monitoring.

**Database Connectivity**

- Database Driver: DB2 (dropdown)
- Max Connections: 75 (spin box) with an 'Unlimited' checkbox.
- Max Pools: 50 (spin box).
- Idle Timeout: 2 (spin box).

**Event Management**

- Host Name: (empty text box)
- Database: cwrepos (text box)
- Max Connections: (empty text box) with an 'Unlimited' checkbox.
- Login: db2admin (text box)
- Password: (masked with asterisks) with a 'Change' button.
- Port Number: (empty text box)

**Transactions**

- Host Name: (empty text box)
- Database: cwrepos (text box)
- Max Connections: (empty text box) with an 'Unlimited' checkbox.
- Login: db2admin (text box)
- Password: (masked with asterisks) with a 'Change' button.
- Port Number: (empty text box)

**Repository**

- Host Name: (empty text box)
- Database: cwrepos (text box)
- Max Connections: (empty text box) with an 'Unlimited' checkbox.
- Login: db2admin (text box)
- Password: (masked with asterisks) with a 'Change' button.
- Port Number: (empty text box)

**Flow Monitoring**

- Host Name: (empty text box)
- Database: cwrepos (text box)
- Schema Name: (empty text box)
- Max Connections: (empty text box) with an 'Unlimited' checkbox.
- Login: db2admin (text box)
- Password: (masked with asterisks) with a 'Change' button.
- Port Number: (empty text box)

At the bottom of the dialog are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 9-17 InterChange Server Configuration: Database tab

4. The fourth tab is for security configuration (Figure 9-4 on page 234). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Security' tab selected. The 'User Registry Type' is set to 'Repository'. The 'Repository' section contains the following fields: 'Host Name' (empty), 'Login' (clb2admin), 'Database' (cwrepos), 'Password' (masked with asterisks), 'Max Connections' (empty), 'Unlimited' (checked), and 'Port Number' (empty). A 'Change' button is next to the password field. At the bottom of the dialog are buttons for 'Default', 'Discard', 'Apply', and 'Exit'.

User Registry Type	
User Registry	Repository

Repository	
Host Name	
Login	clb2admin
Database	cwrepos
Password	*****
Max Connections	<input checked="" type="checkbox"/> Unlimited
Port Number	

Figure 9-18 InterChange Server Configuration: Security tab

If the WebSphere InterChange Server is installed as a Windows Service, restart Windows as instructed.

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.

For example, to disable DEADLOCK\_DETECTOR\_CHECK for collaborations, edit the config file code shown in Example 9-2 using a suitable editor for XML.

### *Example 9-2 Disable DEADLOCK\_DETECTOR\_CHECK*

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>false</tns:reposController>
<tns:reposAgent>false</tns:reposAgent>
<tns:localConfig>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change true to false in the line starting with <tns:value.

2. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, these must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
3. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in %CROSSWORLDS\bin\start\_server.bat. The external classes are first added to a new variable EXT\_LIB. Then the EXT\_LIB is appended to the CWCLASSES variable. Refer to the following example from start\_server.bat:

```
REM external classes added here
set EXT_LIB=C:\Sample.jar
CWCLASSES="%CROSSWORLDS%\lib\crossworlds.jar;%CROSSWORLDS%\lib\cworio
n.jar;%CROSSWORLDS%\lib\CodeGeneration.jar;%EXT_LIB%
```

4. Modify and verify the environment variables as documented. Confirm that the following three environment variables are set properly:

**CROSSWORLDS**      Contains the location of the 4.3.0 installation directory.

<b>MQ_LIB</b>	Contains the location of WebSphere MQ Java /lib directory.
<b>PATH</b>	Contains the /bin directory of IBM Java Development Kit 1.4.2. Ensure that this bin directory precedes any other installed JDKs. Confirm that the compiler is installed by invoking <b>javac</b> from the command line. The usage instructions for <b>javac</b> should be returned.

## Starting the new server

When WebSphere InterChange Server is started for the first time after the installation, the WebSphere InterChange Server repository tables are created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database server
2. Run the PeristentNameServer.bat file located in %CROSSWORLDS%\bin\.
3. Start the WebSphere InterChange Server in design mode by editing the **Properties** of the IBM WebSphere InterChange Server shortcut in the **Start** menu. Add the option -design to the **Target** field. Run the shortcut after modification.

WebSphere InterChange Server logs messages to the InterChangeSystem.log file located in the product directory. The message Interchange Server is Ready indicates that the WebSphere InterChange Server has started successfully.

## Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Import the previous repository into the tooling.
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, import the component libraries using exported integration component library files.

- ▶ Create a new integration component library.
- ▶ Right-click the newly created integration component library and select **Import Repository from File**. Choose the repository file that was created in the backup procedure. Repeat these steps if there are multiple libraries to import. The *Implementation Guide for WebSphere InterChange Server* has more information about this process.

2. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH settings for the Toolset. Add the location of the custom .jar files to this list.

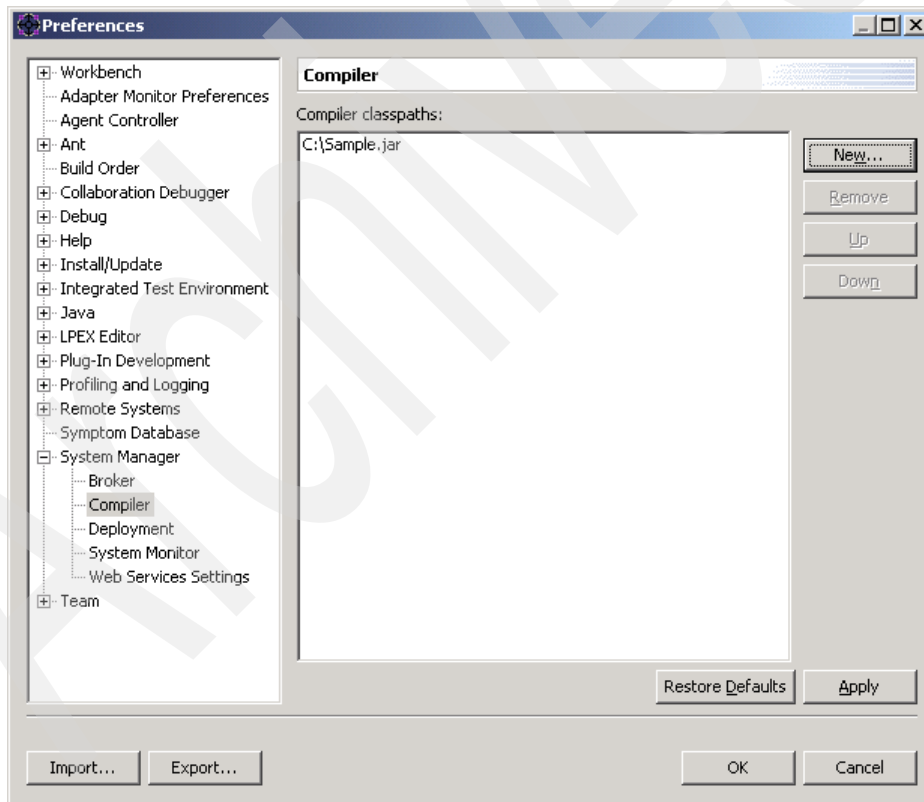


Figure 9-19 Setting the classpath for the Toolset

3. Set the compliance of the compiler to Java 1.4. Click **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag change the Compiler compliance level to **1.4**. A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to not rebuild files at this time.

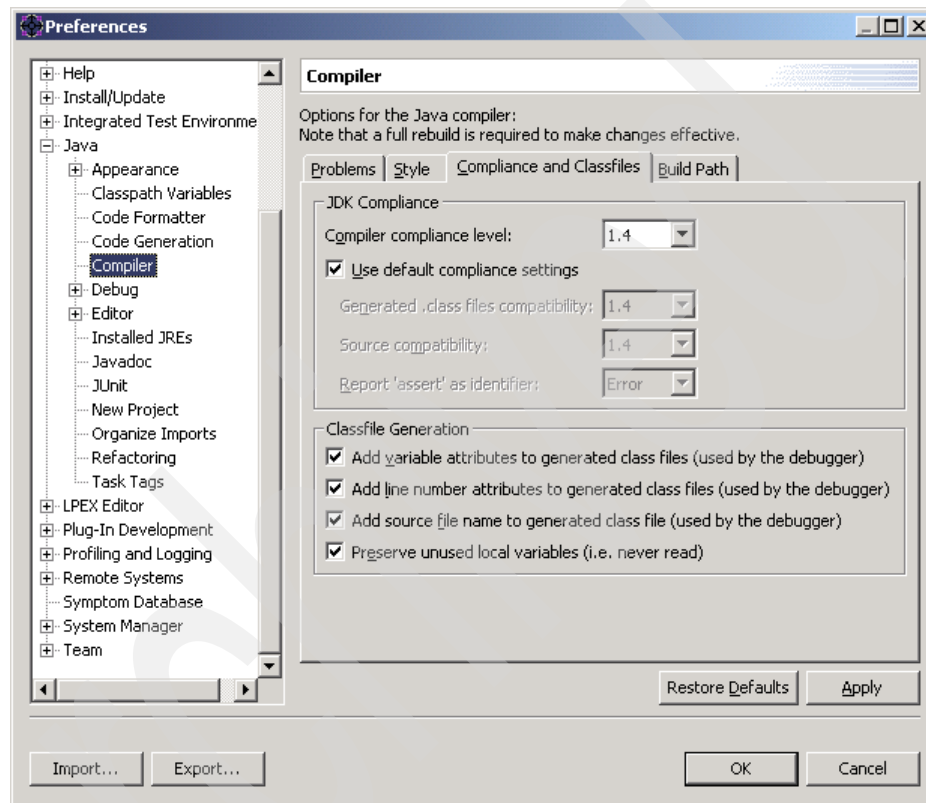


Figure 9-20 Setting the compiler compliance level

4. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the usage of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in a previous step.
5. Compile all maps and templates in the integration component library and examine the console for any error messages.

6. Reconfigure any of the other components that require modification:
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more details.
  - c. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions, examine the Port and Triggering Events tab. For the **BO Type** that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to <None> for all verbs.
  - d. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.

**Note:** If Import Solution was not used, create a new user project and populate the user project with the components from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for further details.

7. Deploy the user project:
  - a. Right-click the user project and select **Deploy User Project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. If the location of the relationship tables has changed, select **Create Schema**.
  - f. Click **Finish** to start the deployment.
8. Restart the WebSphere InterChange Server if prompted.
9. Completing the relationship migration:

If a new relationship location is used, the relationship data from the Version 4.2.1 relationship tables can be migrated in the following way:

  - a. For both static and dynamic relationships, copy the data from each relationship table to the new relationship table. Consult with a database administrator for more details. The steps to complete this process using DB2 are:
    - i. In the Control Center, right-click the original relationship table and choose **Export**.

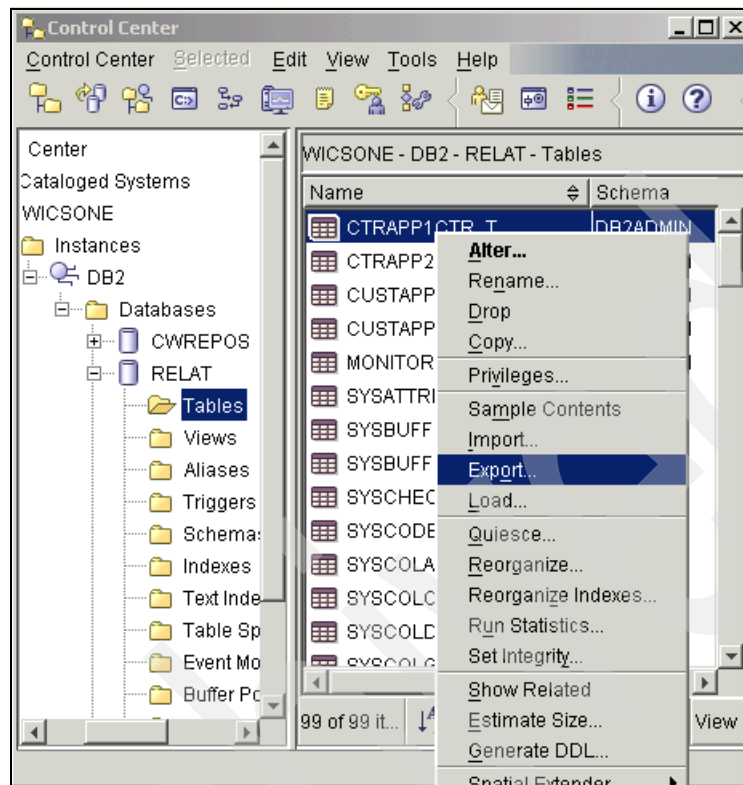


Figure 9-21 Exporting a table



- ii. Enter an output file location and a message file location and click **OK** (Figure 9-22).

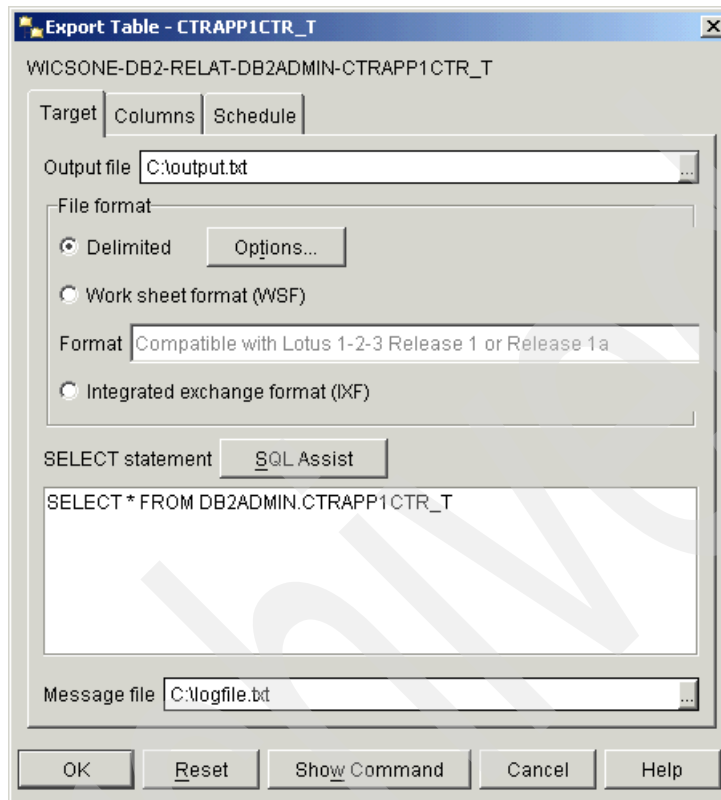


Figure 9-22 Completing the table export

**Tip:** If the OK button is not enabled, in the Schedule tab, select the option **Run now without saving task history**.

- iii. In the Control Center, right-click the newly created relationship table and choose **Import**.

- iv. Enter the location of the saved file as the import file and click **OK** as shown in Figure 9-23.

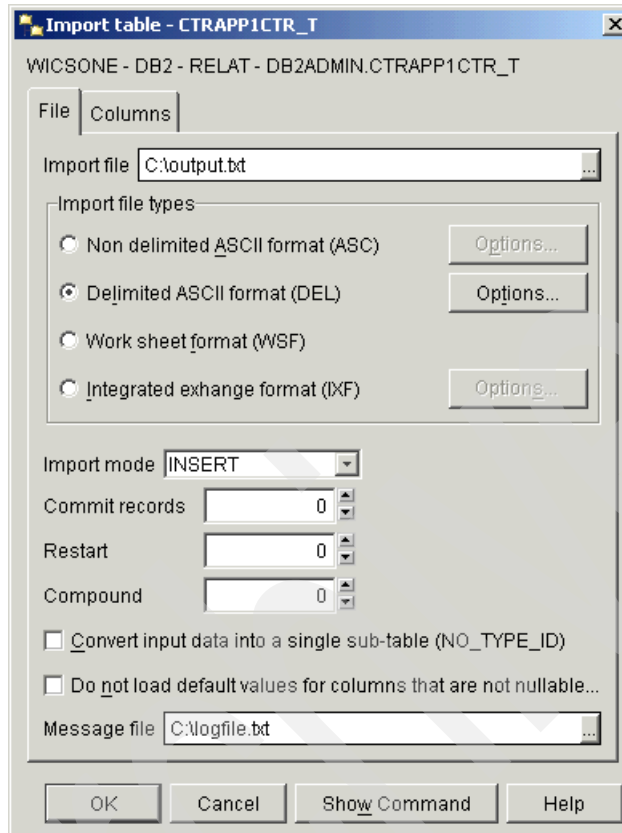


Figure 9-23 Importing the data

- b. Additional steps are needed to set the sequence number for each relationship. Here are the steps required to set the sequence number when the relationships are stored in DB2.
  - i. Find the current highest ID for any relationship by examining the INSTANCEID column for any participant table in the relationship.

```
SELECT MAX(INSTANCEID) FROM DB2ADMIN.CUSTAPP1CUST_T
```

In this example, the query has returned the value 41 as the highest INSTANCEID.
  - ii. Find the sequence variable name. The following command generates a SQL script that provides information about the relationship table. This

information can also be obtained in the control center by right-clicking a database table and choosing **Generate DDL**.

```
db2look -dRELAT -tCUSTAPP1CUST_T -a -e -c -oOUTPUT.SQL
```

Examine the OUTPUT.SQL file to obtain the sequence variable name. Search for the text CREATE SEQUENCE.

```
CREATE SEQUENCE "DB2ADMIN"."CUSTOMER_1_0_0" AS INTEGER  
MINVALUE 1 MAXVALUE 2147483647 START WITH 1 INCREMENT BY 1  
CACHE 20 NO CYCLE NO ORDER;
```

In the previous example, the sequence variable name is Customer\_1\_0\_0. The sequence name is derived from the name of the relationship. In this case the relationship name is Customer.

- iii. Set the sequence variable to be at least one larger than the current number. In this example, the current highest INSTANCEID is 41 so the sequence is set to 42 with this command:

```
alter sequence Customer_1_0_0 restart with 42
```

- c. Verify that the relationship has been successfully moved to the new database by examining the relationship using Relationship Manager.

### 9.2.3 Post-migration steps

This section describes the steps to be taken after migration is complete and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the password previously assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.

2. Right-click the server name in the InterChange Server component management view and select **Users and Roles**. This is shown in Figure 9-24.

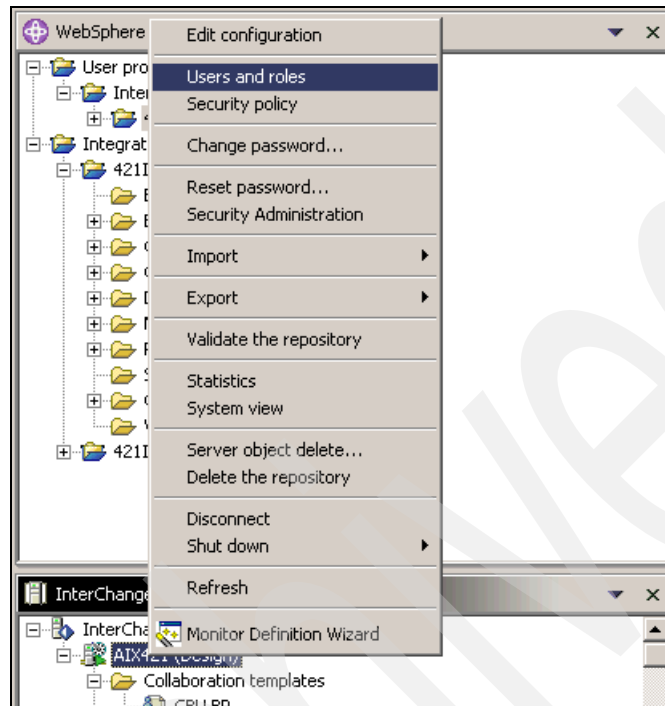


Figure 9-24 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User**. This is shown in Figure 9-25.

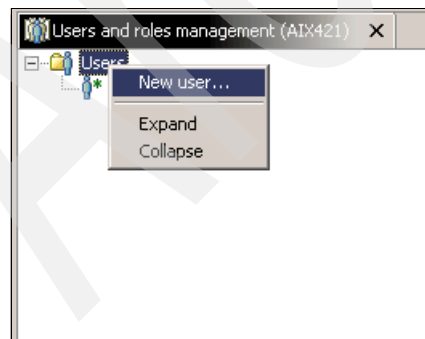


Figure 9-25 Selecting New user

4. The Create User window opens as shown in Figure 9-26. Enter the name **admin** and the previous password, plus additional information if needed. Click **OK** when finished.

A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' (containing 'admin'), 'Password' (containing '\*\*\*\*'), 'Confirm password' (containing '\*\*\*\*'), 'Distinguished name' (empty), 'Common name' (empty), and 'Surname' (empty). At the bottom right are 'OK' and 'Cancel' buttons.

Create user

User name  
admin

Password  
\*\*\*\*

Confirm password  
\*\*\*\*

Distinguished name

Common name

Surname

OK Cancel

Figure 9-26 Creating a user

5. The new user is added to the Users and roles management window.

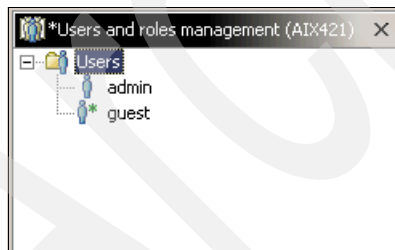


Figure 9-27 The newly created admin user

6. Right-click the newly created admin user and select **Add roles**. This is shown in Figure 9-28.

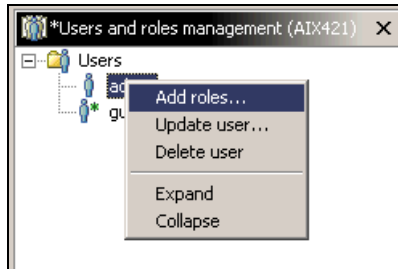


Figure 9-28 Add roles

7. The Role list window opens. Select **administrator** and click **OK**.

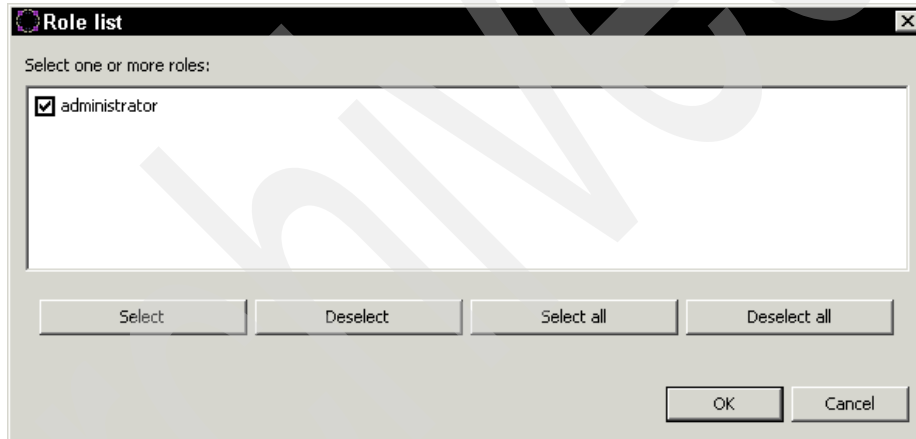


Figure 9-29 Role list

8. Close the Users and roles management window.



Figure 9-30 User admin with administrator role

9. Right-click the server again in the Components management view and choose **Edit Configuration**. This opens an editor for configuring the server.

10. Click the **Security-RBAC** tab (Figure 9-31) and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Enter admin for **Server start user**.
  - c. Set **Server start password** to the password for the admin user.
  - d. Check all other settings in the tab and modify if needed.
  - e. Save the configuration to the server by clicking the save icon or selecting **File → Save**.

Figure 9-31 Server Configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** Adapter agents default to the guest account if the RBAC is activated for the WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions on how to install adapters, refer to the individual adapter guides.

## Server Access Interface

If any Access Clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the migrated InterChange Server system from development to production, it is recommended that you perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information on testing a migrated system.

Table 9-3 shows the scenarios that were tested in this migration, and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 9-3 Test scenario results

Test scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated
Web-based System Monitor	Successfully migrated



**Note:** After all system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is re-started in Production mode.

## 9.3 In-place database migration of V4.2.1 to V4.3 on AIX

This section describes an in-place database migration scenario of WebSphere InterChange Server Version 4.2.1.7 to WebSphere InterChange Server Version 4.3.0. The complete repository from 4.2.1.7 is reused in this scenario. This includes component definitions, failed flows and relationships.

For the full description of a in-place database migration refer to Chapter 4, “General planning” on page 53.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 9.3.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system. The steps for performing these tasks are described in this section.

**Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

**Put the environment in a quiescent state.**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.

- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server in a quiescent state, perform the following steps:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator.
2. Resolve transactional status of transactional collaborations using the Flow Manager. Refer to the *Administration Guide* Version 4.2.1 page 136 for information on how to use the Flow Manager. Transactional collaborations that are in transactional status can be migrated, however it is recommended that they are resolved in this pre-migration step.
3. Resubmit failed events or discard the events using the Flow Manager. Refer to the *Administration Guide* Version 4.2.1 page 136 for information on how to use the Flow Manager. Failed flows can be migrated, however it is recommended that they are resolved in this pre-migration step.
4. Once all events are processed by the system, stop all collaborations. To check if all events are processed, check the collaboration statistics in the Windows-based System Monitor. Make sure the in-process and pending event counter is 0.

## Backing up and shutting down components

This section describes the steps needed to shutdown and take a backup. It is a recommended practice to take a backup which allows the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server Version 4.3.0.

Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*.

There are two types of data in the WebSphere InterChange Server system: static data and dynamic data.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These are the steps to perform the backup:

1. Back up the current InterChange Server repository using the `repos_copy` utility.

For example, suppose the WebSphere InterChange Server instance is AIX421 and has the default login of *admin* and the password of *null*. The following `repos_copy` command creates a backup of the repository objects in a file called `Repository421.jar`. This file is created in the current directory where the `repos_copy` command is invoked.

```
repos_copy -sAIX421 -uadmin -pnull -oRepository421.jar
```

A `repos_copy` is not used during an in place migration, however, it is still a recommended practice in order to back up the current definitions of the WebSphere InterChange Server repository.

To export existing user projects, connect System Manager running on a connected Windows machine to the existing WebSphere InterChange Server instance and take the following steps within System Manager:

- a. Right-click **User Projects** and select **Export Solution**.
- b. Select all user projects that are to be exported and enter a destination path.

**Restriction:** As we write this book, a product defect causes the Export Solution utility to not work properly. This defect has been reported and assigned tracking number CR60456. Check the support site to see whether a fix for this problem number is available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and choose **Export to Repository File**. The user projects must be re-created from the integration component library after the migration. The *Implementation Guide for WebSphere InterChange Server* has more information about this process.

2. Shut down the WebSphere InterChange Server. To use the System Manager, right-click the WebSphere InterChange Server name and choose **Shutdown Gracefully** or run the command on AIX:

```
$CROSSWORLDS/bin/ics_manager -stop
```

3. Arrange for the database administrator to back up the database. This must be a complete backup, including schema information and stored procedures.
4. Stop the Inprise VisiBroker Agent. Run the following command to determine the process ID of the osagent:

```
ps -ef | grep osagent
```

This is a sample output of the command:

```
cwadm421 35012      1   0   Aug 09      -   1:18 osagent
```

To stop the osagent run the command:

```
kill -9 35012
```

(35012 is the process ID of the osagent process.)

5. Arrange for the WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.

## 9.3.2 Migration steps

This section describes the main steps required for the migration, including:

- ▶ Upgrading related software to the required level.
- ▶ Installing and configuring WebSphere InterChange Server 4.3.0
- ▶ Completing the upgrades of WebSphere InterChange Server components

### Preparing the database system

The database used for the repository in WebSphere InterChange Server 4.2.1.7 may not meet the minimum database level required by Version 4.3.0 so it must be upgraded to the supported level. After upgrading the database, ensure that the database has the proper configuration for WebSphere InterChange Server 4.3.0.

- ▶ Check the *IBM WebSphere InterChange Server Installation Guide for UNIX* for the proper database settings for V4.3. For example, when using DB2, a change in the database settings is that the applheapsz parameter must be set to at least 4096.

Read or download the Installation Guide for UNIX at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **UNIX** in the right-side pane.

- Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 fix pack 7. If migrating failed flows that involve WebSphere MQ Delivery Transport, the queue messages must be preserved when upgrading WebSphere MQ. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the WebSphere MQ Messaging menu, select **latest multiplatform books**.

## Installation

At this point, install WebSphere InterChange Server Version 4.3.0. Refer to Chapter 6, "Installation of WebSphere InterChange Server 4.3" on page 99 for further instructions.

It is recommended that WebSphere InterChange Server Version 4.3.0 is installed with a different user ID and to a different directory than the previous version. The previous installation is not modified during the install process.

During the install process, use the same WebSphere InterChange Server name as in the previous version. As a reminder, the WebSphere InterChange Server name must not contain dots (.).

The ICS name can be changed after the installation process. Edit the file `$CROSSSWORLDS/bin/ics_manager` and the file `$ADAPTER_RUNTIME/bin/connector_manager`. Change the parameter `SERVERNAME` to the desired WebSphere InterChange Server name.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following pictures and examples explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
$CROSSWORLDS/bin/ICConfig.sh
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all the configuration settings have to be entered in the file `Options.txt` which is delivered with the installer. For detailed information, refer to the *Installation Guide for UNIX* (page 69).

The following figures show the configuration values used by the redbook team during migration:

1. Figure 9-32 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

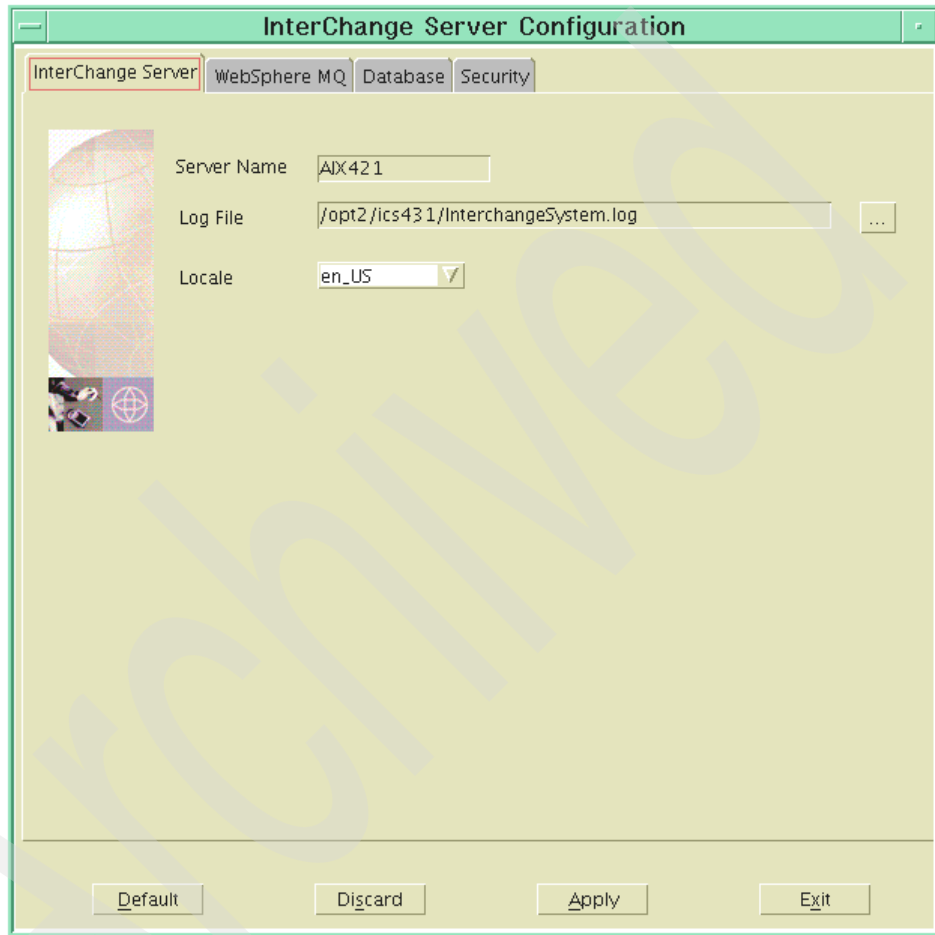
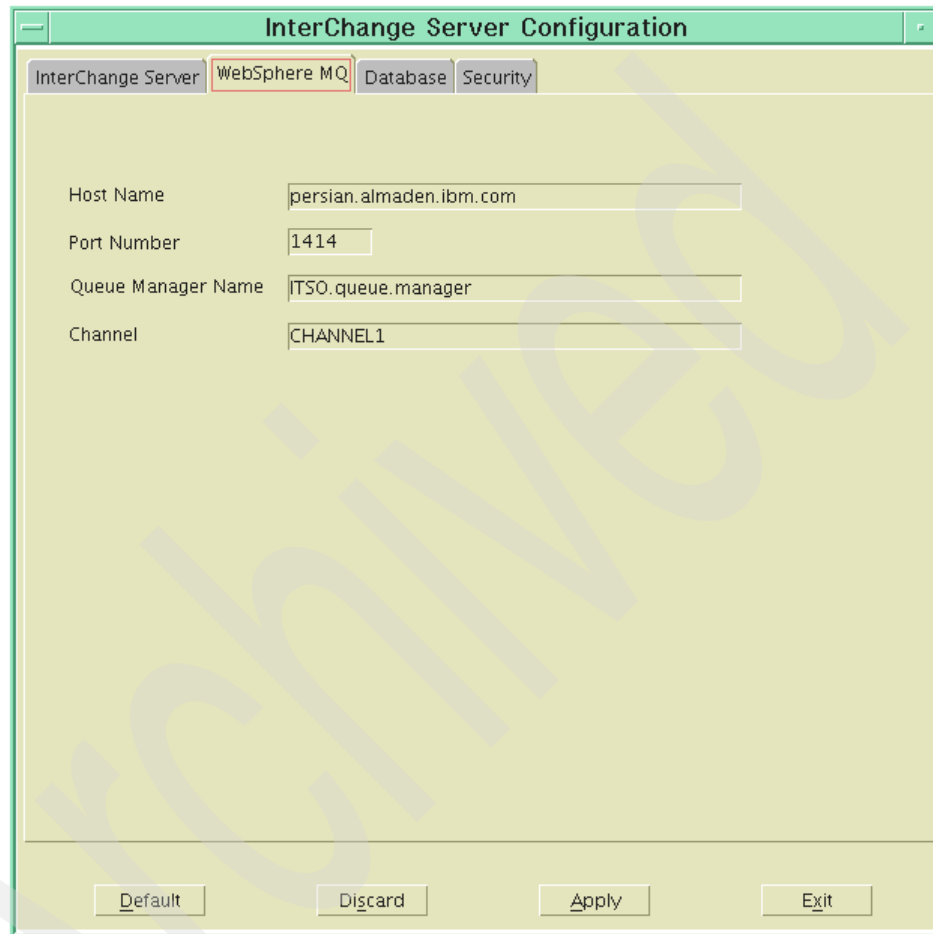


Figure 9-32 InterChange Server Configuration: InterChange Server tab

2. Figure 9-33 shows the WebSphere MQ configuration tab. Make sure the settings for the WebSphere InterChange Server Version 4.2.1.7 are entered here.



The image shows a dialog box titled "InterChange Server Configuration". It has four tabs: "InterChange Server", "WebSphere MQ", "Database", and "Security". The "WebSphere MQ" tab is selected and highlighted with a red border. The dialog contains four input fields with labels to their left: "Host Name" with the value "persian.almaden.ibm.com", "Port Number" with the value "1414", "Queue Manager Name" with the value "ITSO.queue.manager", and "Channel" with the value "CHANNEL1". At the bottom of the dialog, there are four buttons: "Default", "Discard", "Apply", and "Exit".

Field	Value
Host Name	persian.almaden.ibm.com
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 9-33 InterChange Server Configuration: WebSphere MQ tab



3. The third tab is for database configuration (Figure 9-34). Make sure to enter the connection information from WebSphere InterChange Server 4.2.1.7 and apply the following settings:
- Max Connections: unlimited, if possible. For the purposes of this book, the Max Connections parameter was set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' window with the 'Database' tab selected. The window has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab is active and contains several sections:

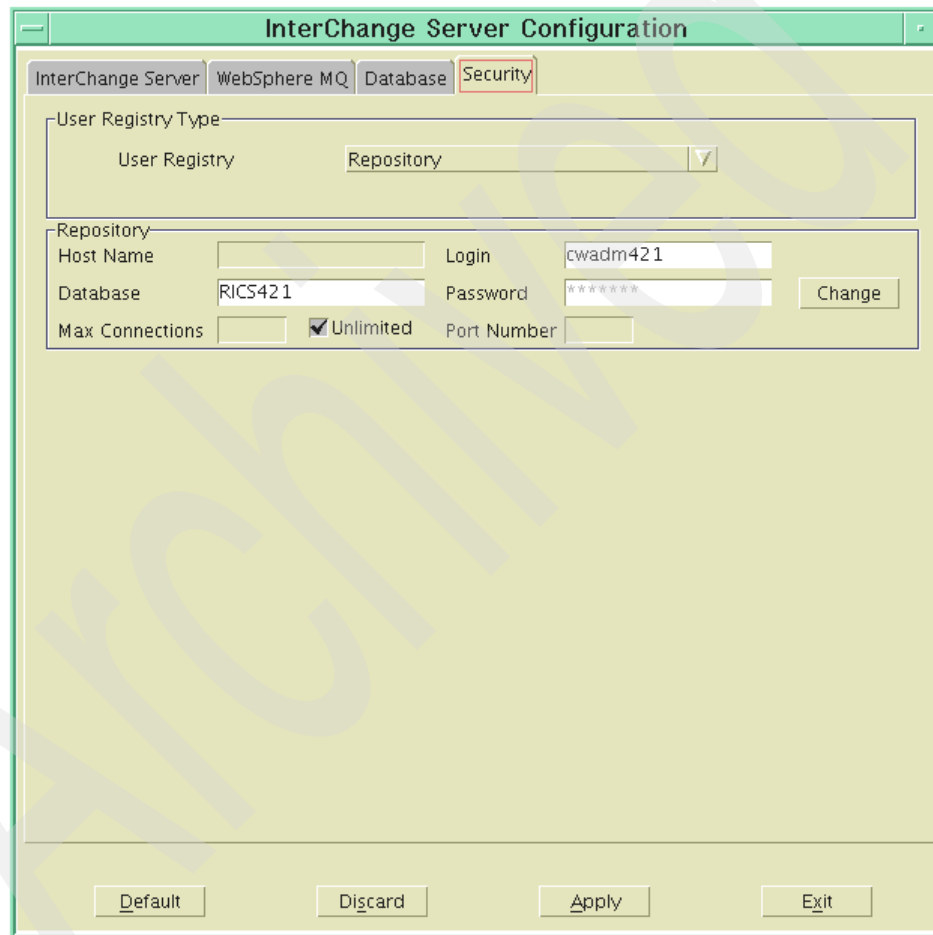
- Database Connectivity:** Includes a dropdown for 'Database Driver' set to 'DB2'. To the right, 'Max Connections' is set to 'Unlimited' (checked), 'Max Pools' is set to '50', and 'Idle Timeout' is set to '2'.
- Event Management:** Includes fields for 'Host Name', 'Database' (set to 'RICS421'), 'Login' (set to 'cwadm421'), and 'Password' (masked with asterisks). There is a 'Change' button. 'Max Connections' is set to 'Unlimited' (checked), and 'Port Number' is empty.
- Transactions:** Similar to Event Management, with 'Database' set to 'RICS421', 'Login' set to 'cwadm421', and 'Max Connections' set to 'Unlimited' (checked).
- Repository:** Similar to Event Management, with 'Database' set to 'RICS421', 'Login' set to 'cwadm421', and 'Max Connections' set to 'Unlimited' (checked).
- Flow Monitoring:** Includes fields for 'Host Name', 'Database' (set to 'RICS421'), 'Login' (set to 'cwadm421'), 'Schema Name', and 'Password' (masked). There is a 'Change' button. 'Max Connections' is set to 'Unlimited' (checked).

At the bottom of the window are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 9-34 InterChange Server configuration: Database tab

4. The fourth tab is for security configuration (Figure 9-35). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'Security' tab. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Security' tab is selected and highlighted with a red border. Inside the 'Security' tab, there is a section titled 'User Registry Type' with a dropdown menu set to 'Repository'. Below this, there is a 'Repository' section with several fields: 'Host Name' (empty), 'Login' (containing 'cwadm421'), 'Database' (containing 'RICS421'), 'Password' (containing '\*\*\*\*\*'), 'Max Connections' (empty), a checked 'Unlimited' checkbox, and 'Port Number' (empty). A 'Change' button is located to the right of the 'Password' field. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

InterChange Server Configuration			
InterChange Server WebSphere MQ Database <b>Security</b>			
User Registry Type			
User Registry		Repository	
Repository			
Host Name		Login	cwadm421
Database	RICS421	Password	*****
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number
		Change	
Default Discard Apply Exit			

Figure 9-35 InterChange Server Configuration: Security tab

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMs and collaboration directories from WebSphere InterChange Server 4.2.1.7 to the new installation directory for Version 4.3.0. Be sure to include all subdirectories.

Example from this migration scenario:

```
cp -R /OLD INSTALLATION PATH/DLMs /NEW INSTALLATION PATH/
cp -R /OLD INSTALLATION PATH/collaborations /NEW INSTALLATION PATH/
```

2. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.

For example, to disable DEADLOCK\_DETECTOR\_CHECK for collaborations, edit the config file code shown in Example 9-3 using a suitable editor for XML.

### Example 9-3 Disable DEADLOCK\_DETECTOR\_CHECK

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>false</tns:reposController>
<tns:reposAgent>false</tns:reposAgent>
<tns:localConfig>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change true to false in the line starting with <tns:value.

3. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, these must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
4. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in %CROSSWORLDS/bin/CWSharedEnv.sh. The external classes are first added to a new variable EXT\_LIB. Then the EXT\_LIB is appended to the CWCLASSES variable. See the following example from CWSharedEnv.sh:

```
# external classes added here
EXT_LIB=${CROSSWORLDS}/lib/Sample.jar
```

```
CWCLASSES=${WLICENSE}:${CORECW}:${DATAHANDLERS}:${DATADIRECT}:${DATAMANAGER}:${ITLM}:${JAVAMAIL}:${WEBSERVICES_JARS}:${EXT_LIB}
export CWCLASSES
```

5. Modify and verify the environment variables as documented in the Installation Guide for UNIX.

For example, make sure the path to the Java 1.4.2 compiler precedes the Java 1.3.1 compiler if the 1.3.1 compiler is still in the path. The following example represents part of the \$HOME/.profile from one of our test servers:

```
#Start DB Environment
. /home/db2inst1/sqllib/db2profile
export EXTSHM=ON
# End DB Environment

# WBI environment
export PATH=/usr/java14/bin:$PATH
. /opt2/ics43/bin/CWSharedEnv.sh
#END WBI environment
```

Make sure the latest settings are in effect. To confirm that the right Java Compiler is activated, type **which javac**.

The following output is returned for AIX, which is the installation path for Java 1.4.2 JDK:

```
/usr/java14/bin/javac
```

6. Install the WebSphere Business Integration Toolset on a supported Windows client. If the tools are to be installed on a Windows client that currently hosts an older version of tools, the old tools must be uninstalled prior to the installation of the new tools. You should make a backup of the Windows client before uninstalling and installing the new tools.

During installation, a prompt for the location of the Object Request Broker is displayed. Enter the host name or IP address of the machine hosting the Persistent Name Server that WebSphere InterChange Server is using. Also enter the port number if it is different from the default, 14500. To change the settings after installation, edit %CROSSWORLDS%\bin\CWSharedEnv.bat:

```
set ORB_PORT=14500
set ORB_HOST=persian.almaden.ibm.com
```

## Starting the new server

When WebSphere InterChange Server is started for the first time after installation, the WebSphere InterChange Server repository tables are upgraded

or created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database server
2. Start the Persistent Name Server. In a new console, run this command:

```
$CROSSWORLDS/bin/PersistentNameServer.sh
```

3. Start the WebSphere InterChange Server in design mode:

```
$CROSSWORLDS/bin/ics_manager -start -design
```

The WebSphere InterChange Server upgrades existing tables and creates new tables for new features.

WebSphere InterChange Server logs messages to the `InterChangeSystem.log` file located in the product directory. The message `Interchange Server is Ready` indicates that WebSphere InterChange Server has started successfully.

## Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, import the component libraries using exported integration component library files:

- ▶ Create a new integration component library.
- ▶ Right-click the newly created library and select **Import Repository from File**. Choose the repository file that was created in the backup procedure. Repeat these steps if there are multiple integration component libraries to import. The *Implementation Guide for WebSphere InterChange Server* has more information about this process.

2. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH for the Toolset. Add the location of the custom .jar files to this list.

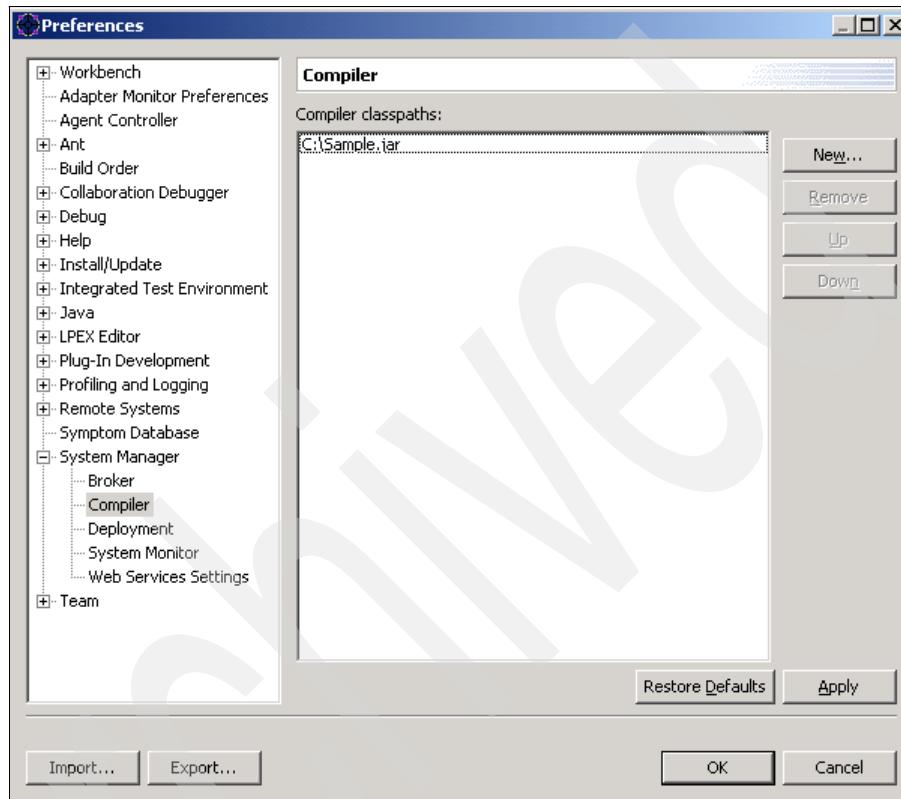


Figure 9-36 Setting the classpath for the Toolset

- Set the compliance of the compiler to Java 1.4. Select **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag, change the Compiler compliance level to **1.4**. A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to not rebuild files at this time.

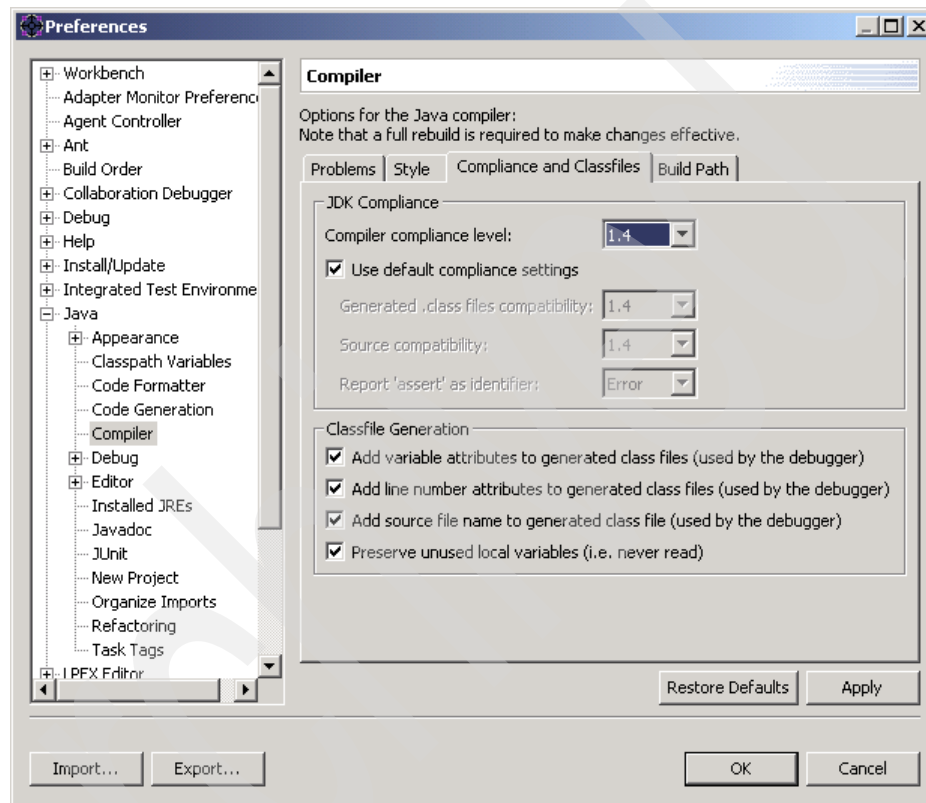


Figure 9-37 Setting the Compiler compliance level

- If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the usage of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level has been correctly set in previous step.
- Compile all the maps and templates in the integration component libraries and examine the console for any error messages.

6. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more detail.
  - c. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions examine the Port and Triggering Events tab. For the **BO Type** that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to <None> for all verbs.
  - d. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.

**Note:** If Import Solution was not used, create a new user project and populate it with the components from the integration component library. See the *Implementation Guide for WebSphere InterChange Server* for more information.

7. Resolve any collaborations that are transactional status, because transactional status is lost after redeployment of the collaboration. This can be done from the Flow Manager. Refer to the *WebSphere InterChange Server System Administration Guide* for further information.

If the collaboration cannot be brought to an active state to resolve the transactional status, deploy only the updated components required to activate the integration, then resolve the transactional status.

**Note:** To make use of all administrative features for collaborations and maps, the maps and templates must be deleted and redeployed. Typically, each component must be placed in an inactive state before it can be overwritten. To simplify the process and avoid unresolved dependencies, the whole repository is deleted in step 8 and redeployed in step 9. Additionally, recompilation ensures that the code is optimized and Java 1.4 compliant.

However, redeploying the maps and templates is not required, because the Java Runtime Environment is backward-compatible and poses no problems executing map and template class files compiled in WebSphere InterChange Server 4.2.1.7.

If the decision is made to reuse the previous maps and collaborations, only deploy the components which have been updated.



8. In the System Manager, delete the repository by right-clicking the WebSphere InterChange Server name in the InterChange Server component management view and select **Delete the repository**.
9. Deploy the user project.
  - a. Right-click the user project and select **Deploy User Project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This recreates the stored procedures and does not overwrite existing data.

10. Restart the WebSphere InterChange Server if prompted.

### 9.3.3 Post-migration steps

This section describes the steps that need to be taken after migration and after the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the previous password, assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.
2. Right-click the server name in the InterChange Server component management view and select **Users and Roles**. This is shown in Figure 9-38.

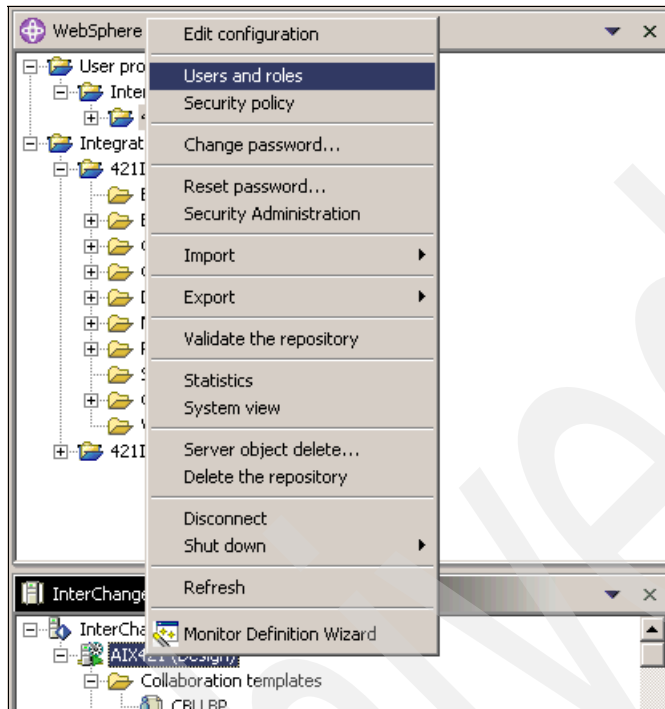


Figure 9-38 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click Users icon and select **New User** (Figure 9-39).

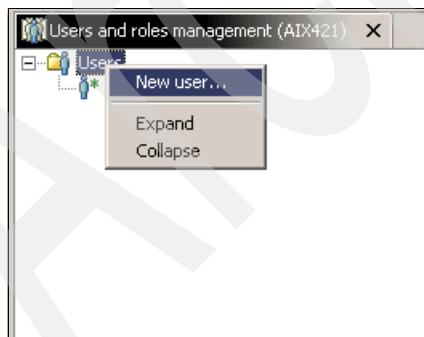


Figure 9-39 Selecting New user

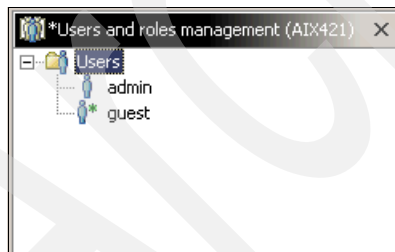
4. The Create user window opens (Figure 9-40). Enter the name `admin` and the previous password, plus additional information if needed. Click **OK**.



A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' (containing 'admin'), 'Password' (containing '\*\*\*\*'), 'Confirm password' (containing '\*\*\*\*'), 'Distinguished name' (empty), 'Common name' (empty), and 'Surname' (empty). At the bottom right are 'OK' and 'Cancel' buttons.

*Figure 9-40 Creating a user*

5. The new user is added to the User and Roles management window (Figure 9-41).



*Figure 9-41 The newly created admin user*

6. Right-click the newly created admin user and select **Add roles** (Figure 9-42 on page 290).

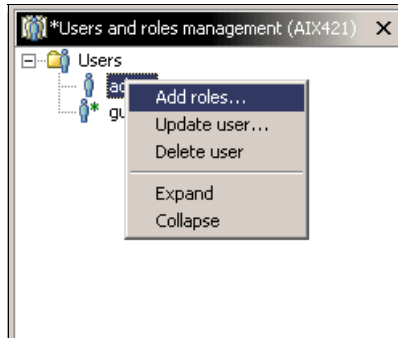


Figure 9-42 Add roles

7. The Role list window is displayed (Figure 9-43).

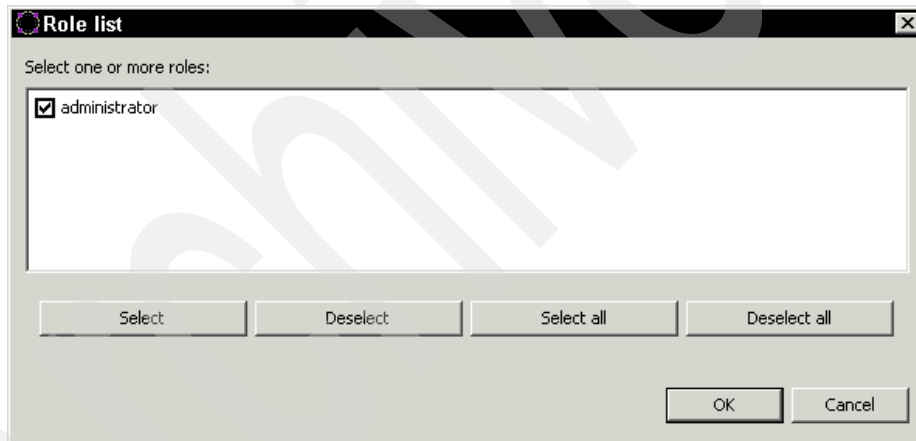


Figure 9-43 Role list

8. Select **administrator** and click **OK**.

9. Close the Users and roles management window.

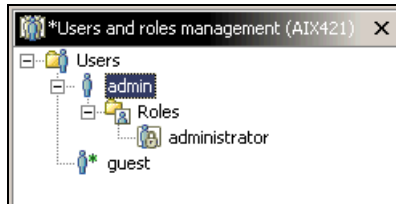


Figure 9-44 User admin with administrator role

10. Right-click the server again in the InterChange Server Component management view and choose **Edit Configuration**. An editor to configure the server is displayed.

11. Click the **Security - RBAC** tab shown in Figure 9-45.

Figure 9-45 Server Configuration - Security-RBAC tab

12. In the tab, apply the following settings:

- Select **Enable RBAC**.
- Set **Server start user** to admin.
- Set **Server start password** to the password for the admin user.
- Check all other settings in the tab and modify if needed.

- e. Save the configuration to the server by clicking the save icon or choosing **File → Save**.

13.Reboot WebSphere InterChange Server to activate security.

**Note:** The adapter agent defaults to the guest account if RBAC is activated for WebSphere InterChange Server and no user/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions on how to install adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they need to be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for further information.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end and performance testing. Before moving the migrated InterChange Server system from development to production it is recommended to perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 9-1 shows the scenarios that were tested in this migration, and the results found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 9-4 Test scenarios results

Test scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated

Test scenario	Results
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated
Web-based System Monitor	Successfully migrated

A failed flow is considered to be successfully migrated if the event can be resubmitted and completed after migration. Table 9-2 shows the results of the migration of Failed Flows.

*Table 9-5 Migrated failed flows results*

Failed flows tested	Results
Failed flow for IDL	Successfully migrated
Failed flow for MQ	Successfully migrated
Failed flow for JMS	Successfully migrated
Transactional Collaboration in transactional status (compensation call failed)	Successfully migrated
Transactional Collaboration failed flow (compensation call successful)	Successfully migrated
LLBP waiting asynchronous inbound call	Successfully migrated
LLBP failed flow	Successfully migrated
Collaboration group failed at Destination 1	Successfully migrated
Collaboration group failed at Destination 2	Successfully migrated

**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is re-started in Production mode.

### Uninstall (optional)

This section describes the steps to uninstall the previous version of WebSphere InterChange Server and the related software. The following steps list the order of uninstalling relevant software

1. Uninstall the VisiBroker ORB for version earlier than 4.2.2 using the uninstaller located in the VisiBroker product directory.
2. Uninstall JDK as the root user using `smi t`, only if it is not required by other products. For this release, the JDK is typically Java 2 SDK v.1.3.1. Contact the system administrator for further assistance.
3. Uninstall WebSphere InterChange Server using the uninstaller located in the product directory once it is certain that the previous version is no longer needed. After the uninstaller has completed, the product directory can be deleted.

## 9.4 Without in-place database migration of V4.2.1 to V4.3 on AIX

This section describes a without in-place database migration scenario of Version 4.2.1 to Version 4.3.0. The 4.3.0 WebSphere InterChange Server is started with a new and empty repository. Repository content (component definitions) are exported from the original WebSphere InterChange Server Version 4.2.1 and deployed in WebSphere InterChange Server Version 4.3.0 environment.

For the full description of a without in-place database migration refer to Chapter 4, “General planning” on page 53.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 9.4.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system. The steps for performing these tasks are described in this section.



**Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

**Put the environment in a quiescent state**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator.
2. Resolve transactional status of transactional collaborations using the Flow Manager. Refer to the *Administration Guide* Version 4.2.1 for information about using the Flow Manager. Transactional collaborations that are in transactional status can be migrated, but it is recommended that they be resolved in this pre-migration step.
3. Resubmit failed events or discard the events using the Flow Manager. Refer to the *Administration Guide* Version 4.2.1 page 136 for information on how to use the Flow Manager. Failed flows can be migrated, however it is recommended that they are resolved in this pre-migration step.
4. When all events have been processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in the Windows-based System Monitor. Make sure that the in-process and pending event counter is 0.

## Backing up and shutting down components

This section describes the steps for shutting down and taking a backup. It is a recommended practice to take a backup which allows the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server Version 4.3. Additional information about backing up the system can be found in the *WebSphere InterChange Server System Administration Guide*.

There are two types of data in the WebSphere InterChange Server system: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current WebSphere InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is AIX421 and has the default login of **admin** and the password of **null**. The following **repos\_copy** command creates a backup of the repository objects in a file called **Repository421.jar**. This file is created in the directory where the **repos\_copy** command is invoked.

```
repos_copy -sAIX421 -uadmin -pnull -oRepository421.jar
```

To export existing user projects, connect System Manager running on a connected Windows machine to the existing WebSphere InterChange Server instance and take the following steps in System Manager:

- a. Right-click **User Projects** and select **Export Solution**.
- b. Select all user projects that are to be exported and enter a destination path.

**Restriction:** As we write this book, a product defect causes the Export Solution utility to not work properly. This defect has been reported and assigned tracking number CR60456. Check the support site to see whether a fix for this problem number is available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and choose **Export to Repository File**. The *Implementation Guide for WebSphere InterChange Server* has more information about this process. The user projects must be re-created from the integration component library after the migration.

2. Shut down the WebSphere InterChange Server. To use the System Manager, right-click the WebSphere InterChange Server name and choose **Shutdown Gracefully** or run the command on AIX:

```
$CROSSWORLDS/bin/ics_manager -stop
```

3. Arrange for the database administrator to back up the database. This must be a complete backup, including schema information, and stored procedures.
4. Stop the Inprise VisiBroker Agent. Run the following command to determine the process ID of the osagent.

```
ps -ef | grep osagent
```

This is a sample output of the command:

```
cwadm421 35012      1   0   Aug 09      -   1:18 osagent
```

To stop the osagent run the command:

```
kill -9 35012
```

where 35012 is the process ID of the osagent process.

5. Clear WebSphere MQ queues of any outstanding messages if the same queue manager is reused for Version 4.3.0. Old messages in the queues may be considered as new events and may cause data inconsistencies. Refer to the WebSphere MQ documentation for more information.

## 9.4.2 Migration steps

This section describes the main steps required for the migration:

- ▶ Upgrading related software to the required level
- ▶ Installing and configuring WebSphere InterChange Server 4.3.0
- ▶ Completing the upgrades of WebSphere InterChange Server components

## Preparing the database system

Prepare an empty database for the 4.3.0 repository: Create a new database following the steps in the Installation Guide, and ensure that it has the proper configuration for WebSphere InterChange Server 4.3.0.

- ▶ Check the installation guide for the proper database settings for 4.3.0. For example, when using DB2, one change in the database settings is that the `applheapsz` parameter must be set to at least 4096.
- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

- ▶ For DB2, it is recommended that you catalog the database system to route using TCP/IP as documented in the System Installation Guide for Unix.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 fix pack 7. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 for instructions.

It is recommended that WebSphere InterChange Server Version 4.3.0 be installed with a different user ID and to a different directory than the previous version. The previous installation is not modified during the install process.

During the install process, the WebSphere InterChange Server name may be changed. If so, the adapters and queues may have to be configured. Use the same WebSphere InterChange Server name if possible.

The WebSphere InterChange Server name can be changed after the installation process. Edit the file `$CROSSWOLDS/bin/ics_manager` and the file `$ADAPTER_RUNTIME/bin/connector_manager`. Change the parameter `SERVERNAME` to the desired WebSphere InterChange Server name.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following pictures and examples explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
$CROSSWOLDS/bin/ICConfig.sh
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all the configuration settings have to be entered in the file `Options.txt` which is delivered with the installer. For detailed information, refer to the *Installation Guide for UNIX* (page 69).

The following figures show the configuration values used by the redbook team during migration:

1. Figure 9-46 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

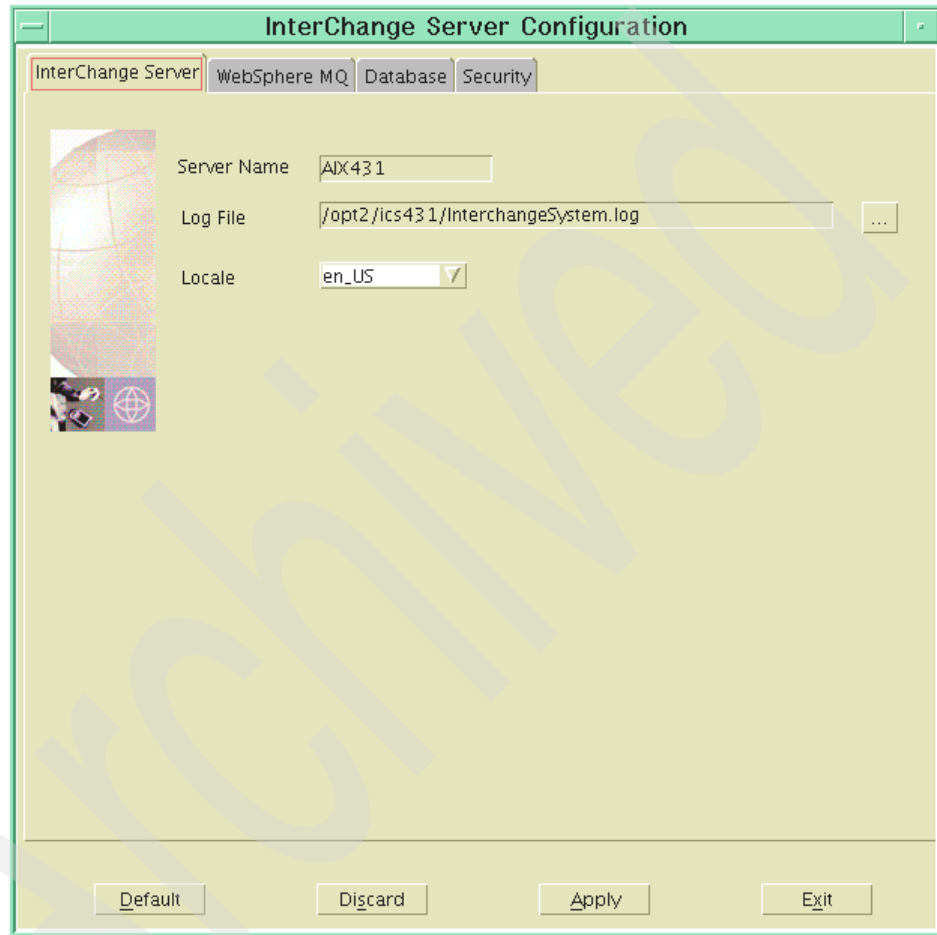
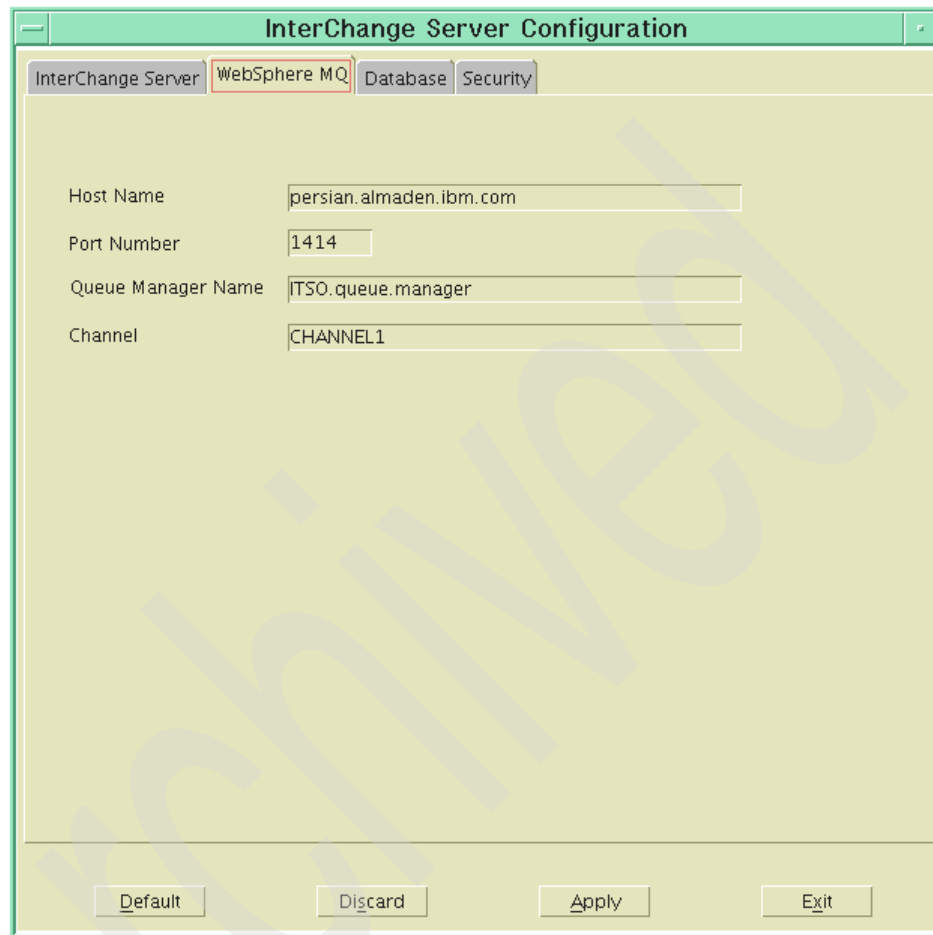


Figure 9-46 InterChange Server configuration: InterChange Server tab

2. Figure 9-47 shows the WebSphere MQ configuration tab.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'WebSphere MQ' tab. The dialog has a title bar and four tabs: 'InterChange Server', 'WebSphere MQ' (which is selected and highlighted with a red border), 'Database', and 'Security'. The main area contains four labeled text input fields: 'Host Name' with the value 'persian.almaden.ibm.com', 'Port Number' with the value '1414', 'Queue Manager Name' with the value 'ITSO.queue.manager', and 'Channel' with the value 'CHANNEL1'. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Field	Value
Host Name	persian.almaden.ibm.com
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 9-47 InterChange Server Configuration: WebSphere MQ tab

3. The third tab is for database configuration (Figure 9-48). Make sure to enter the connection information for the empty database created in the previous step and apply the following settings:
- Max Connections: unlimited, if possible. For the purposes of this book, the Max Connections parameter was set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

**InterChange Server Configuration**

InterChange Server | WebSphere MQ | **Database** | Security

**Database Connectivity**

Database Driver: DB2 | Max Connections: ☐ ☒ Unlimited | Max Pools: 50 | Idle Timeout: 2

**Event Management**

Host Name: | Login: cwadm431 | Database: RICS431 | Password: | Change | Max Connections: ☐ ☒ Unlimited | Port Number: |

**Transactions**

Host Name: | Login: cwadm431 | Database: RICS431 | Password: | Change | Max Connections: ☐ ☒ Unlimited | Port Number: |

**Repository**

Host Name: | Login: cwadm431 | Database: RICS431 | Password: | Change | Max Connections: ☐ ☒ Unlimited | Port Number: |

**Flow Monitoring**

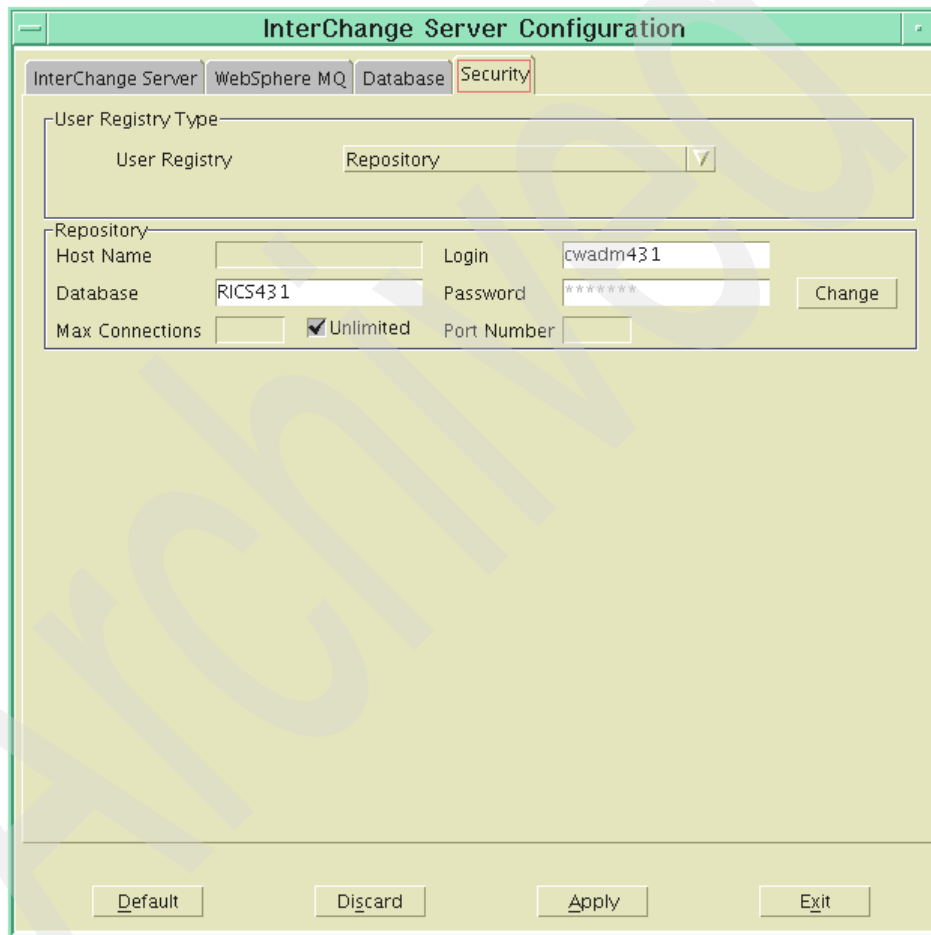
Host Name: | Login: cwadm431 | Database: RICS431 | Password: | Change | Schema Name: | Port Number: | Max Connections: ☐ ☒ Unlimited

Figure 9-48 InterChange Server Configuration: Database tab



4. The fourth tab is for security configuration (Figure 9-49). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'Security' tab. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Security' tab is active and highlighted with a red border. Inside the 'Security' tab, there is a section titled 'User Registry Type' with a dropdown menu set to 'Repository'. Below this, there is a 'Repository' section with several fields: 'Host Name' (empty), 'Login' (containing 'cwadm431'), 'Database' (containing 'RICS431'), 'Password' (containing '\*\*\*\*\*'), 'Max Connections' (empty), a checked 'Unlimited' checkbox, and 'Port Number' (empty). A 'Change' button is located to the right of the 'Password' field. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

InterChange Server Configuration			
InterChange Server WebSphere MQ Database <b>Security</b>			
User Registry Type			
User Registry	Repository		
Repository			
Host Name		Login	cwadm431
Database	RICS431	Password	*****
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number
Change			
Default Discard Apply Exit			

Figure 9-49 InterChange Server Configuration: Security tab

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server Version 4.3.0, additional steps are required to prepare the environment. These are outlined as follows.

1. Apply any custom configuration settings to `$CROSSWORLDS/InterchangeSystem.cfg`.

For example, to disable `DEADLOCK_DETECTOR_CHECK` for collaborations, edit the config file excerpt in Example 9-4 using a suitable editor for XML.

### *Example 9-4 Disable DEADLOCK\_DETECTOR\_CHECK*

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>false</tns:reposController>
<tns:reposAgent>false</tns:reposAgent>
<tns:localConfig>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change true to false in the line starting with `<tns:value`.

2. With WebSphere InterChange Server 4.2.2, all startup scripts were changed to accommodate the move from the VisiBroker ORB to the IBM ORB and support for the IBM JRE. If there are any startup or custom scripts that reference VisiBroker ORB, these must be modified. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more information.
3. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in `$CROSSWORLDS/bin/CWSharedEnv.sh`. The external classes are first added to a new variable, `EXT_LIB`, which is then appended to the `CWCLASSES` variable. See the following excerpt from `CWSharedEnv.sh`:

```
# external classes added here
EXT_LIB=${CROSSWORLDS}/lib/Sample.jar

CWCLASSES=${WLICENSE}:${CORECW}:${DATAHANDLERS}:${DATADIRECT}:${DATAMANAGER}:${ITLM}:${JAVAMAIL}:${WEBSERVICES_JARS}:${EXT_LIB}
export CWCLASSES
```

4. Modify and verify the environment variables as documented in the Installation Guide for UNIX.

For example, make sure the path to the JAVA 1.4.2 compiler precedes the JAVA 1.3.1 compiler if the JAVA 1.3.1 compiler is still in the path. This excerpt represents part of the \$HOME/.profile from one of our test servers:

```
#Start DB Environment
. /home/db2inst1/sqllib/db2profile
export EXTSHM=ON
# End DB Environment

# WBI environment
export PATH=/usr/java14/bin:$PATH
. /opt2/ics43/bin/CWSharedEnv.sh
#END WBI environment
```

Make sure the latest settings are in effect. To confirm that the right Java Compiler is activated, run the **which javac** command.

The following output is returned for AIX, which is the installation path for Java 1.4.2 JDK:

```
/usr/java14/bin/javac
```

5. Install the WebSphere Business Integration Toolset on a supported Windows client. If the tools are to be installed on a Windows client that currently hosts an older version of tools, the old tools must be uninstalled prior to the installation of the new tools. It is recommended to take a backup of the Windows client before uninstalling and installing the new tools.

During installation, a prompt for the location of the Object Request Broker is displayed. Enter the host name or IP address of the machine hosting the Persistent Name Server that the WebSphere InterChange Server is using. Also enter the port number if it is different from the default, 14500. To change settings after installation, edit %CROSSWORLDS%\bin\CWSharedEnv.bat.

```
set ORB_PORT=14500
set ORB_HOST=persian.almaden.ibm.com
```

## Starting the new server

When WebSphere InterChange Server is started for the first time after installation, the WebSphere InterChange Server repository tables are created. This section describes the steps to prepare and start the server for the first time.

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database

2. Start the Persistent Name Server. In a new console run the command.

```
$CROSSWORLDS/bin/PersistentNameServer.sh
```

3. Start the WebSphere InterChange Server in design mode:

```
$CROSSWORLDS/bin/ics_manager -start -design
```

The WebSphere InterChange Server logs messages to the InterChangeSystem.log file located in the product directory. The message Interchange Server is Ready indicates that the WebSphere InterChange Server has started successfully.

### Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, import the component libraries using exported integration component library files.

1. Create a new integration component library.
2. Right-click the newly created integration component library and select **Import Repository from File**. Choose the repository file that was created in the backup procedure. Repeat these steps if there are multiple integration component libraries to import. The *Implementation Guide for WebSphere InterChange Server* contains further information about this process.

3. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH for the toolset. Add the location of the custom .jar files to this list (Figure 9-50).

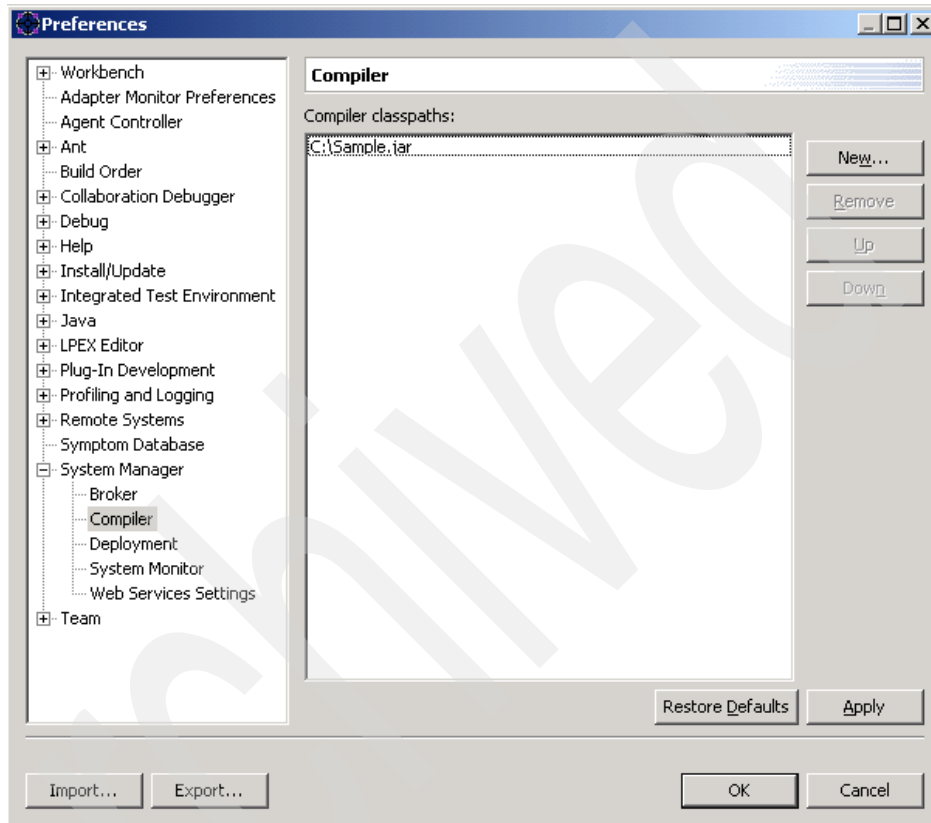


Figure 9-50 Setting the classpath for the toolset

4. Set the compliance of the compiler to Java 1.4. Click **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag, change the Compiler compliance level to **1.4**. A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to not rebuild files at this time. Figure 9-51 shows an example

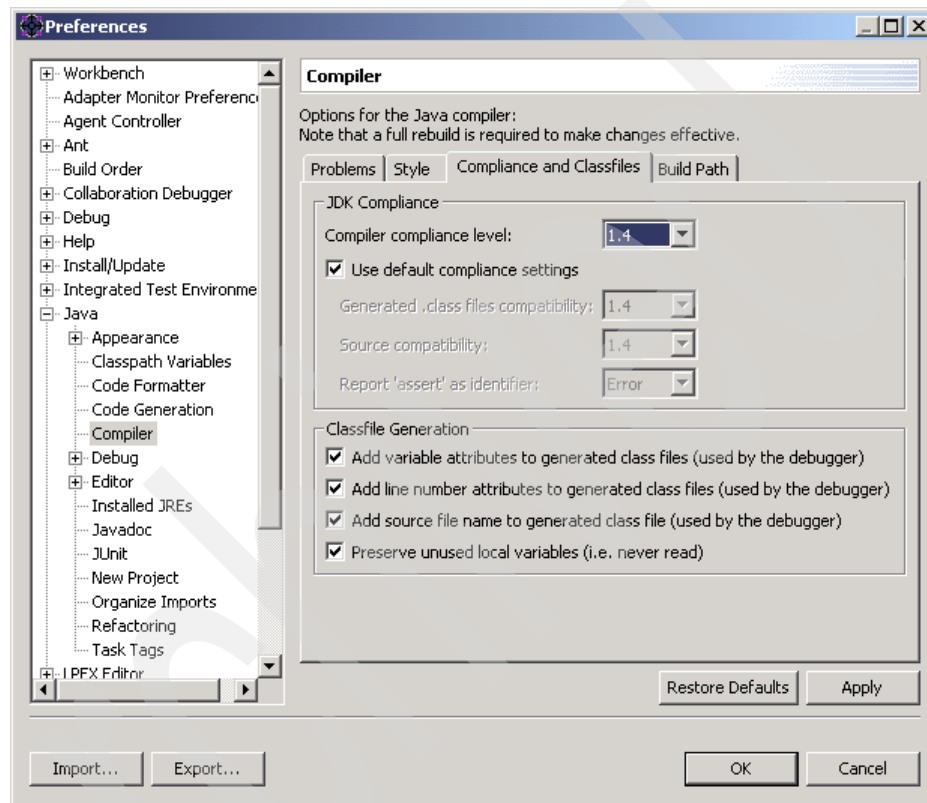


Figure 9-51 Setting the Compiler compliance level

5. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the usage of the import statement. In Java 1.4, the import statement must be followed by a packaged class, but Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in the previous step.
6. Compile all maps and templates in the integration component libraries and examine the console for any error messages.

7. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. See Chapter 5, “Technical impact of major design changes” on page 67 for more details.
  - c. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions examine the Port and Triggering Events tab. For the **BO Type** that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to <None> for all verbs.
  - d. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.

**Note:** If Import Solution was not used, create a new user project and populate it with components from the integration component library. See the *Implementation Guide for WebSphere InterChange Server* for details.

8. Deploy the user project.
  - a. Right-click the user project and select **Deploy user project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. If the location of the relationship tables has changed, select **Create Schema**.
  - f. Click **Finish** to start the deployment.
9. Restart the WebSphere InterChange Server if prompted.

10. Completing relationship migration: If a new relationship location is used, the relationship data from the 4.2.1.7 relationship tables can be migrated as so:
  - a. For both static and dynamic relationships, copy the data from each relationship table to the new relationship table. (Consult with the database administrator.) The steps to complete this process using DB2 are:
    - i. In the Control Center, right-click the original relationship table and choose **Export**. This is shown in Figure 9-52.

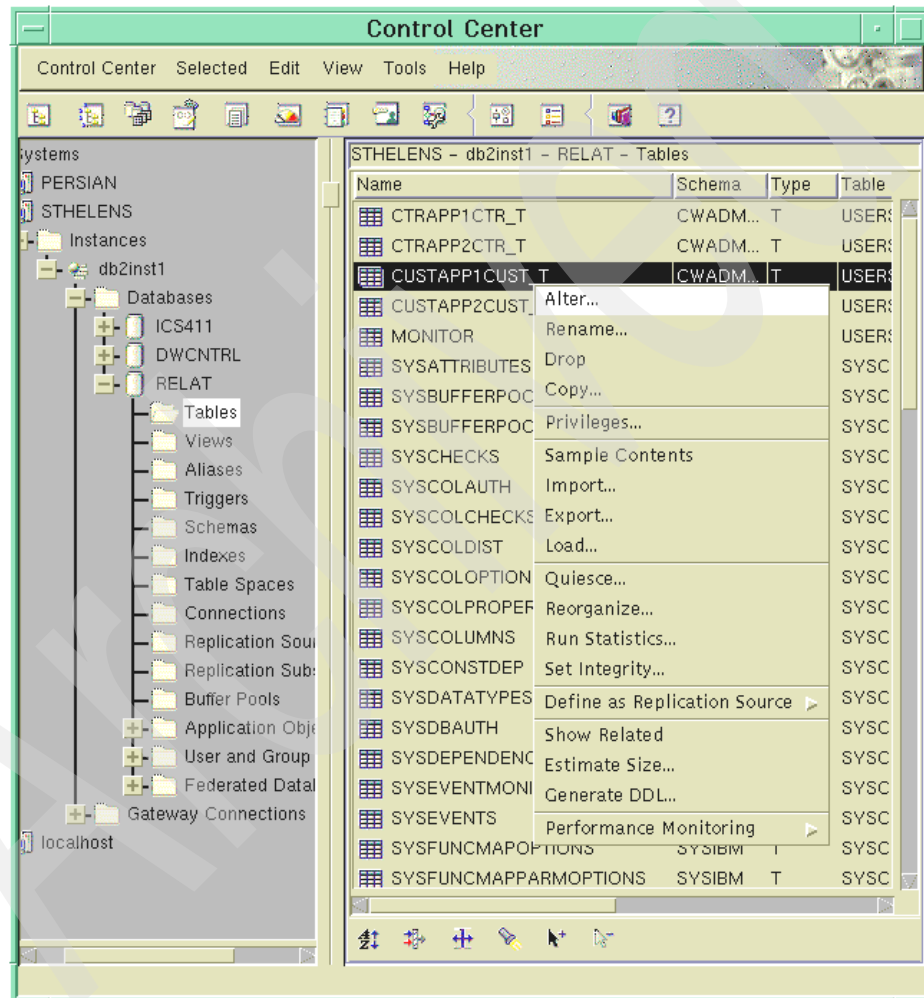


Figure 9-52 Exporting a table



- ii. Enter an output file location and a message file location and click **OK**.

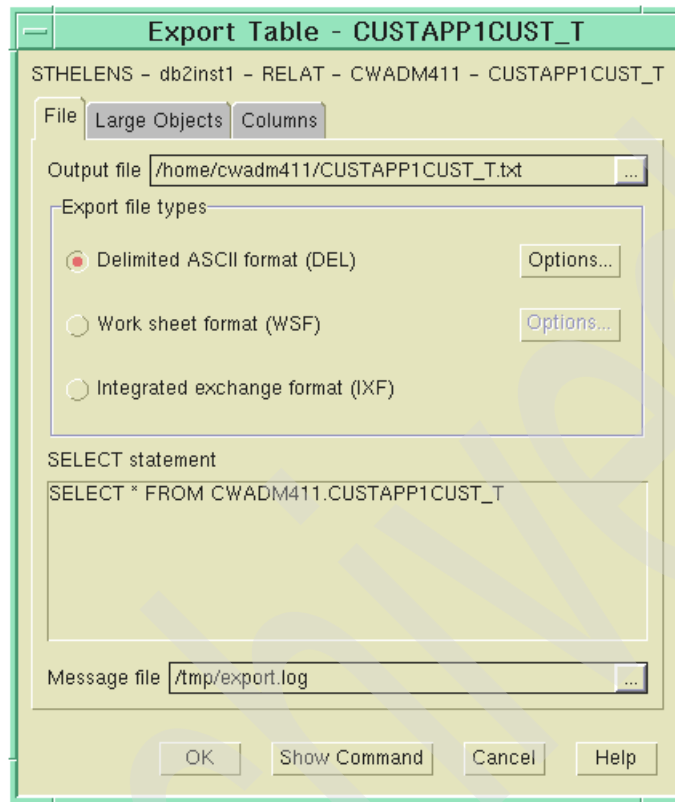


Figure 9-53 Completing the table export

- iii. In the Control Center, right-click the newly created relationship table and choose **Import**. Enter the location of the saved file as the import file and click **OK**.

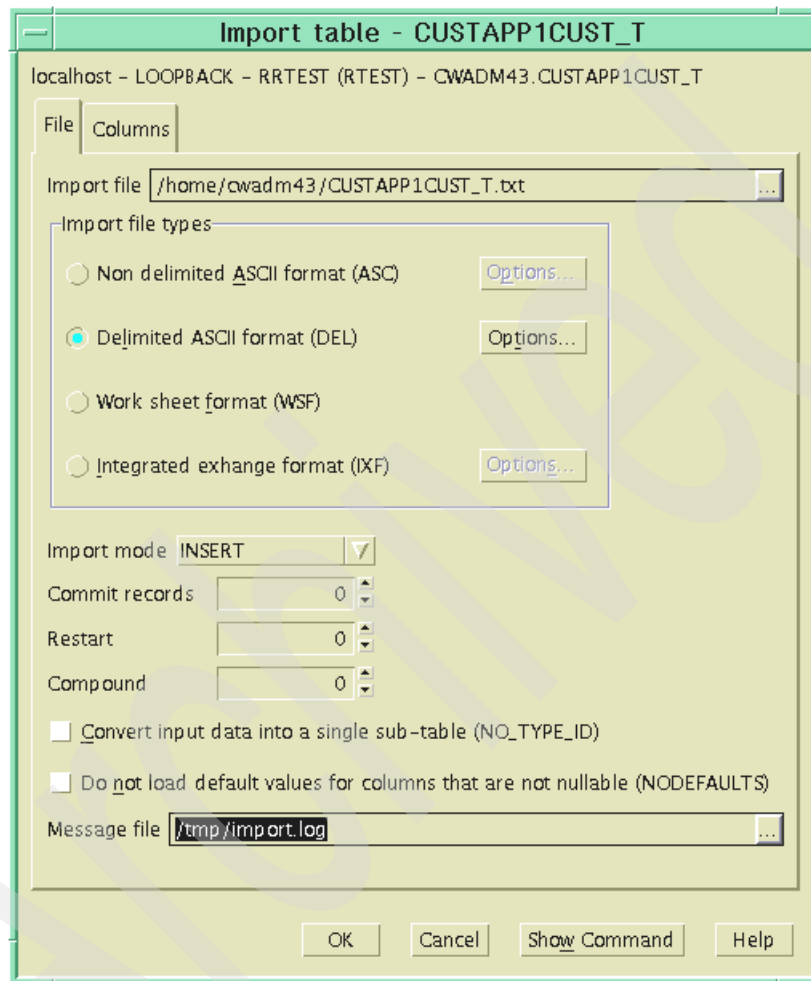


Figure 9-54 Importing the data

- b. Additional steps are needed to set the sequence number for each relationship. Here are the steps to set the sequence number when the relationships are stored in DB2:
  - i. Find the current highest ID for any relationship by examining the INSTANCEID column for any participant table in the relationship.

```
SELECT MAX(INSTANCEID) FROM DB2ADMIN.CUSTAPP1CUST_T
```

In this example, the query has returned the value 41 as the highest INSTANCEID.

- ii. Find the sequence variable name. The following command generates a SQL script that provides information about the relationship table. This information can also be obtained in the Control Center by right-clicking a database table and choosing **Generate DDL**.

```
db2look -dRELAT -tCUSTAPP1CUST_T -a -e -c -oOUTPUT.SQL
```

Examine the *OUTPUT.SQL* file to obtain the sequence variable name. Search for the text CREATE SEQUENCE.

```
CREATE SEQUENCE "DB2ADMIN"."CUSTOMER_1_0_0" AS INTEGER  
MINVALUE 1 MAXVALUE 2147483647 START WITH 1 INCREMENT BY 1  
CACHE 20 NO CYCLE NO ORDER;
```

In the previous example, the sequence variable name is CUSTOMER\_1\_0\_0. The sequence name is derived from the name of the relationship. In this case the relationship name is CUSTOMER.

- iii. Set the sequence variable to be at least one larger than the current number. In this example, the current highest INSTANCEID is 41 so the sequence is set to 42 with the following command.

```
alter sequence Customer_1_0_0 restart with 42
```

- c. Verify that the relationship has been moved successfully to the new database by examining the relationship using Relationship Manager.

### 9.4.3 Post-migration steps

This section describes the steps to take after migration completes and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the previously used password, assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.

2. Right-click the server name in the InterChange Server Component management view and select **Users and roles**.

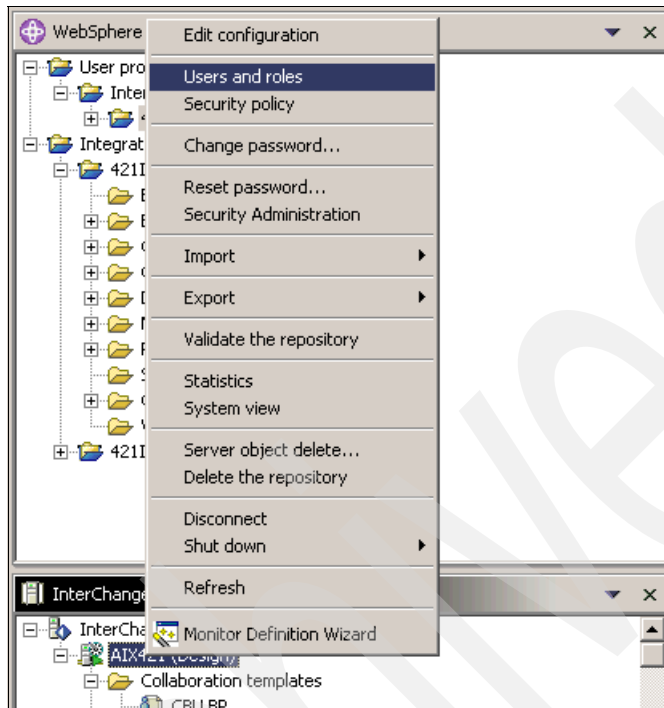


Figure 9-55 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User**.

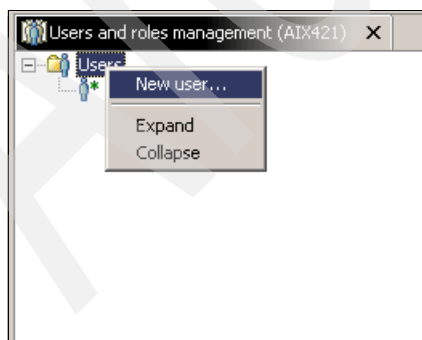


Figure 9-56 Selecting new user

4. The Create user window opens. Enter the name `admin` and the previous password, plus additional information if needed. Click **OK**.



The 'Create user' dialog box contains the following fields and controls:

- User name:** Text field containing 'admin'.
- Password:** Password field containing '\*\*\*\*'.
- Confirm password:** Password field containing '\*\*\*\*'.
- Distinguished name:** Empty text field.
- Common name:** Empty text field.
- Surname:** Empty text field.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

Figure 9-57 Creating a user

5. The new user is added to the Users and roles management window.

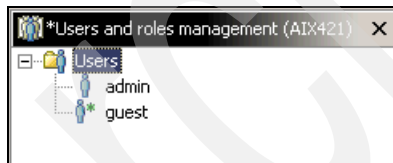


Figure 9-58 The newly created admin user

6. Right-click the newly created **admin** user and select **Add roles**.

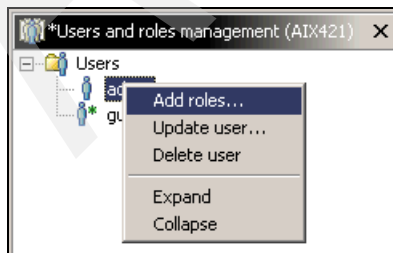


Figure 9-59 Add roles

7. The Role list window is displayed. Select **administrator** and click **OK**.

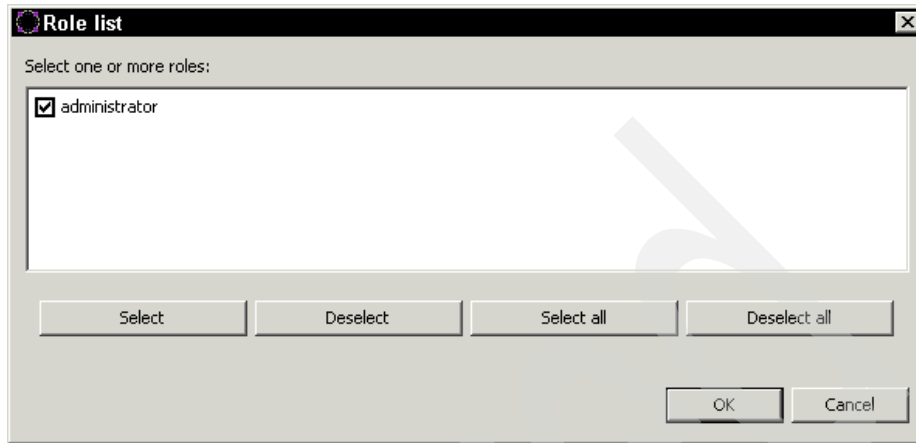


Figure 9-60 Role list

8. Close the Users and roles management window.

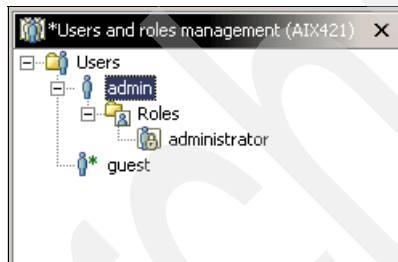


Figure 9-61 User admin with administrator role

9. Right-click the server again in the InterChange Component management view and select **Edit Configuration**. An editor opens to configure the server.
10. Click the **Security-RBAC** tab (Figure 9-62 on page 317) and apply the following settings:
- Select **Enable RBAC**.
  - Set **Server start user** to admin.
  - Set **Server start password** to the password for the admin user.
  - Check all other settings and modify if necessary.
  - Save the configuration to the server by clicking the save icon or selecting **File → Save**.

Figure 9-62 Server Configuration: Security-RBAC tab

11.Reboot WebSphere InterChange Server to activate security.

**Note:** The adapter agent defaults to the guest account if RBAC is activated for WebSphere InterChange Server and no user/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions about installing adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for further information.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the upgraded WebSphere InterChange Server system from development to production, you should perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 9-6 shows the scenarios that were tested in this migration and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 9-6 Test scenarios results

Test scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated

**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that WebSphere InterChange Server is re-started in Production mode.



## Uninstall (optional)

This section describes the steps to uninstall the previous version of WebSphere InterChange Server and the related software. The following steps list the order of uninstalling relevant software

1. Uninstall the VisiBroker ORB for version earlier than 4.2.2 using the uninstaller located in the VisiBroker product directory.
2. Uninstall JDK as the root user using `smi t`, only if it is not required by other products. For this release, the JDK is typically Java 2 SDK V.1.3.1. Contact the system administrator for further assistance.
3. Uninstall the IBM CrossWorlds InterChange Server using the uninstaller located in the product directory when it is certain that the previous version is no longer needed. After the uninstaller has completed, the product directory can be deleted.



# Migration procedures for WebSphere InterChange Server V4.2.2 to V4.3

This chapter provides instructions for migrating from an IBM WebSphere InterChange Server Version 4.2.2 to IBM WebSphere InterChange Server (WebSphere ICS) Version 4.3.0 environment.

This chapter provides four major sections for in-place database migration and without in-place database migration for both IBM AIX and Microsoft Windows operating system environments. For a detailed description of the issues related to in-place database migration and without in-place database migration, refer to Chapter 4, “General planning” on page 53.

Each major section is divided into subsections describing pre-migration procedures, migration procedures, and post-migration procedures.

## 10.1 In-place database migration: V4.2.2 to V4.3 on Windows

This section describes an in-place database migration scenario of WebSphere InterChange Server Version 4.2.2 to WebSphere InterChange Server Version 4.3.0. The complete repository from Version 4.2.2 is reused in this scenario. This includes component definitions, failed flows, and relationships.

For the full description of an in-place database migration, refer to Chapter 4, “General planning” on page 53.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 10.1.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system.

#### **Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least Version 4.2.1.7.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can only be used with US English character set (umlauts can not be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67.

For more information, view the relevant publications at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

#### **Put the environment in a quiescent state**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.

- All data is consistent between the integrated applications.

To put WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator.
2. Resolve transactional status of transactional collaborations using the Flow Manager of the WebSphere Business Integration Toolset. Transactional collaborations that are in transactional status can be migrated; however, it is recommended that they be resolved in this pre-migration step.
3. Resubmit failed events or discard the events using the Flow Manager of the WebSphere Business Integration Toolset. Failed flows can be migrated, but it is recommended that they be resolved in this pre-migration step.
4. After all events are processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in System Manager. Ensure that the in-process and pending event counter is 0.

## **Backing up and shutting down the components**

This section describes the steps needed to shut down and take a backup. It is recommended that you take a backup that enables the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3.0. Additional information about backing up the system can be found in the System Administration Guide.

There are two types of data in the WebSphere InterChange Server system: static and dynamic.

Static data includes:

- WebSphere InterChange Server repository (except for relationship tables)
- Custom collaboration components, such as Java class files and message files
- Custom connectors
- Map components, including map design files, Java class files, and message files

Dynamic data includes:

- Relationship database tables
- Work-in-progress tables and transaction database tables
- WebSphere MQ message data
- Log files (as desired for historical information)

These are the steps to perform the backup:

1. Back up the current InterChange Server repository using the **repos\_copy** utility.

For example, suppose the Interchange Server instance is WIN422 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository422.jar. This file is created in the directory where the **repos\_copy** command was run.

```
repos_copy -sWIN422 -uadmin -pnull -oRepository422.jar
```

**Note:** A **repos\_copy** may not be required during an in place-migration; however, it is still a recommended practice for backing up the current definitions of the WebSphere InterChange Server repository.

2. To export existing user projects, connect System Manager to the running WebSphere InterChange Server instance and follow the steps in System Manager:
  - a. Right-click **User Projects** and select **Export Solution**.
  - b. Select all user projects that are to be exported and enter a destination path.

**Restriction:** As we write this book, a product defect causes the Export Solution utility to not work properly. This defect has been reported and assigned tracking number CR60456. Visit the support site to see if a fix for this problem number is available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and choose **Export to Repository File**. User projects must be re-created from the integration component library after the migration.

The *Implementation Guide for WebSphere InterChange Server* has more information about this process. Go to:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Development** → **PDFs** and click **System implementation** in the right-side pane.

3. In System Manager, right-click the WebSphere InterChange Server name and choose **Shutdown Gracefully**.
4. Back up the product directory as indicated by the %CROSSWORLDS% environment variable. By default this is C:\IBM\WebSphere\ICS.
5. Arrange for a database administrator to back up the database. This must be a complete backup, including the schema information and stored procedures.
6. Shut down the Persistent Name Server.
7. Arrange for a WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.
8. Save the environment settings. In particular, the PATH, CLASSPATH, and CROSSWORLDS environment variables may be modified during the installation of WebSphere InterChange Server 4.3. The current environment settings can be viewed with the **set** command. The following example saves the environment settings to a file called env.txt.

```
set > env.txt
```

## 10.1.2 Migration steps

This section describes the main steps required for the migration, including:

- ▶ Upgrading the related software to the required level
- ▶ Installing and configuring the WebSphere InterChange Server 4.3
- ▶ Completing the upgrades of WebSphere InterChange Server components

### Preparing the database system

The database used for the repository with WebSphere InterChange Server 4.2.2 may not meet the minimum database level required by Version 4.3. If upgrading the database to the supported level is necessary, ensure afterward that it has the proper configuration for WebSphere InterChange Server 4.3.

- ▶ Check the *IBM WebSphere InterChange Server Installation Guide for Windows* for the proper database settings for V4.3. For example, when using DB2, a change in the database settings is that the applheapsz parameter must be set to at least 4096.

Read or download the *Installation Guide for Windows* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **Windows** in the right-side pane.

- Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To test that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run the following command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been successfully created. After the stored procedure has been successfully created, it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 Fix Pack 7 (formerly known as CSD 07). If failed flows are being migrating that involve MQ Delivery Transport, the queue messages must be preserved when upgrading WebSphere MQ. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Prepare the Web-based System Monitor

If WebSphere Application Server is at version 5.0.2.4 or 5.1 before running the WebSphere InterChange Server installer, the installer program automatically installs and configures System Monitor and uninstalls the previous version of the ICSMonitor application. This feature is available on Windows only when WebSphere Application Server is located on the same machine as the WebSphere InterChange Server. The WebSphere Application Server must be at the supported level before installing WebSphere InterChange Server.

## Uninstall software

This section describes the uninstall procedures for the WebSphere InterChange Server and related software.

1. Uninstall WebSphere InterChange Server and WebSphere Business Integration Adapters Windows services (if any) using the following command:

```
cwservice -xr -sServiceName
```



2. Uninstall WebSphere InterChange Server. During the uninstall process, all files can be removed safely because of the product directory backup you made. After the uninstaller has completed, the product directory can be deleted.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99.

**Note:** During the installation process, use the same WebSphere InterChange Server name as in the previous version. Remember, the WebSphere InterChange Server name must not contain dots (.).

If a decision was made to have the installer automatically install System Monitor, confirm that the System Monitor has been installed successfully. Examine the file `%CROSSWORLDS%/WBSM/configlog` and search for the message `Application ICSMonitor installed successfully`.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is prompted to configure the WebSphere InterChange Server, as we demonstrate in the following pictures and examples. The IBM WebSphere InterChange Server Configuration wizard can be run after the installation has finished by invoking:

```
%CROSSWORLDS%\bin\ICSConfig.bat
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all configuration settings must be entered in the `Options.txt` file that is delivered with the installer. For detailed information, refer to the Installation Guide for Windows.

The following figures show the configuration values used by the redbook team during migration:

1. Figure 10-1 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

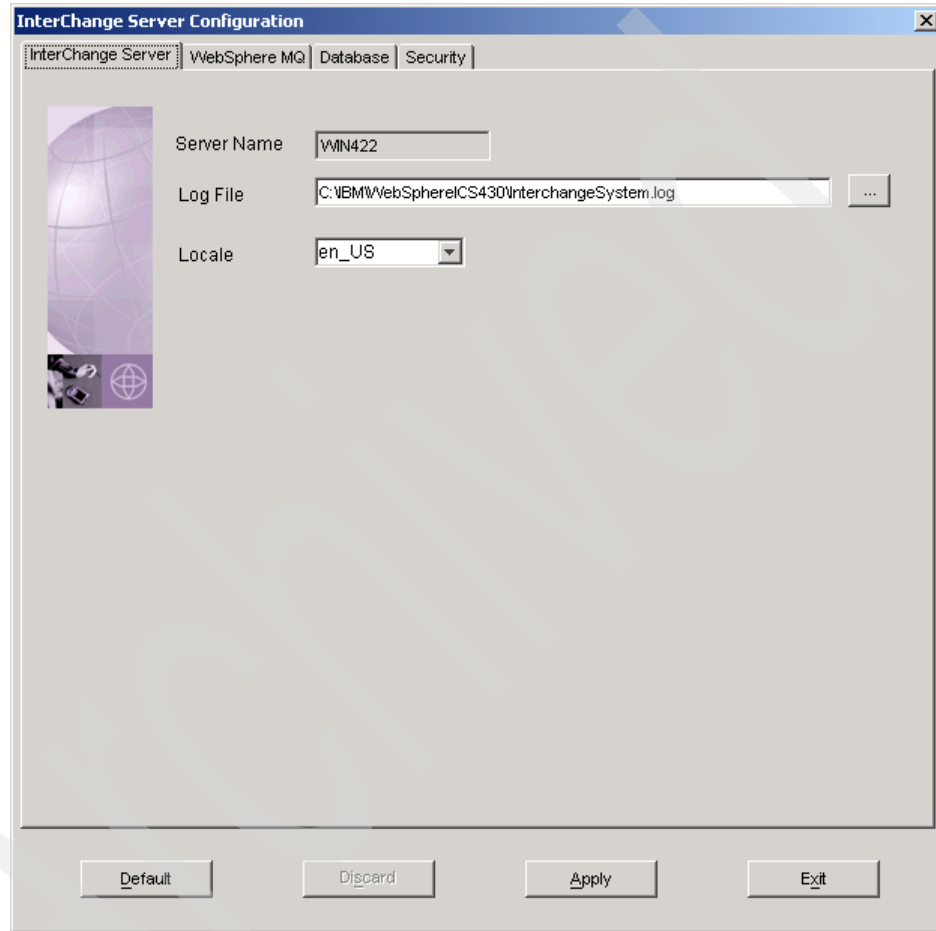
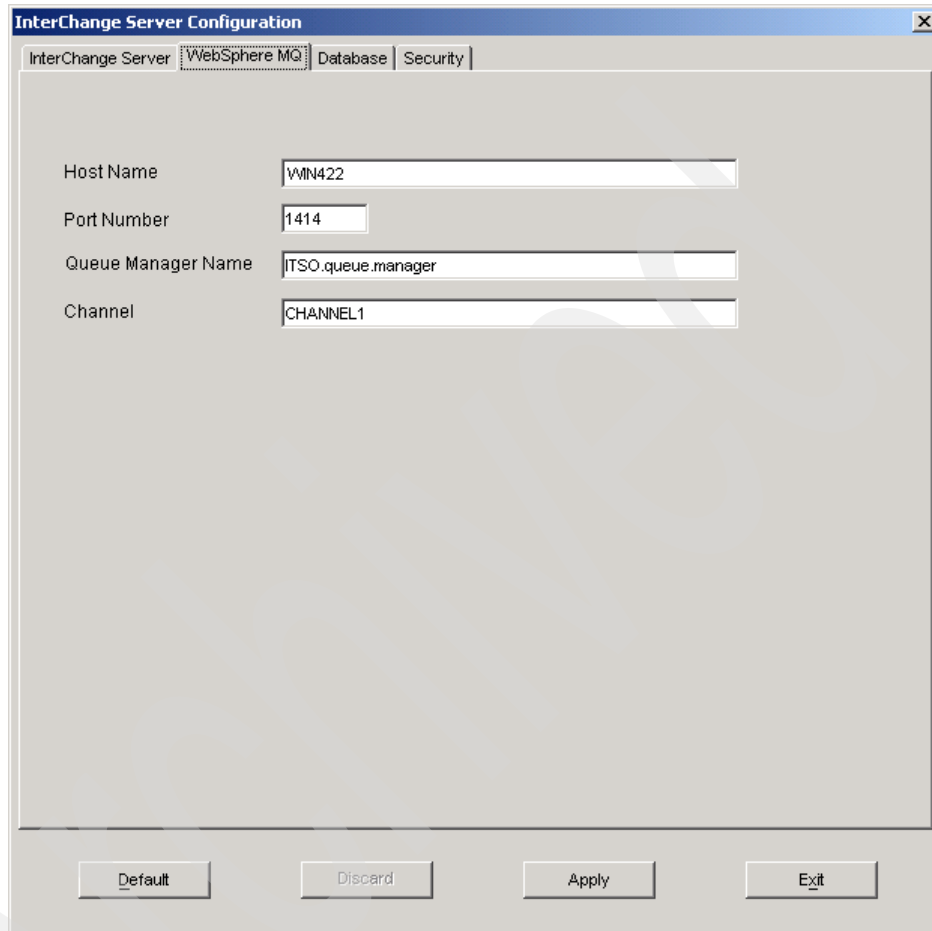


Figure 10-1 InterChange Server configuration: InterChange Server tab

2. Figure 10-2 shows the WebSphere MQ configuration tab. Make sure the settings for WebSphere InterChange Server 4.2.2 are entered here.



The image shows a dialog box titled "InterChange Server Configuration". It has four tabs: "InterChange Server", "WebSphere MQ", "Database", and "Security". The "WebSphere MQ" tab is selected. Inside the dialog, there are four text input fields with labels to their left: "Host Name" with the value "WIN422", "Port Number" with the value "1414", "Queue Manager Name" with the value "ITSO.queue.manager", and "Channel" with the value "CHANNEL1". At the bottom of the dialog, there are four buttons: "Default", "Discard", "Apply", and "Exit".

Field	Value
Host Name	WIN422
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 10-2 InterChange Server configuration: WebSphere MQ tab

3. The third tab is for database configuration. Enter the connection information from WebSphere InterChange Server 4.2.2 and apply the following settings:
  - Max Connections: unlimited, if possible. For the purposes of this book, the Max Connections parameter was set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Database' tab selected. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab contains several sections for configuring database connectivity and monitoring.

**Database Connectivity:**

- Database Driver: DB2 (dropdown)
- Max Connections: 75 (spin box) with an 'Unlimited' checkbox.
- Max Pools: 50 (spin box).
- Idle Timeout: 2 (spin box).

**Event Management:**

- Host Name: (empty text box)
- Login: db2admin (text box)
- Database: cwrepos (text box)
- Password: (masked with asterisks) with a 'Change' button.
- Max Connections: (empty text box) with an 'Unlimited' checkbox.
- Port Number: (empty text box)

**Transactions:**

- Host Name: (empty text box)
- Login: db2admin (text box)
- Database: cwrepos (text box)
- Password: (masked with asterisks) with a 'Change' button.
- Max Connections: (empty text box) with an 'Unlimited' checkbox.
- Port Number: (empty text box)

**Repository:**

- Host Name: (empty text box)
- Login: db2admin (text box)
- Database: cwrepos (text box)
- Password: (masked with asterisks) with a 'Change' button.
- Max Connections: (empty text box) with an 'Unlimited' checkbox.
- Port Number: (empty text box)

**Flow Monitoring:**

- Host Name: (empty text box)
- Login: db2admin (text box)
- Database: cwrepos (text box)
- Schema Name: (empty text box)
- Password: (masked with asterisks) with a 'Change' button.
- Port Number: (empty text box)
- Max Connections: (empty text box) with an 'Unlimited' checkbox.

At the bottom of the dialog are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 10-3 Interchange Server configuration: Database tab

4. The fourth tab is for security configuration (Figure 10-4). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information. For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the database and connection information from the Database tab for the Security database.

The screenshot shows the 'InterChange Server Configuration' window with the 'Security' tab selected. The 'User Registry Type' is set to 'Repository'. Under the 'Repository' section, the 'Host Name' is empty, 'Database' is 'CWREPOS', 'Login' is 'clb2admin', 'Password' is masked with asterisks, and 'Max Connections' is set to 'Unlimited'. A 'Change' button is next to the password field. At the bottom are 'Default', 'Discard', 'Apply', and 'Exit' buttons.

InterChange Server Configuration							
InterChange Server		WebSphere MQ		Database		Security	
User Registry Type							
User Registry		Repository					
Repository							
Host Name				Login		clb2admin	
Database		CWREPOS		Password		*****	
Max Connections		<input checked="" type="checkbox"/> Unlimited		Port Number			
				Change			
Default Discard Apply Exit							

Figure 10-4 Interchange Server configuration: Security tab

To save the settings, click **Apply**. To exit the configurator, click **Exit**.

If the WebSphere InterChange Server is installed as a Windows Service, restart Windows as instructed.

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMS and collaboration directories from WebSphere InterChange Server Version 4.2.2 to the new installation directory for Version 4.3.0. Be sure to include all subdirectories.
2. Apply any custom configuration settings to `$CROSSWORLDS/InterchangeSystem.cfg`.

For example, to disable `DEADLOCK_DETECTOR_CHECK` for collaborations, edit the config file code shown in Example 10-1 with a suitable editor for XML.

### *Example 10-1 Disable DEADLOCK\_DETECTOR\_CHECK*

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>>false</tns:reposController>
<tns:reposAgent>>false</tns:reposAgent>
<tns:localConfig>>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change `true` to `false` in the line starting with `<tns:value>`.

3. Include any external code and .jar files in `CLASSPATH` environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in `%CROSSWORLDS\bin\start_server.bat`. The external classes are first added to a new variable `EXT_LIB`. Then the `EXT_LIB` is appended to the `CWCLASSES` variable. Refer to the following excerpt from `start_server.bat`:

```
REM external classes added here
set EXT_LIB=C:\Sample.jar
CWCLASSES="%CROSSWORLDS%\lib\crossworlds.jar;%CROSSWORLDS%\lib\cworio
n.jar;%CROSSWORLDS%\lib\CodeGeneration.jar;%EXT_LIB%
```

4. Modify and verify the environment variables as documented. Check that the following three environment variables are set properly:

<b>CROSSWORLDS</b>	Contains the location of the 4.3.0 installation directory.
<b>MQ_LIB</b>	Contains the location of WebSphere MQ Java /lib directory.

## **PATH**

Contains the /bin directory of IBM Java Development Kit 1.4.2. Ensure that this bin directory precedes any other installed JDKs. Confirm that the compiler is installed by invoking **javac** from the command line. The usage instructions for **javac** should be returned.

## **Starting the new server**

When the server is started for the first time after the installation, the WebSphere InterChange Server repository tables are upgraded or created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database server
2. Run the PersistentNameServer.bat file, located in %CROSSWORLDS%\bin\.
3. Start the WebSphere InterChange Server in design mode: Edit the Properties of the IBM WebSphere InterChange Server shortcut in the **Start** menu. Add the -design option to the Target field. Run the shortcut after modification.

WebSphere InterChange Server upgrades existing tables, creates new tables for new features, and logs messages to the InterChangeSystem.log file located in the product directory. The message Interchange Server is Ready indicates that WebSphere InterChange Server has started successfully.

## **Migrating repository content to WebSphere ICS 4.3.0**

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. To complete this task.

1. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, import the component libraries using exported integration component library files:

1. Create a new integration component library.
2. Right-click the newly created integration component library and select **Import Repository from File**. Choose the repository file created during the backup procedure. Repeat these steps if there are multiple integration component libraries to import.

The *Implementation Guide for WebSphere InterChange Server* contains further information about this process.

3. Include any custom .jar files in the CLASSPATH environment variable for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH setting for the Toolset (Figure 10-5). Add the location of any custom .jar files to this list.

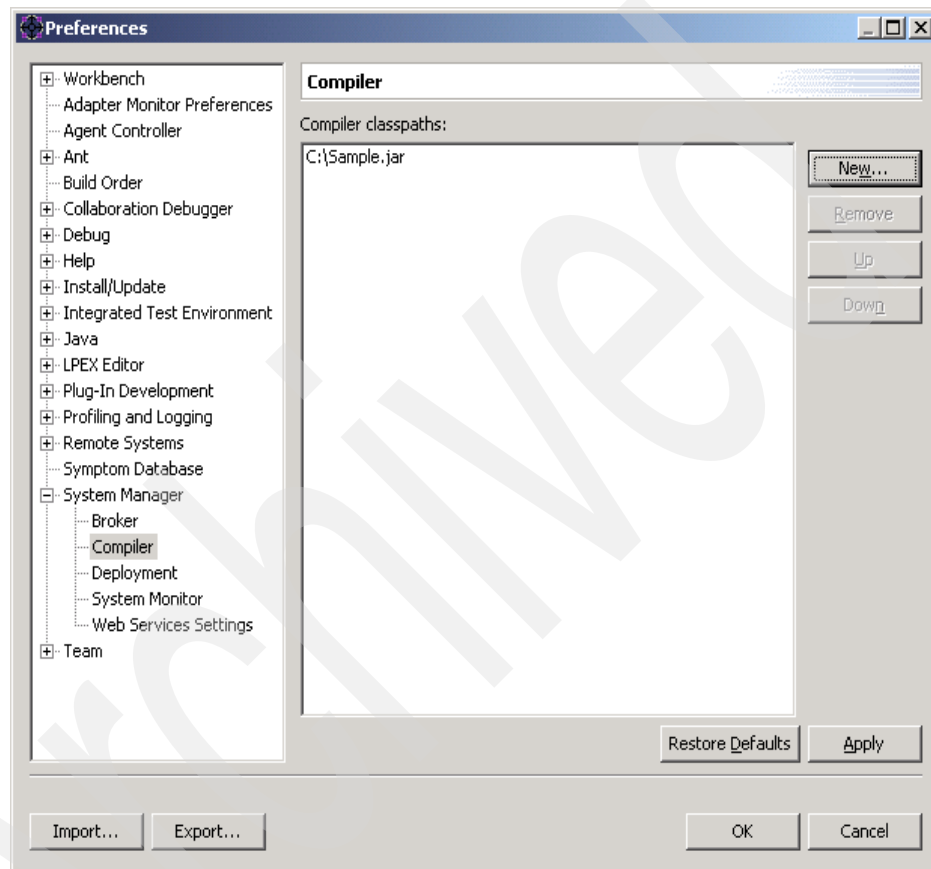


Figure 10-5 Setting classpath for the Toolset



4. Set the compiler compliance to Java 1.4: Navigate to **Java** → **Compiler**. Under the Compliance and classfiles tab, change the compiler compliance level to **1.4** (Figure 10-6). If a dialog box indicating that the compiler settings have changed is displayed, choose **No** to *not* rebuild files at this time.

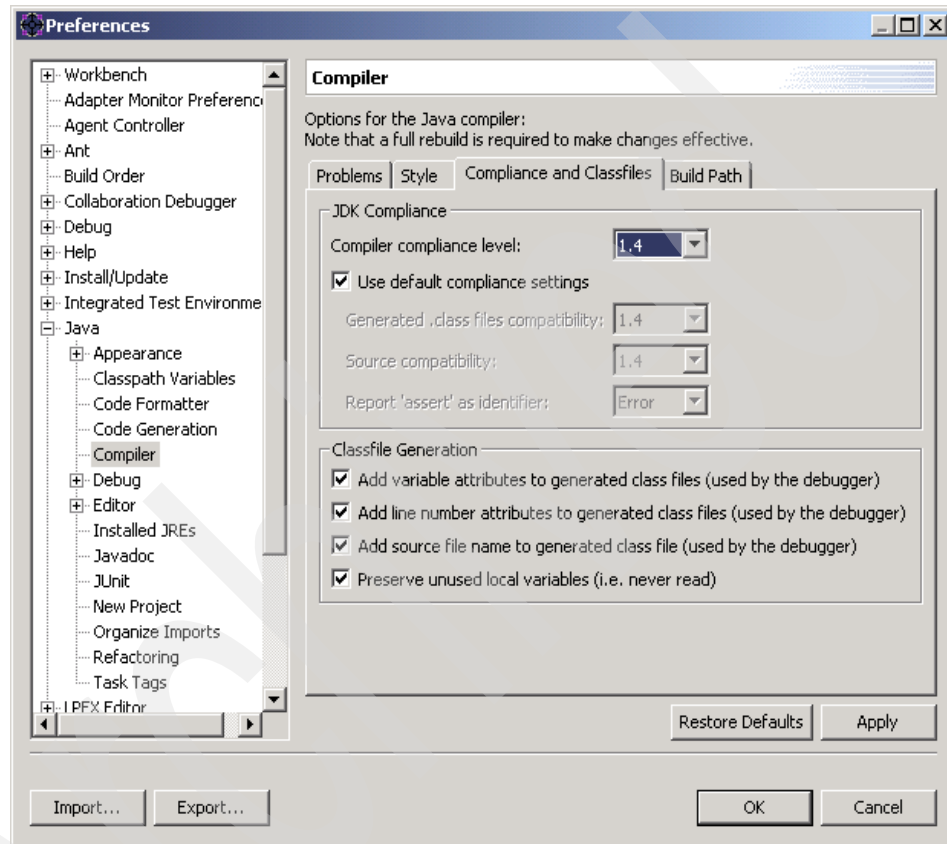


Figure 10-6 Setting the compiler compliance level

5. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the use of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in the previous step.
6. Compile all of the maps and templates in the integration component library and examine the console for any error messages.

7. Reconfigure any of the other components that require modification:
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions, examine the Port and Triggering Events tab. For the BO Type that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to **<None>** for all verbs.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
8. Create a new user project and populate the component from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for details.

**Note:** To make use of all administrative features for maps and collaborations, the maps and templates must be deleted and redeployed. Typically, each component must be placed in an inactive state before it can be overwritten. To simplify the process and avoid any unresolved dependencies, the whole repository is deleted in step 9 and redeployed in step 10. Additionally, recompilation ensures that the code is optimized and Java 1.4 compliant.

However, redeploying the maps and templates, because the Java Runtime Environment is backward-compatible and poses no problems executing map and template class files compiled in WebSphere InterChange Server 4.2.2.

If a decision is made to reuse the previous maps and collaborations, only deploy the components that have been updated.

9. In the System Manager, delete the repository by right-clicking the WebSphere InterChange Server name in the Interchange Server component management view and select **Delete the repository**.
10. Deploy the user project:
  - a. Right-click the user project and select **Deploy user project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This re-creates the stored procedures and does not overwrite existing data.

### 10.1.3 Post-migration steps

This section describes the steps that are taken after migration and after the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the previous password assigned to the administrator role. The following steps and figures show how to do this:

1. In the System Manager, connect to the WebSphere InterChange Server.

2. Right-click the server name in the InterChange Server component management view and select **Users and roles** (Figure 10-7).

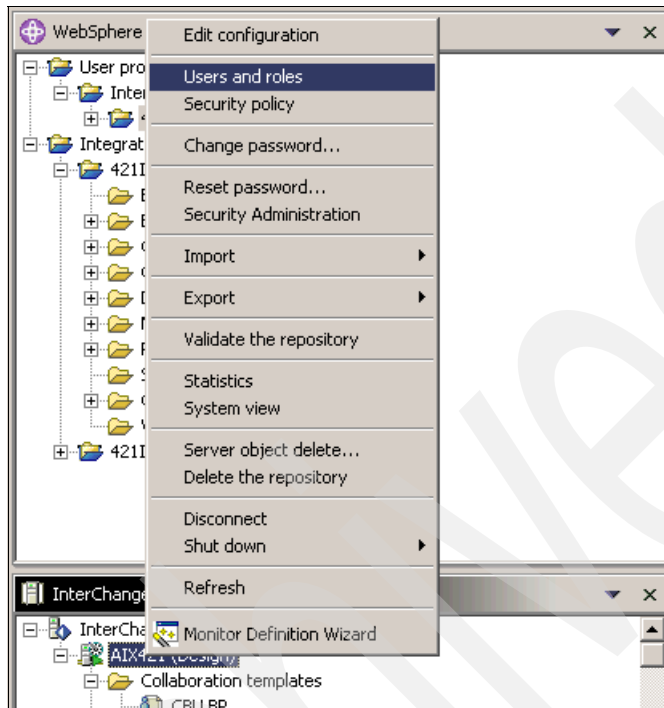


Figure 10-7 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User** (Figure 10-8).

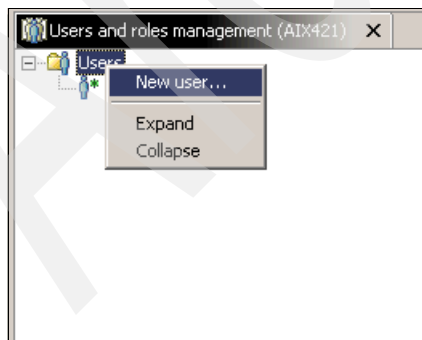


Figure 10-8 Select New user

4. In the Create user window (Figure 10-9), enter the name admin and the previously used password. Add additional information if needed and click **OK**.

A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' with 'admin' entered, 'Password' with '\*\*\*\*', 'Confirm password' with '\*\*\*\*', 'Distinguished name' (empty), 'Common name' (empty), and 'Surname' (empty). At the bottom right are 'OK' and 'Cancel' buttons.

Create user

User name  
admin

Password  
\*\*\*\*

Confirm password  
\*\*\*\*

Distinguished name

Common name

Surname

OK Cancel

Figure 10-9 Creating a user

5. The new user is added to the User and Roles management window (Figure 10-10).

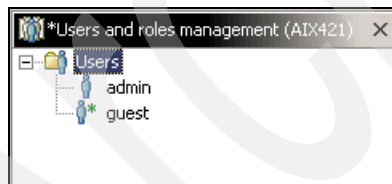


Figure 10-10 The newly created admin user

6. Right-click the new admin user and select **Add roles** (Figure 10-11).

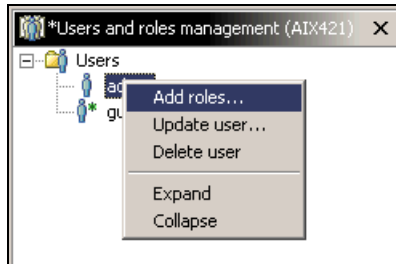


Figure 10-11 Add roles

7. The Role list window is displayed (Figure 10-12).

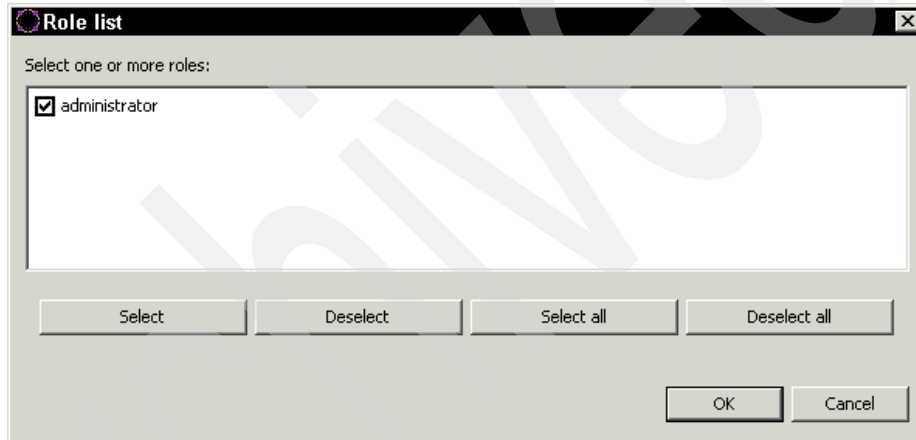


Figure 10-12 Role list

8. Select **administrator** and click **OK**. Close the Users and roles management window.



Figure 10-13 User admin with administrator role

9. Right-click the server again and choose **Edit Configuration**. This opens an editor for configuring the server.

10. Click the **Security-RBAC** tab (Figure 10-14) and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Enter admin for **Server start user**.
  - c. For **Server start password** enter the password for the admin user.
  - d. Check all other settings in the tab and modify if needed.
  - e. Save the configuration to the server by clicking the save icon or selecting **File** → **Save**.

Figure 10-14 Server Configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** Adapter agents default to the guest account if the RBAC is activated for WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3, a WebSphere Business Integration Adapter V2.6 must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter and includes the Visual Test Connector. For more about installing adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the migrated InterChange Server system from development to production, it is recommended that you perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 10-1 shows the scenarios that were tested in this migration and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be migrated successfully if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 10-1 Test scenarios results

Test scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated
Web-based System Monitor	Successfully migrated

A failed flow is considered to be successfully migrated if the event can be resubmitted and completed after migration. Table 10-2 on page 343 shows the results of the migration of failed flows.



Table 10-2 Migrated failed flows results

Failed flows tested	Results
Failed flow for IDL	Failed because of hierarchical Business Object structure
Failed flow for MQ	Failed because of hierarchical Business Object structure
Failed flow for JMS	Failed because of hierarchical Business Object structure
Transactional collaboration in transactional status (compensation call failed)	Successfully migrated with flat Business Object structure
Transactional collaboration failed flow (compensation call successful)	Successfully migrated with flat Business Object structure
LLBP waiting asynchronous inbound call	Failed because of hierarchical Business Object structure
LLBP failed flow	Failed because of hierarchical Business Object structure
Collaboration group failed at Destination 1	Successfully migrated with flat Business Object structure
Collaboration group failed at Destination 2	Successfully migrated with flat Business Object structure

It can be seen in Table 10-2 that the majority of the failed flows could not be migrated successfully. In response to this, Change Request CR60367 was raised. The cause of this problem is that in WebSphere InterChange Server 4.2.2.4, a Child BO Verb with value *null* is set to CxIgnore, which is not expected by WebSphere InterChange Server 4.3.0. and is therefore rejected when resubmitted. WebSphere InterChange Server 4.3.0 has been changed to handle the CxIgnore case in a future version. It is recommended that failed flows be handled before migration until this problem has been resolved.

Check the support site to see whether a fix for the problems mentioned in this section are available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

**Note:** After all system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is restarted in Production mode.

## 10.2 Without in-place database migration for V4.2.2 to V4.3 on Windows

This section describes a without in-place database migration scenario and the migration of Version 4.2.2 to Version 4.3.0. WebSphere InterChange Server 4.3.0 is started with a new and empty repository. Repository content (component definitions) is exported from the original V4.2.2 and deployed in the Version 4.3.0 environment.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 10.2.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system.

#### **Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.2.2.4.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

#### **Put the environment in a quiescent state**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All the collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the Connector Configurator tool.
2. Resolve transactional status of transactional collaborations using the Flow Manager tool of WebSphere Business Integration Toolset. It is recommended that they be resolved in this pre-migration step as all status information is lost during the migration.
3. Resubmit failed events or discard the events using the Flow Manager tool of WebSphere Business Integration Toolset. It is recommended that they be resolved in this pre-migration step as failed flows are lost after migration.
4. When all events are processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in System Manager. Make sure the in-process and pending event counter is 0.

### **Backing up and shutting down components**

This section describes the steps for shutting down and taking a backup. It is a recommended practice to take a backup that will enable the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3.0.

Additional information about backing up the system can be found in the System Administration Guide. The WebSphere InterChange Server system has two types of data: static and dynamic.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current WebSphere InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is WIN422 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository422.txt. This file is created in the directory where the **repos\_copy** command is run.

```
repos_copy -sWIN422 -uadmin -pnull -oRepository422.jar
```

2. To export existing user projects, connect System Manager running on a connected Windows machine to the existing InterChange Server instance and take the following steps in System Manager:
  - a. Right-click **User Projects** and select **Export Solution**.
  - b. Select all user projects that to be exported and enter a destination path.

**Restriction:** As we write this book, a product defect prevents the Export Solution utility from working properly. This defect has been reported and assigned tracking number CR60456. Check the support site below to see whether a fix for this problem number is available.

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and select **Export to Repository File**. The *Implementation Guide for WebSphere InterChange Server* has more information about this process. The user projects must be re-created from the integration component library after the migration.

3. Shut down the WebSphere InterChange Server. In the System Manager, right-click the WebSphere InterChange Server name and choose **Shutdown Gracefully**.
4. Shut down the Persistent Name Server.
5. Back up the product directory indicated by the %CROSSWORLDS% environment variable. By default this is C:\IBM\WebSphere\ICS.
6. Arrange with the database administrator to back up the database. This must be a complete backup, including schema information and stored procedures.
7. Arrange for a WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.

8. Clear WebSphere MQ queues of any outstanding messages if the same queue manager is reused for Version 4.3.0. Old messages in the queues may be seen as new events, which could cause data inconsistencies. Refer to the WebSphere MQ documentation for more information.
9. Save the environment settings. In particular, the PATH, CLASSPATH, and CROSSWORLDS environment variables may be modified during the installation of WebSphere InterChange Server 4.3.0. The current environment settings can be viewed with the **set** command. The following example saves the environment settings to a file called env.txt:

```
set > env.txt
```

## 10.2.2 Migration steps

This section describes the main steps required for the migration including:

- ▶ Upgrading related software to the required level.
- ▶ Installing and configuring WebSphere InterChange Server 4.3.0.
- ▶ Completing the upgrades of WebSphere InterChange Server components.

### Preparing the database system

Prepare an empty database for the 4.3.0 repository: Create a new database following the steps in the Installation Guide, and ensure that the database has the proper configuration for WebSphere InterChange Server 4.3.0.

- ▶ Check the installation guide for the proper database settings for 4.3.0. For example, when using DB2, one change in the database settings is that the applheapsz parameter must be set to at least 4096.
- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be at Version 5.3.0.2 Fix Pack 7 (formerly known as CSD 07). Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Uninstall software

This section describes the uninstallation procedures of the WebSphere InterChange Server and related software:

1. Uninstall WebSphere InterChange Server and WebSphere Business Integration Adapters Windows Services (if any) using this command:  

```
cwservice -xr -sServiceName
```
2. Uninstall WebSphere InterChange Server. During the uninstall process, all files can be removed safely because of the backup of the product directory that was taken earlier. After the uninstaller has completed, the product directory can be deleted.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 for details.

**Note:** During the installation process, the WebSphere InterChange Server name may be changed. If so, the adapters and queues may have to be configured. You should use the same WebSphere InterChange Server name if possible.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following steps explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
%CROSSWORLDS%\bin\ICSConfig.bat
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all configuration settings must be entered in the file Options.txt, which is delivered with the installer. For detailed information, refer to the Installation Guide for Windows.

The following figures show the configuration values used by the redbook team during migration:

1. Figure 10-15 shows the InterChange Server tab, where the location and name of the log file are specified as well as the locale setting for WebSphere InterChange Server.

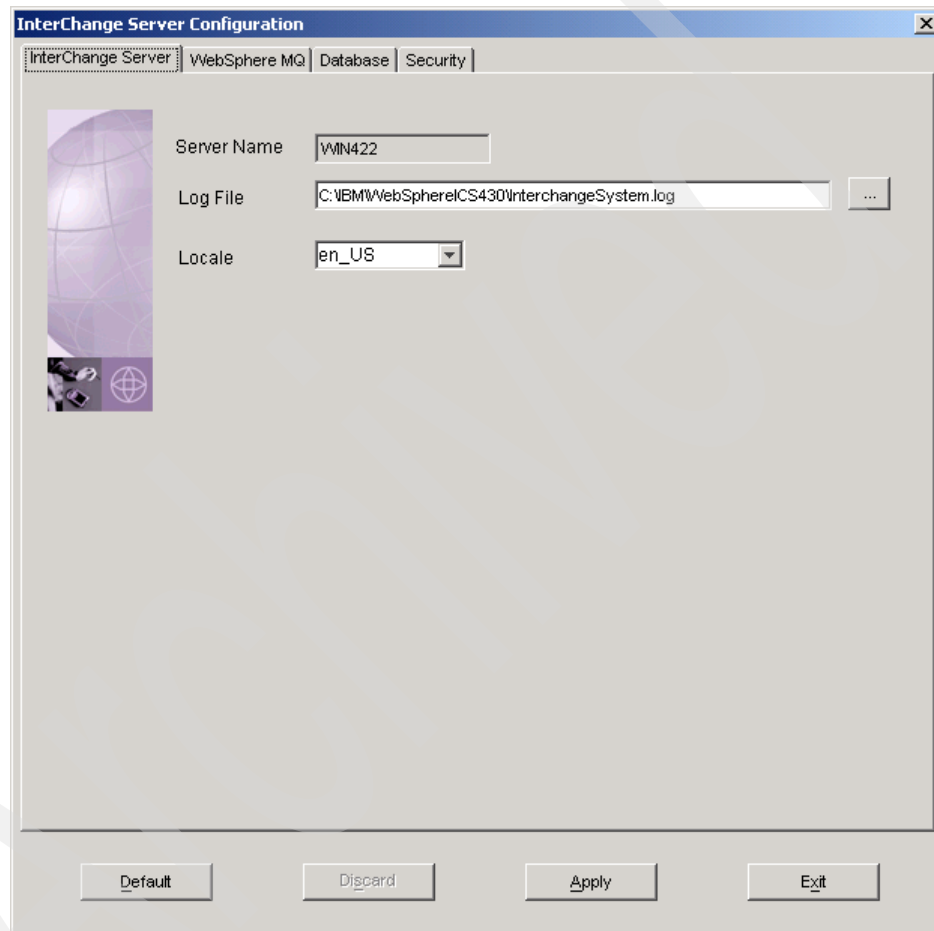
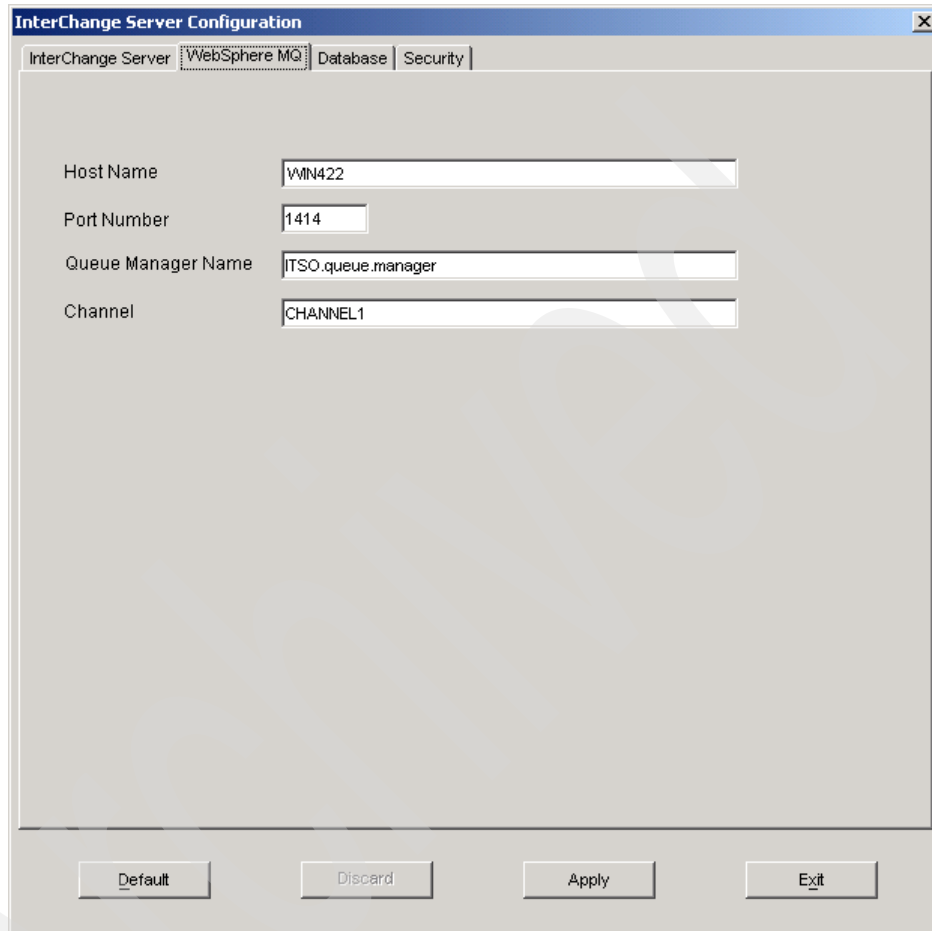


Figure 10-15 InterChange Server Configuration: InterChange Server tab

2. Figure 10-16 shows the WebSphere MQ configuration tab. Ensure that WebSphere MQ settings are entered here.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'WebSphere MQ' tab. The dialog box has a title bar with the text 'InterChange Server Configuration' and a close button (X). Below the title bar are four tabs: 'InterChange Server', 'WebSphere MQ' (which is selected), 'Database', and 'Security'. The main area of the dialog contains four labeled text input fields: 'Host Name' with the value 'WIN422', 'Port Number' with the value '1414', 'Queue Manager Name' with the value 'ITSO.queue.manager', and 'Channel' with the value 'CHANNEL1'. At the bottom of the dialog are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Field	Value
Host Name	WIN422
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 10-16 *InterChange Server Configuration: WebSphere MQ tab*



3. The third tab is for database configuration (Figure 10-17). Make sure to enter the connection information for the empty database created in the previous step and apply the following settings:
- Max Connections: unlimited, if possible. For the purposes of this book, the Max Connections parameter was set to 75. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Database' tab selected. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab contains several sections for configuring database connectivity and monitoring.

**Database Connectivity**

Database Driver	DB2	Max Connections	75	<input type="checkbox"/> Unlimited
		Max Pools	50	
		Idle Timeout	2	

**Event Management**

Host Name		Login	db2admin	
Database	cwrepos	Password	*****	Change
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number	

**Transactions**

Host Name		Login	db2admin	
Database	cwrepos	Password	*****	Change
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number	

**Repository**

Host Name		Login	db2admin	
Database	cwrepos	Password	*****	Change
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number	

**Flow Monitoring**

Host Name		Login	db2admin	
Database	cwrepos	Password	*****	Change
Schema Name		Port Number		
Max Connections		<input checked="" type="checkbox"/> Unlimited		

At the bottom of the dialog are four buttons: Default, Discard, Apply, and Exit.

Figure 10-17 Database configuration

4. The fourth tab is for security configuration (Figure 10-4 on page 331). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.

The screenshot shows the 'InterChange Server Configuration' window with the 'Security' tab selected. The 'User Registry Type' is set to 'Repository'. The 'Repository' section contains the following fields: 'Host Name' (empty), 'Login' (clb2admin), 'Database' (cwrepos), 'Password' (masked with asterisks), 'Max Connections' (empty), 'Unlimited' (checked), and 'Port Number' (empty). A 'Change' button is next to the password field. At the bottom are buttons for 'Default', 'Discard', 'Apply', and 'Exit'.

InterChange Server Configuration							
InterChange Server		WebSphere MQ		Database		Security	
User Registry Type							
User Registry		Repository					
Repository							
Host Name		Login		clb2admin			
Database		Password		*****		Change	
Max Connections		Unlimited		Port Number			

Figure 10-18 Interchange Server Configuration: Security tab

If the WebSphere InterChange Server is installed as a Windows Service, restart Windows as instructed.

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.

For example, to disable DEADLOCK\_DETECTOR\_CHECK for collaborations, edit the config file code shown in Example 10-2 using a suitable editor for XML.

*Example 10-2 Disable DEADLOCK\_DETECTOR\_CHECK*

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>false</tns:reposController>
<tns:reposAgent>false</tns:reposAgent>
<tns:localConfig>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change true to false in the line starting with <tns:value.

2. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in %CROSSWORLDS\bin\start\_server.bat. The external classes are first added to a new variable EXT\_LIB. Then the EXT\_LIB is appended to the CWCLASSES variable. Refer to the following example from start\_server.bat:

```
REM external classes added here
set EXT_LIB=C:\Sample.jar
CWCLASSES="%CROSSWORLDS%\lib\crossworlds.jar;%CROSSWORLDS%\lib\cworio
n.jar;%CROSSWORLDS%\lib\CodeGeneration.jar;%EXT_LIB%
```

3. Modify and verify the environment variables as documented. Confirm that the following three environment variables are set properly:

<b>CROSSWORLDS</b>	Contains the location of the 4.3.0 installation directory.
<b>MQ_LIB</b>	Contains the location of WebSphere MQ Java /lib directory.
<b>PATH</b>	Contains the /bin directory of IBM Java Development Kit 1.4.2. Ensure that this bin directory precedes any other installed JDKs. Confirm that the compiler is

installed by invoking **javac** from the command line.  
The usage instructions for **javac** should be returned.

## Starting the new server

When the WebSphere InterChange Server is started for the first time after the installation, the WebSphere InterChange Server repository tables are created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database server
2. Run the PeristentNameServer.bat file located in %CROSSWORLDS%\bin\.
3. Start the WebSphere InterChange Server in design mode by editing the **Properties** of the IBM WebSphere InterChange Server shortcut in the **Start** menu. Add the option -design to the **Target** field. Run the shortcut after modification.

WebSphere InterChange Server logs messages to the InterChangeSystem.log file located in the product directory. A message “Interchange Server is Ready” indicates that the WebSphere InterChange Server has started successfully.

## Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Import the previous repository into the tooling.
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, instead import the component libraries using exported integration component library files.

- ▶ Create a new integration component library.
- ▶ Right-click the newly created integration component library and select **Import Repository from File**. Choose the repository file that was created in the backup procedure. Repeat these steps if there are multiple Libraries to import. The *Implementation Guide for WebSphere InterChange Server* has more information about this process.

2. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH settings for the Toolset. Add the location of the custom .jar files to this list.

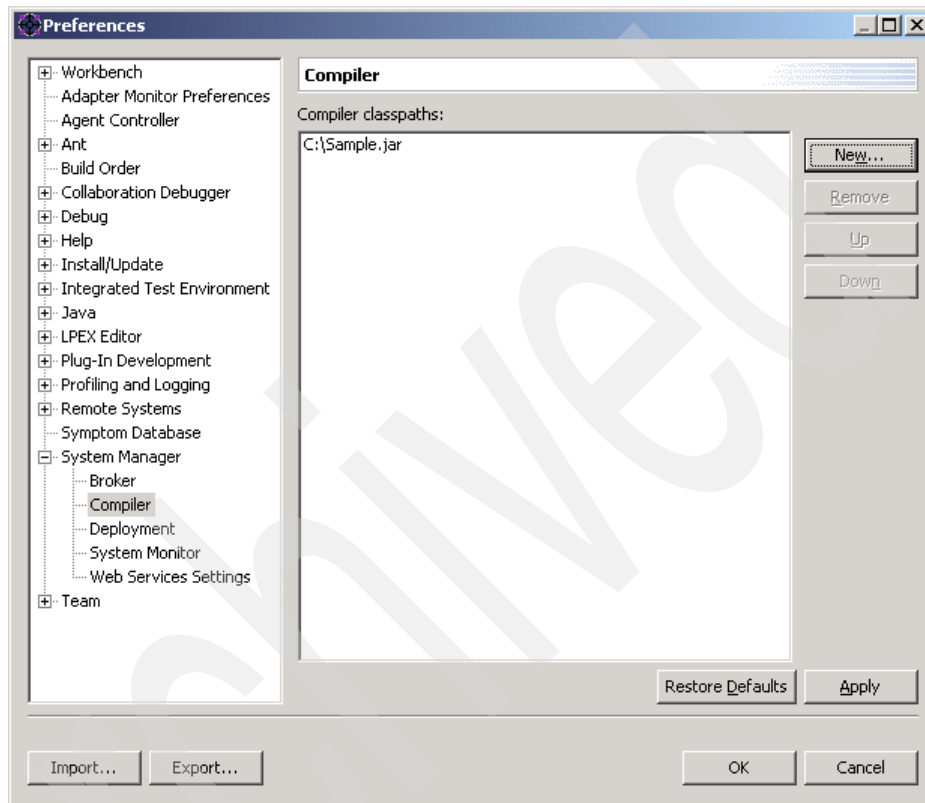


Figure 10-19 Setting the CLASSPATH for the Toolset

3. Set the compliance of the compiler to Java 1.4. Click **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag change the Compiler compliance level to **1.4**. A dialog box indicating that the compiler settings have changed may be displayed. Choose **No** to not rebuild files at this time.

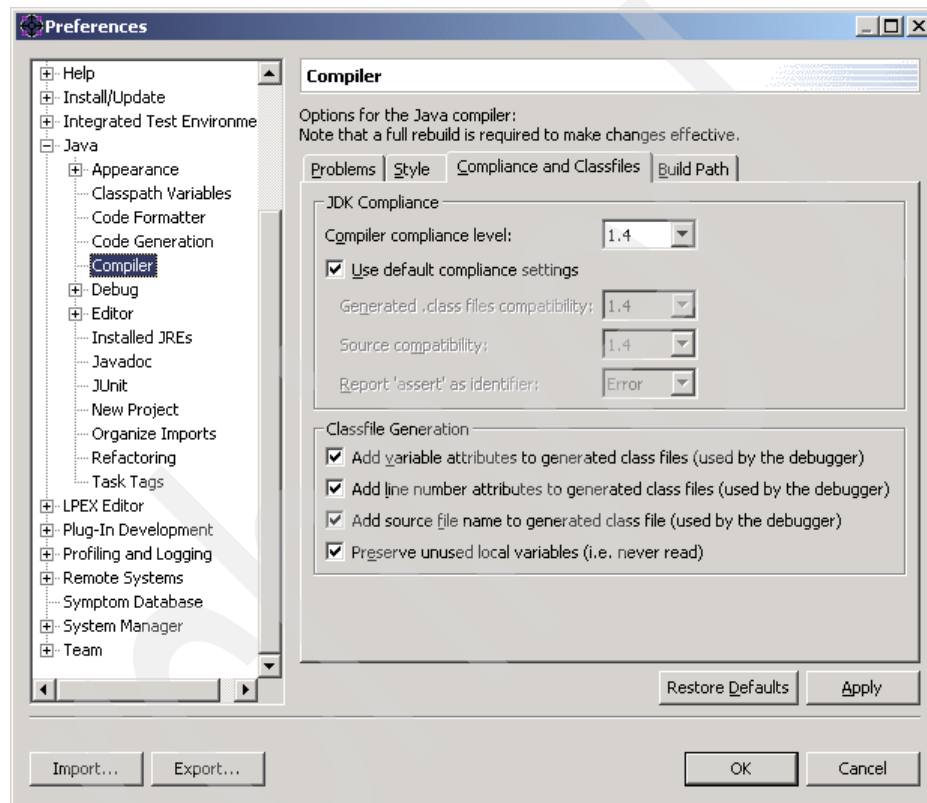


Figure 10-20 Setting the compiler compliance level

4. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the usage of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in a previous step.
5. Compile all maps and templates in the integration component library and examine the console for any error messages.

6. Reconfigure any of the other components that require modification:
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface. Refer to Chapter 5, “Technical impact of major design changes” on page 67 for more details.
  - c. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions, examine the Port and Triggering Events tab. For the **BO Type** that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to <None> for all verbs.
  - d. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.

**Note:** If Import Solution was not used, create a new user project and populate the user project with the components from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for details.

7. Deploy the user project:
  - a. Right-click the user project and select **Deploy user project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. If the location of the relationship tables has changed, select **Create Schema**.
  - f. Click **Finish** to start the deployment.
8. Restart the WebSphere InterChange Server if prompted.

9. Completing the relationship migration:

If a new relationship location is used, the relationship data from the 4.2.2 relationship tables can be migrated.

- a. For both static and dynamic relationships, copy the data from each relationship table to the new relationship table. Consult with a database administrator for more details. The steps to complete this process using DB2 are:
  - i. In the Control Center, right-click the original relationship table and choose **Export**.

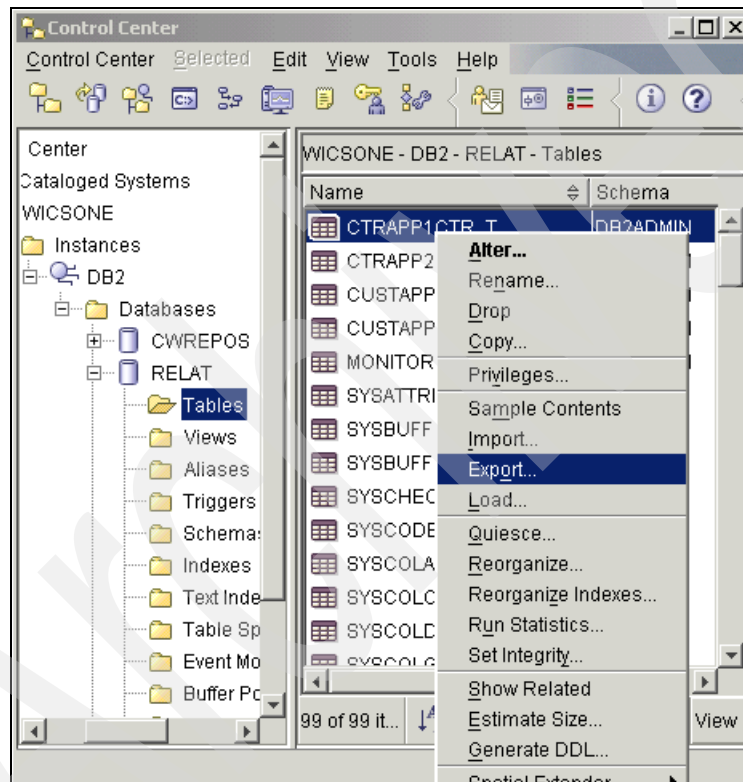


Figure 10-21 Exporting a table



- ii. Enter an output file location and a message file location and click **OK**. This is shown in Figure 10-22.

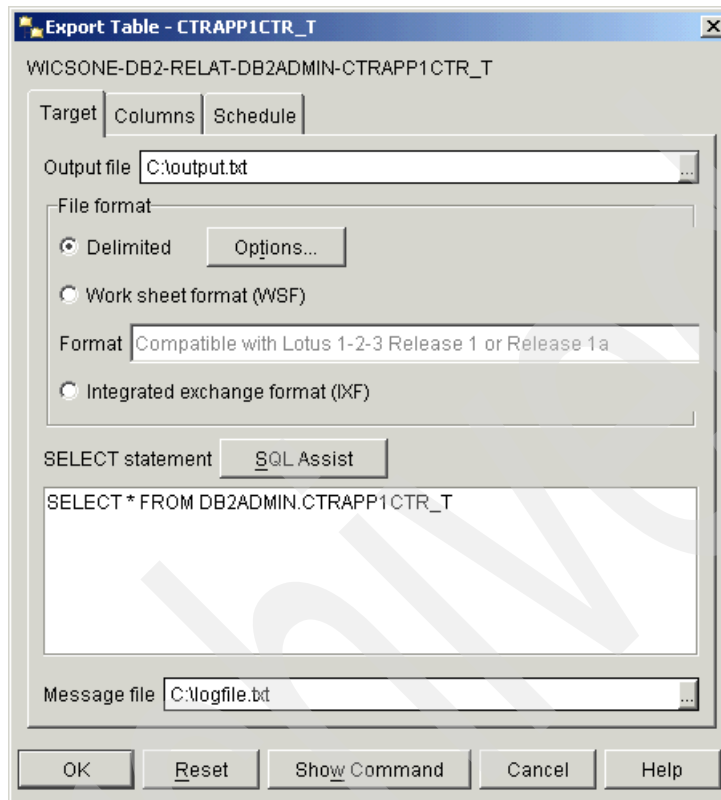


Figure 10-22 Completing the table export

- iii. In the Control Center, right-click the newly created relationship table and choose **Import**.

- iv. Enter the location of the saved file as the import file and click **OK** as shown in Figure 10-23.

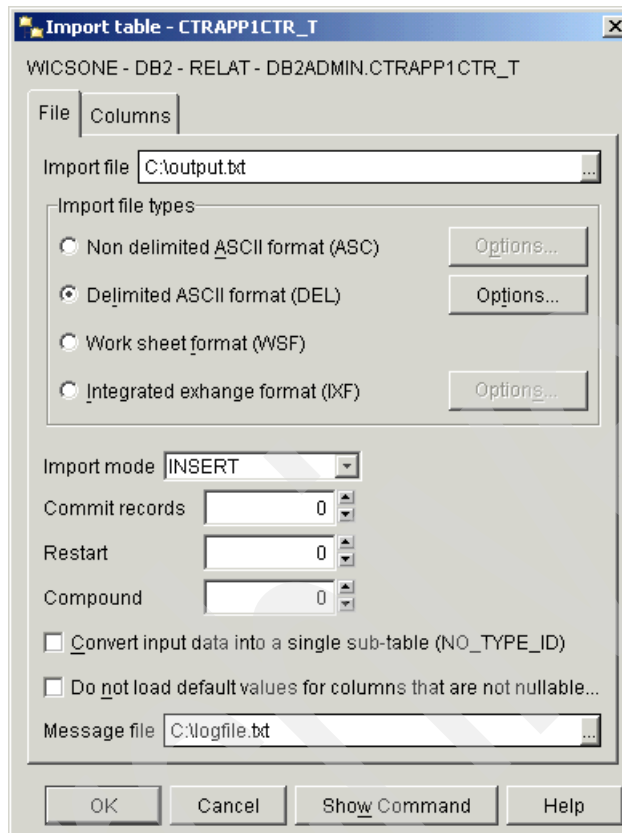


Figure 10-23 Importing the data

- b. Additional steps are needed to set the sequence number for each relationship. Here are the steps required to set the sequence number when the relationships are stored in DB2.
- Find the current highest ID for any relationship by examining the INSTANCEID column for any participant table in the relationship.  

```
SELECT MAX(INSTANCEID) FROM DB2ADMIN.CUSTAPP1CUST_T
```

In this example, the query has returned the value 41 as the highest INSTANCEID.
  - Find the sequence variable name. The following command generates a SQL script that provides information about the relationship table. This

information can also be obtained in the control center by right-clicking a database table and choosing **Generate DDL**.

```
db2look -dRELAT -tCUSTAPP1CUST_T -a -e -c -oOUTPUT.SQL
```

Examine the OUTPUT.SQL file to obtain the sequence variable name. Search for the text CREATE SEQUENCE.

```
CREATE SEQUENCE "DB2ADMIN"."CUSTOMER_1_0_0" AS INTEGER  
MINVALUE 1 MAXVALUE 2147483647 START WITH 1 INCREMENT BY 1  
CACHE 20 NO CYCLE NO ORDER;
```

In the previous example, the sequence variable name is Customer\_1\_0\_0. The sequence name is derived from the name of the relationship. In this case the relationship name is Customer.

- iii. Set the sequence variable to be at least one larger than the current number. In this example, the current highest INSTANCEID is 41 so the sequence is set to 42 with this command:

```
alter sequence Customer_1_0_0 restart with 42
```

- c. Verify that the relationship has been successfully moved to the new database by examining the relationship using Relationship Manager.

### 10.2.3 Post-migration steps

This section describes the steps to be taken after migration is complete and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the password previously assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.

2. Right-click the server name in the Interchange component management view and select **Users and Roles**. This is shown in Figure 10-24.

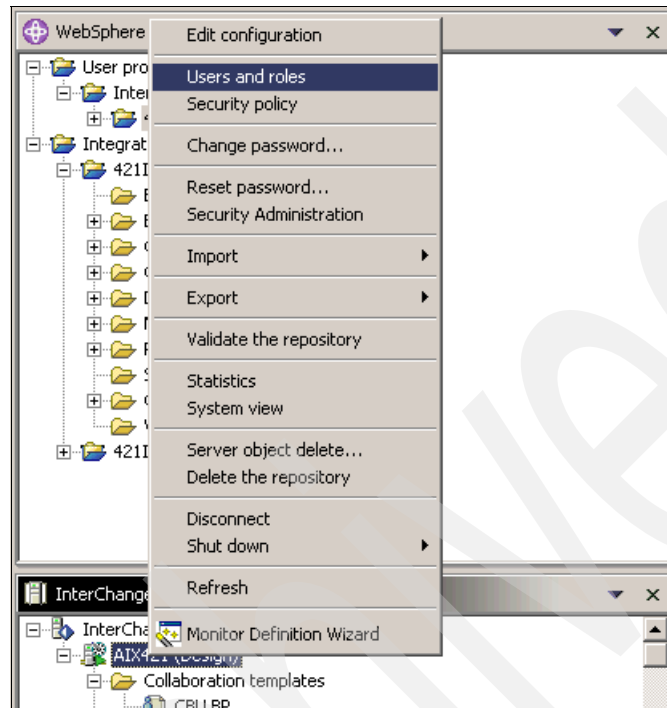


Figure 10-24 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User**. This is shown in Figure 10-25.

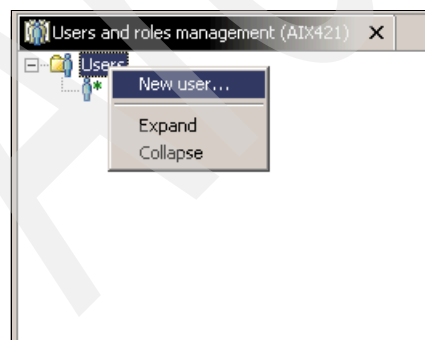


Figure 10-25 Selecting New user

4. The Create User window opens as shown in Figure 10-26. Enter the name admin and the previous password, plus additional information if needed. Click **OK** when finished.

A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' with 'admin' entered, 'Password' with '\*\*\*\*', 'Confirm password' with '\*\*\*\*', 'Distinguished name', 'Common name', and 'Surname'. At the bottom right are 'OK' and 'Cancel' buttons.

Create user

User name  
admin

Password  
\*\*\*\*

Confirm password  
\*\*\*\*

Distinguished name

Common name

Surname

OK Cancel

Figure 10-26 Creating a user

5. The new user is added to the Users and roles management window.



Figure 10-27 The newly created admin user

6. Right-click the newly created admin user and select **Add roles**. This is shown in Figure 10-28.

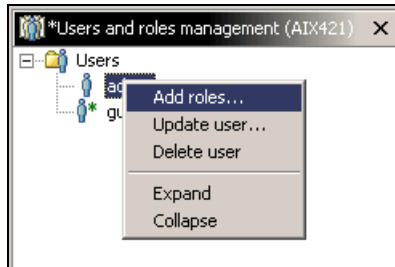


Figure 10-28 Add roles

7. The Role list window opens. Select **administrator** and click **OK**.

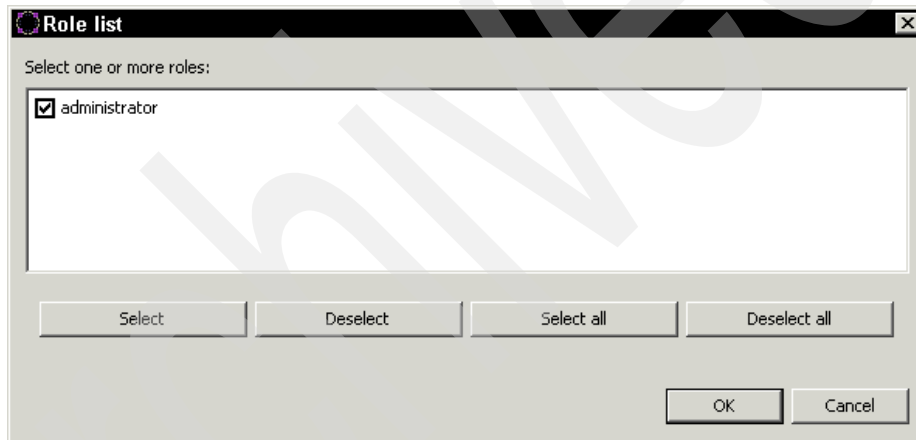


Figure 10-29 Role list

8. Close the Users and roles management window.



Figure 10-30 User admin with administrator role

9. Right-click the server again in the Components management view and choose **Edit Configuration**. This opens an editor for configuring the server.

10. Click the **Security-RBAC** tab (Figure 10-31) and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Enter admin for **Server start user**.
  - c. Set **Server start password** to the password for the admin user.
  - d. Check all other settings in the tab and modify if needed.
  - e. Save the configuration to the server by clicking the save icon or selecting **File** → **Save**.

Figure 10-31 Server Configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** Adapter agents default to the guest account if the RBAC is activated for the WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions on how to install adapters, refer to the individual adapter guides.

## Server Access Interface

If any Access Clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the upgraded InterChange Server system from development to production, it is recommended that you perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information on testing a migrated system.

Table 10-3 shows the scenarios that were tested in this migration, and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 10-3 Test scenario results

Test scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated
Web-based System Monitor	Successfully migrated



Check the support site to see whether a fix for the problem mentioned in this section is available.

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is re-started in Production mode.

## 10.3 In-place database migration for V4.2.2 to V4.3. on AIX

This section describes an in-place database migration scenario and the migration of WebSphere InterChange Server Version 4.2.2 to WebSphere InterChange Server Version 4.3.0. The complete repository from WebSphere InterChange Server 4.2.2 is reused in this scenario. This includes component definitions, failed flows and relationships.

For the full description of a in-place database migration refer to Chapter 4, “General planning” on page 53.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 10.3.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system. The steps for performing these tasks are described in this section.

**Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

## Put the environment in a quiescent state.

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server in a quiescent state, perform the following steps:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the System Manager.
2. Resolve transactional status of transactional collaborations using the Flow Manager. Transactional collaborations that are in transactional status can be migrated, however it is recommended that they are resolved in this pre-migration step.
3. Resubmit failed events or discard the events using Flow Manager. Failed flows can be migrated, however it is recommended that they are resolved in this pre-migration step.
4. Once all events are processed by the system, stop all collaborations. To check if all events are processed, check the collaboration statistics in System Manager. Make sure the in-process and pending event counter is 0.

## Backing up and shutting down components

This section describes the steps needed to shut down and take a backup. It is a recommended practice to take a backup which allows the recovery of any data that might be inadvertently overwritten during the installation of ICS 4.3.0.

Additional information about backing up the system can be found in the *System Administration Guide*. There are two types of data in the ICS system: static data and dynamic data.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These are the steps to perform the backup:

1. Back up the current InterChange Server repository using the **repos\_copy** utility.

For example, suppose the Interchange Server instance is AIX422 and has the default login of admin and the password of null. The following **repos\_copy** command creates a backup of the repository objects in a file called Repository422.jar. This file is created in the current directory where the **repos\_copy** command is run.

```
repos_copy -sAIX422 -uadmin -pnull -oRepository422.jar
```

A **repos\_copy** is not used during an in place migration, however, it is still a recommended practice in order to back up the current definitions of the WebSphere InterChange Server repository.

2. To export existing user projects, connect System Manager running on a connected Windows machine to the existing ICS instance and take the following steps within System Manager:
  - a. Right-click **User Projects** and select **Export Solution**.
  - b. Select all user projects that are to be want to export and enter a destination path.

**Restriction:** At the time of this writing, a product defect causes the Export Solution utility to not work properly. This defect has been reported and assigned tracking number CR60456. Check the support site below to see if a fix for this problem number is available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and choose **Export to Repository File**. The *Implementation Guide for WebSphere InterChange Server* contains further information about this process. The User projects are needed to be recreated from the integration component library after the migration.

3. Shutdown the WebSphere InterChange Server. Type **ics\_manager -stopgraceful** at the AIX command line.

4. Arrange for the database administrator to back up the database. This must be a complete backup, including schema information, and stored procedures.
5. Stop the Persistent Name Server
6. Arrange for the WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.

## 10.3.2 Migration steps

This section describes the main steps required for the migration including:

- ▶ Upgrading related software to the required level
- ▶ Installing and configuring WebSphere InterChange Server 4.3.0
- ▶ Completing the upgrades of WebSphere InterChange Server components

### Preparing the database system

The database used for the repository in WebSphere InterChange Server 4.2.2.4 may not meet the minimum database level required by Version 4.3.0, so it must be upgraded to the supported level. After upgrading the database, ensure that the database has the proper configuration for WebSphere InterChange Server 4.3.0.

- ▶ Check the *IBM WebSphere InterChange Server Installation Guide for UNIX* for the proper database settings for V4.3. For example, when using DB2, a change in the database settings is that the `applsapksz` parameter must be set to at least 4096.

Read or download the *IBM WebSphere InterChange Server System Installation Guide for UNIX* at:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

In the left pane, expand **Interchange Server and Toolset** → **Installation** → **PDFs** and click **UNIX** in the right-side pane.

- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 fix pack 7. If failed flows are to be migrated that involve MQ Delivery Transport, the queue messages must be preserved when upgrading WebSphere MQ. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the WebSphere MQ Messaging menu, select **latest multiplatform books**.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99.

It is recommended that WebSphere InterChange Server Version 4.3.0 is installed with a different user ID and to a different directory than the previous version. The previous installation is not modified during the install process.

During the install process, use the same WebSphere InterChange Server name as in the previous version. As a reminder, the WebSphere InterChange Server Name must not contain dots.

The WebSphere InterChange Server name can be changed after the installation process. Edit the file `$CROSSWORLDS/bin/ics_manager` and the file `$ADAPTER_RUNTIME/bin/connector_manager`. Change the parameter `SERVERNAME` to the desired WebSphere InterChange Server name.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following pictures and examples explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
$CROSSWORLDS/bin/ICConfig.sh
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all the configuration settings have to be entered in the file `Options.txt` which is delivered with the installer. For detailed information, refer to the *Installation Guide for UNIX* (page 69).

The following figures show the configuration values used by the redbook team during migration:

1. Figure 10-33 on page 373 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

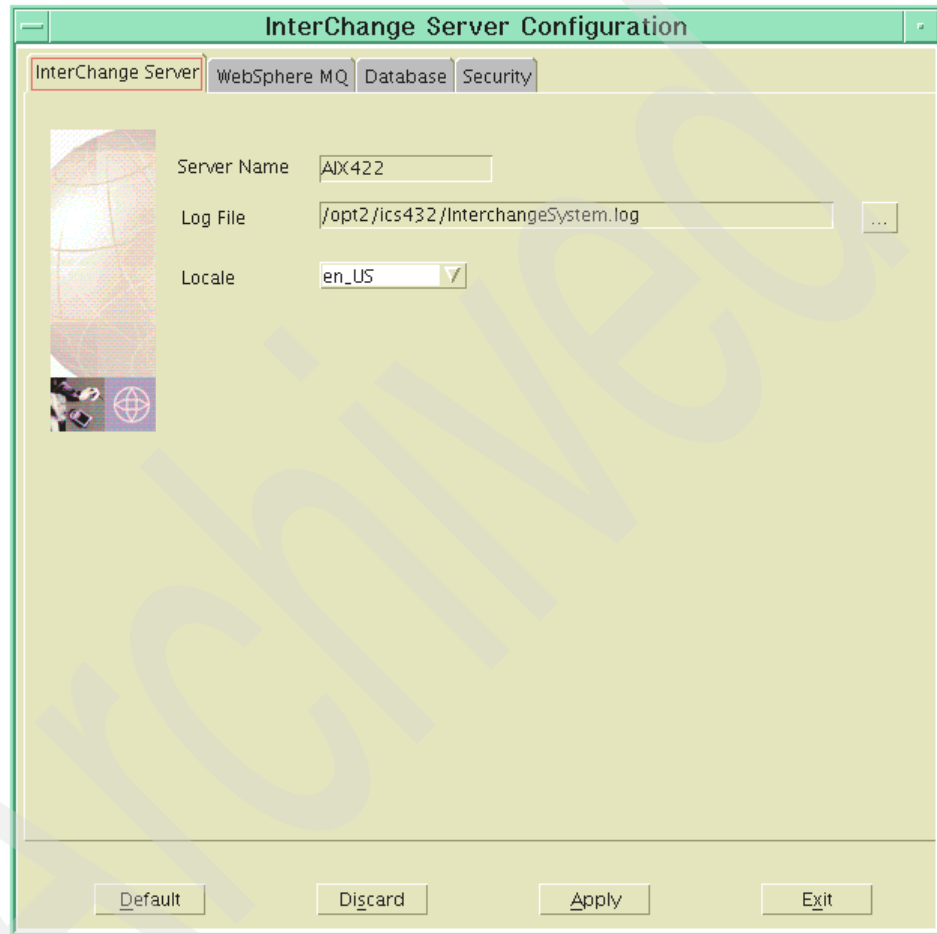
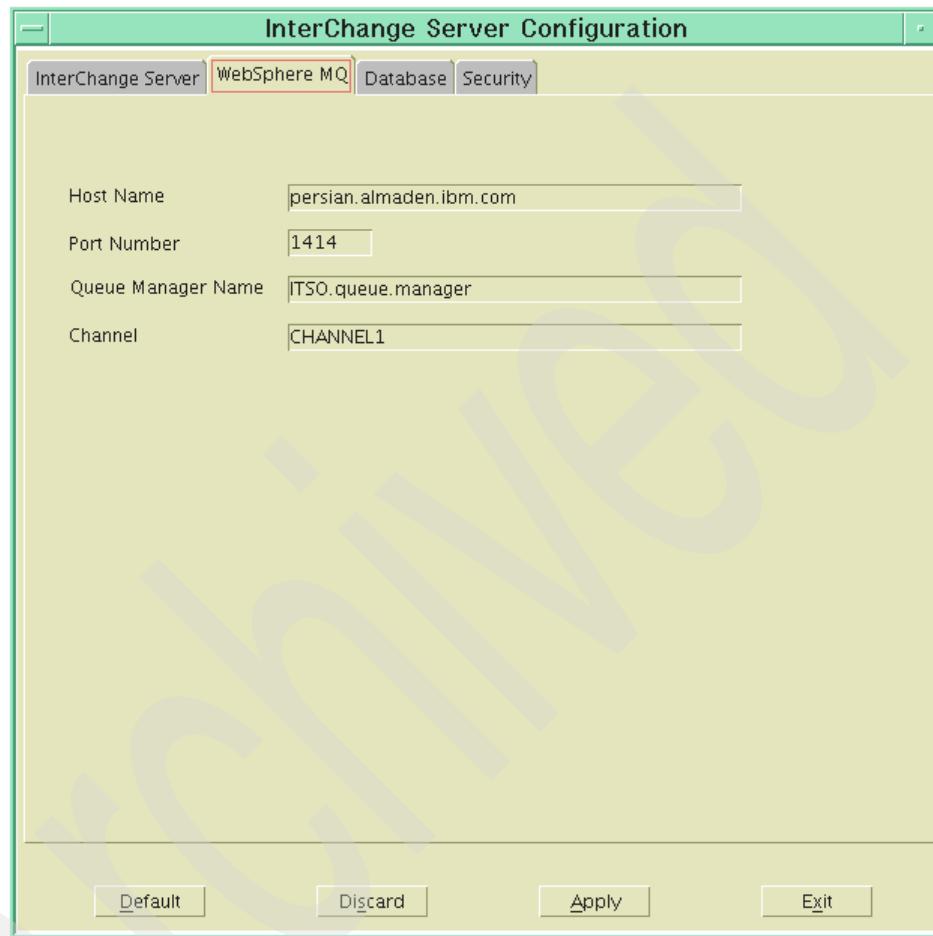


Figure 10-32 InterChange Server Configuration: InterChange Server tab

- Figure 10-33 shows the WebSphere MQ configuration tab. Make sure the settings for the WebSphere InterChange Server 4.2.2.4 are entered here.



The image shows a dialog box titled "InterChange Server Configuration". It has four tabs: "InterChange Server", "WebSphere MQ", "Database", and "Security". The "WebSphere MQ" tab is selected and highlighted with a red border. The dialog contains four text input fields with the following labels and values:

Field Label	Value
Host Name	persian.almaden.ibm.com
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

At the bottom of the dialog, there are four buttons: "Default", "Discard", "Apply", and "Exit".

Figure 10-33 InterChange Server cConfiguration: WebSphere MQ tab

3. Figure 10-34 shows the Database configuration. Make sure to enter the connection information from WebSphere InterChange Server 4.2.2.4 and apply the following settings:
- Max Connections: unlimited, if possible. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' window with the 'Database' tab selected. The window is divided into several sections: Database Connectivity, Event Management, Transactions, Repository, and Flow Monitoring. Each section contains fields for Host Name, Database, Login, Password, Max Connections, and Port Number. The 'Database Connectivity' section has a 'Database Driver' dropdown set to 'DB2', 'Max Connections' set to 'Unlimited', 'Max Pools' set to '50', and 'Idle Timeout' set to '2'. The other sections have 'Host Name' and 'Database' fields, and 'Login' and 'Password' fields. The 'Max Connections' field is set to 'Unlimited' in all sections. The 'Port Number' field is empty in all sections. There are 'Change' buttons next to the 'Password' fields. At the bottom of the window are buttons for 'Default', 'Discard', 'Apply', and 'Exit'.

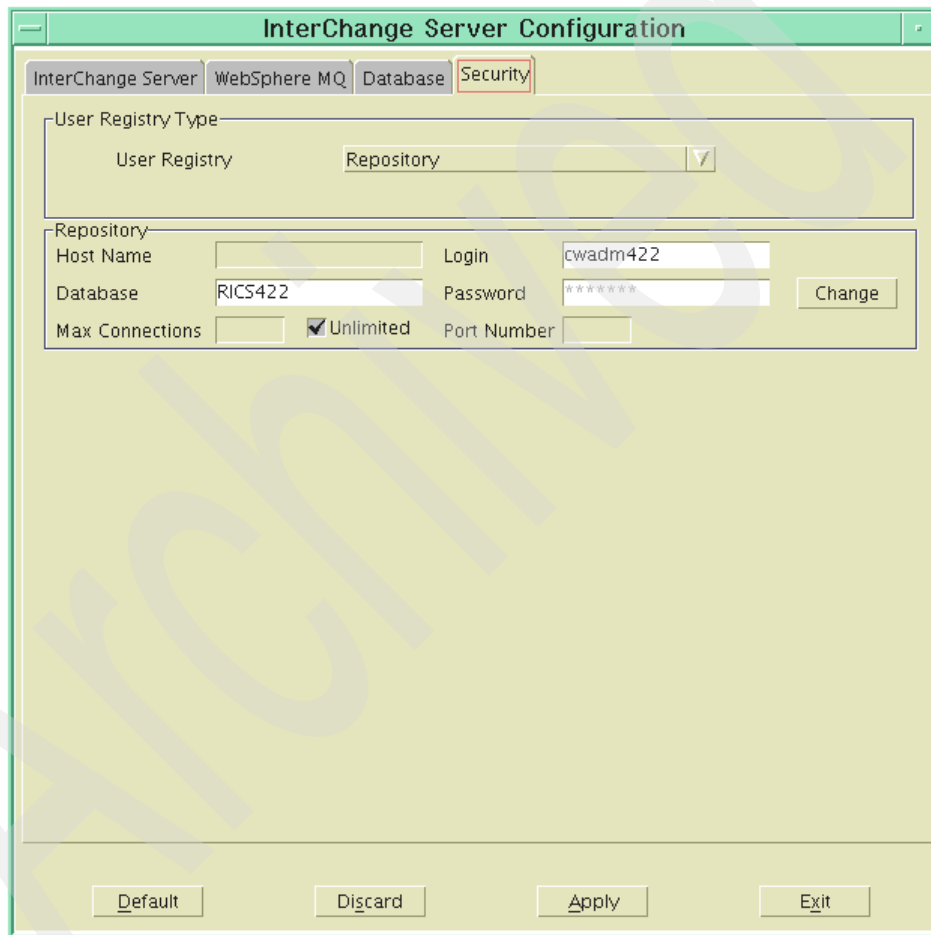
Section	Host Name	Database	Login	Password	Max Connections	Port Number
Database Connectivity					Unlimited	
Event Management		RICS422	cwadm422	*****	Unlimited	
Transactions		RICS422	cwadm422	*****	Unlimited	
Repository		RICS422	cwadm422	*****	Unlimited	
Flow Monitoring		RICS422	cwadm422	*****	Unlimited	

Figure 10-34 InterChange Server configuration: Database tab



4. The fourth tab is for security configuration (Figure 10-35). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'Security' tab. The dialog has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Security' tab is active and highlighted with a red border. Inside the 'Security' tab, there is a section for 'User Registry Type' with a dropdown menu set to 'Repository'. Below this, there is a 'Repository' section with several fields: 'Host Name' (empty), 'Login' (containing 'cwadm422'), 'Database' (containing 'RICS422'), 'Password' (containing '\*\*\*\*\*'), 'Max Connections' (empty), a checked 'Unlimited' checkbox, and 'Port Number' (empty). A 'Change' button is located to the right of the 'Password' field. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

InterChange Server Configuration			
InterChange Server WebSphere MQ Database <b>Security</b>			
User Registry Type			
User Registry		Repository	
Repository			
Host Name		Login	cwadm422
Database	RICS422	Password	*****
Max Connections		<input checked="" type="checkbox"/> Unlimited	Port Number
		Change	
Default Discard Apply Exit			

Figure 10-35 InterChange Server Configuration: Security tab

## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of WebSphere InterChange Server V4.3.0, additional steps are required to prepare the environment:

1. Copy DLMS and collaboration directories from WebSphere InterChange Server 4.2.2.4 to the new installation directory for Version 4.3.0. Be sure to include all subdirectories.

Example from this migration scenario:

```
cp -R /opt2/ics422/DLMS /opt2/ics432/  
cp -R /opt2/ics422/collaborations /opt2/ics432
```

2. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.

For example, to disable DEADLOCK\_DETECTOR\_CHECK for collaborations, edit the config file code shown in Example 10-3 using a suitable editor for XML.

### Example 10-3 Disable DEADLOCK\_DETECTOR\_CHECK

---

```
<tns:property>  
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>  
<tns:value xml:space="preserve">true</tns:value>  
<tns:isEncrypted>false</tns:isEncrypted>  
<tns:updateMethod>system restart</tns:updateMethod>  
<tns:location>  
<tns:reposController>false</tns:reposController>  
<tns:reposAgent>false</tns:reposAgent>  
<tns:localConfig>true</tns:localConfig>  
</tns:location>  
</tns:property>
```

---

Change true to false in the line starting with <tns:value.

3. Include any external code and .jar files in the CLASSPATH environment variable. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external .jar files are included in %CROSSWORLDS/bin/CWSharedEnv.sh. The external classes are first added to a new variable EXT\_LIB. Then the EXT\_LIB is appended to the CWCLASSES variable. See the following example from CWSharedEnv.sh:

```
# external classes added here  
EXT_LIB=${CROSSWORLDS}/lib/Sample.jar  
  
CWCLASSES=${WLICENSE}:${CORECW}:${DATAHANDLERS}:${DATADIRECT}:${DATAMANA  
GER}:${ITLM}:${JAVAMAIL}:${WEBSERVICES_JARS}:${EXT_LIB}  
export CWCLASSES
```

4. Modify and verify the environment variables as documented in the Installation Guide for UNIX.

For example, make sure the path to the Java 1.4.2 compiler precedes the Java 1.3.1 compiler if the 1.3.1 compiler is still in the path. The following example represents part of the \$HOME/.profile from one of our test servers:

```
#Start DB Environment
. /home/db2inst1/sqllib/db2profile
export EXTSHM=ON
# End DB Environment

# WBI environment
export PATH=/usr/java14/bin:$PATH
. /opt2/ics43/bin/CWSharedEnv.sh
#END WBI environment
```

Make sure the latest settings are in effect. To confirm that the right Java Compiler is activated, type **which javac**.

The following output is returned for AIX, which is the installation path for JDK 1.4.2:

```
/usr/java14/bin/javac
```

5. Install the WebSphere Business Integration Toolset on a supported Windows client. If the tools are to be installed on a Windows client that currently hosts an older version of tools, the old tools must be uninstalled prior to the installation of the new tools. You should make a backup of the Windows client before uninstalling and installing the new tools.

During installation, a prompt for the location of the Object Request Broker is displayed. Enter the host name or IP address of the machine hosting the Persistent Name Server that WebSphere InterChange Server is using. Also enter the port number if it is different from the default, 14500. To change the settings after installation, edit %CROSSWORLDS%\bin\CWSharedEnv.bat:

```
set ORB_PORT=14500
set ORB_HOST=persian.almaden.ibm.com
```

## Starting the new server

When WebSphere InterChange Server is started for the first time after installation, the WebSphere InterChange Server repository tables are upgraded or created. This section describes the steps to prepare and start the server for the first time:

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database

2. Start the Persistent Name Server. In a new console, run this command:

```
$CROSSWORLDS/bin/PersistentNameServer.sh
```

3. Start the WebSphere InterChange Server in design mode:

```
$CROSSWORLDS/bin/ics_manager -start -design
```

The WebSphere InterChange Server upgrades existing tables and creates new tables for new features.

WebSphere InterChange Server logs messages to the InterChangeSystem.log file located in the product directory. The message Interchange Server is Ready indicates that WebSphere InterChange Server has started successfully.

### Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, import the component libraries using exported integration component library files:

- Create a new integration component library.
- Right-click the newly created library and select **Import Repository from File**. Choose the repository file that was created in the backup procedure. Repeat these steps if there are multiple integration component libraries to import. Read more about this process in the *Implementation Guide for WebSphere InterChange Server*.

2. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH for the Toolset. Add the location of the custom .jar files to this list.

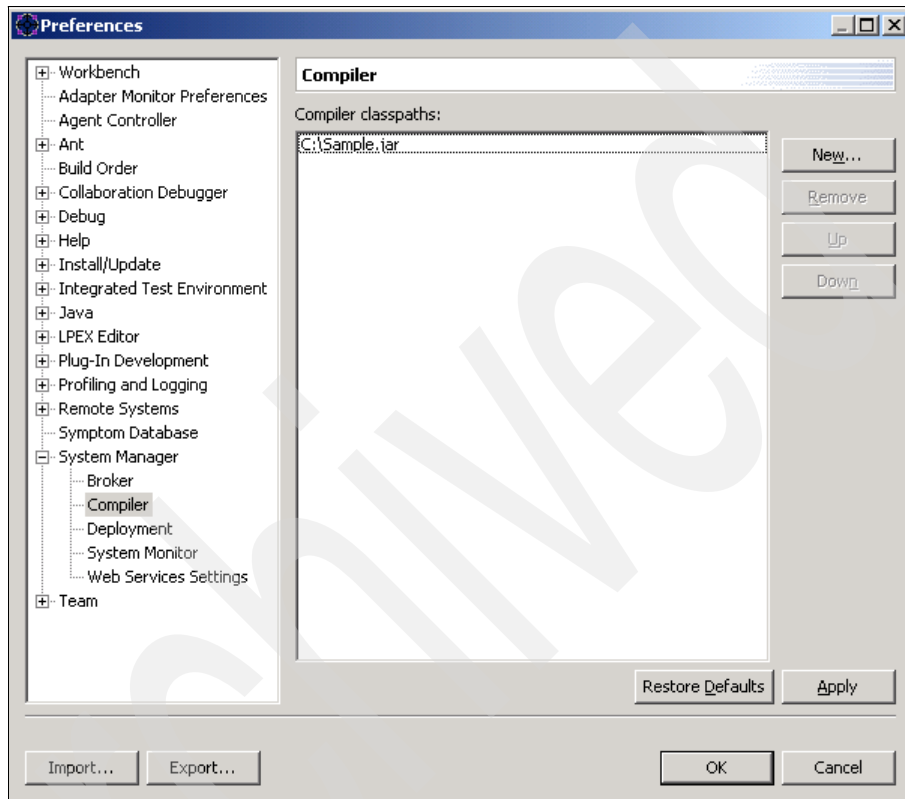


Figure 10-36 Setting the classpath for the Toolset

3. Set the compliance of the compiler to Java 1.4. Select **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag, change the Compiler compliance level to **1.4**. A dialog box is displayed indicating that the compiler settings have changed. Choose **No** to not rebuild files at this time.

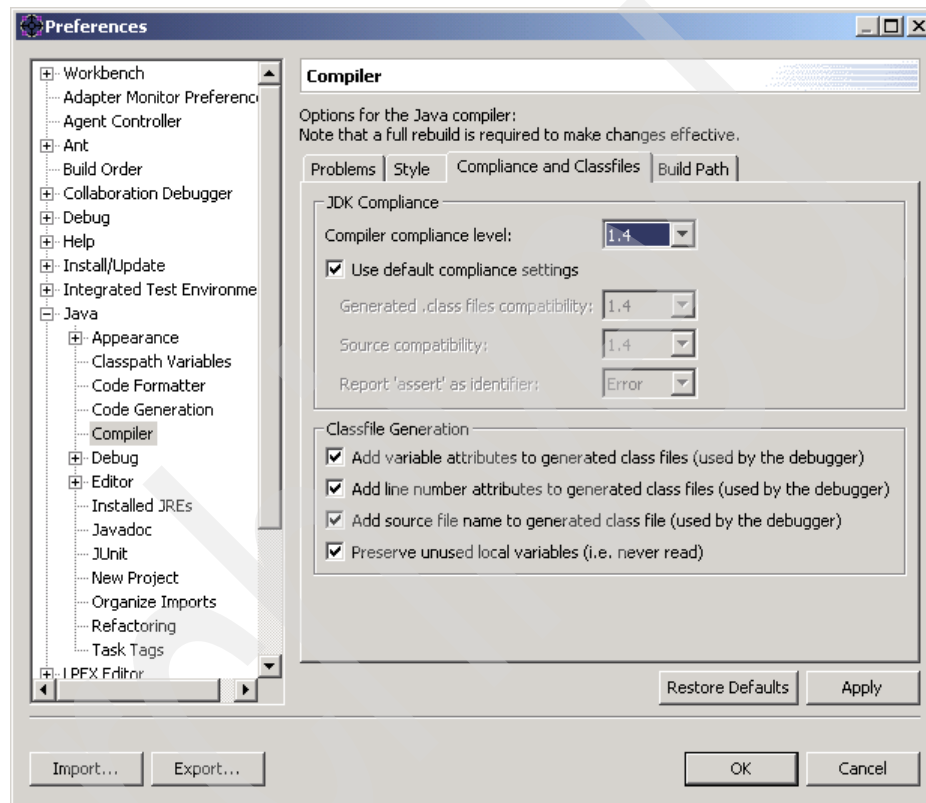


Figure 10-37 Setting the Compiler compliance level

4. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the usage of the import statement. In Java 1.4, the import statement must be followed by a packaged class, whereas Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level has been correctly set in previous step.
5. Compile all the maps and templates in the integration component libraries and examine the console for any error messages.

6. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions examine the Port and Triggering Events tab. For the **BO Type** that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to <None> for all verbs.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
7. Resolve any collaborations that are transactional status, because transactional status is lost after redeployment of the collaboration. This can be done from the Flow Manager. If the collaboration cannot be brought to an active state to resolve the transactional status, deploy only the updated components required to activate the integration, then resolve the transactional status.

**Note:** To make use of all administrative features for maps and collaborations, the maps and templates need to be deleted and redeployed. Typically, each component must be placed in an inactive state before it can be overwritten. To simplify the process and to avoid any unresolved dependencies, the whole repository is deleted in step 9 and redeployed in step 10. Additionally, recompilation ensures that the code is optimized and Java 1.4 compliant.

However, it is not required to redeploy the maps and templates, because the Java Runtime Environment is backwards compatible and poses no problems executing map and template class files compiled in WebSphere InterChange Server 4.2.2.4.

If the decision is made to reuse the previous maps and collaborations, only deploy the components which have been updated.

8. In the System Manager, delete the repository by right-clicking the WebSphere InterChange Server name in the Interchange Server component management view and select **Delete the repository**.

**Note:** If Import Solution was not used, create a new user project and populate the user project with the components from the integration component library. Refer to the *Implementation Guide for WebSphere InterChange Server* for details.

9. Deploy the user project:
  - a. Right-click the user project and select **Deploy user project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This recreates the stored procedures and does not overwrite existing data.

10. Restart the WebSphere InterChange Server if prompted.

### 10.3.3 Post-migration steps

This section describes the steps that need to be taken after migration and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the previous password assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.



2. Right-click the server name in the Interchange component management view and select **Users and Roles**. This is shown in Figure 10-38.

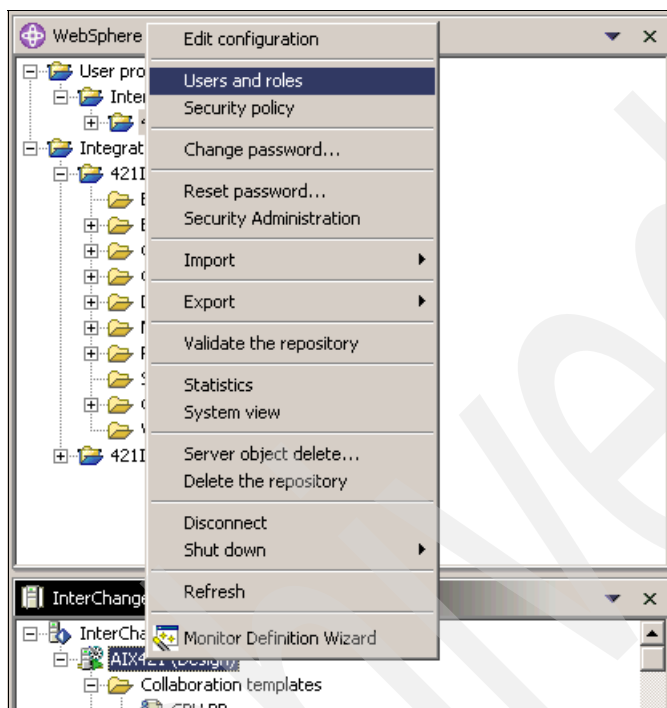


Figure 10-38 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click Users icon and select **New User**. This is shown in Figure 10-39.

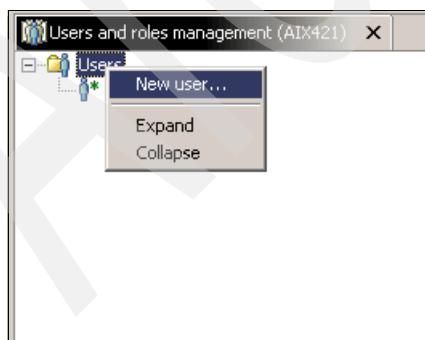


Figure 10-39 Selecting New user

4. The window Create User comes up. This is shown in Figure 10-40. Enter the name admin and select the previous password plus additional information if needed. Press **OK** when finished.

A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' with 'admin' entered, 'Password' with '\*\*\*\*', 'Confirm password' with '\*\*\*\*', 'Distinguished name' (empty), 'Common name' (empty), and 'Surname' (empty). At the bottom right are 'OK' and 'Cancel' buttons.

Create user

User name  
admin

Password  
\*\*\*\*

Confirm password  
\*\*\*\*

Distinguished name

Common name

Surname

OK Cancel

Figure 10-40 Creating a user

5. The new user is added to the User and Roles management window. This is shown in Figure 10-41.

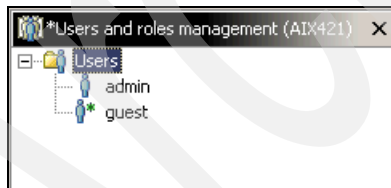


Figure 10-41 The newly created admin user

6. Right-click the newly created admin user and select **Add roles...**This is shown in Figure 10-42.

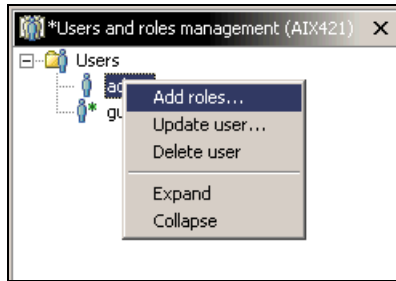


Figure 10-42 Add roles

7. The Role list window is displayed. This is shown in Figure 10-43.

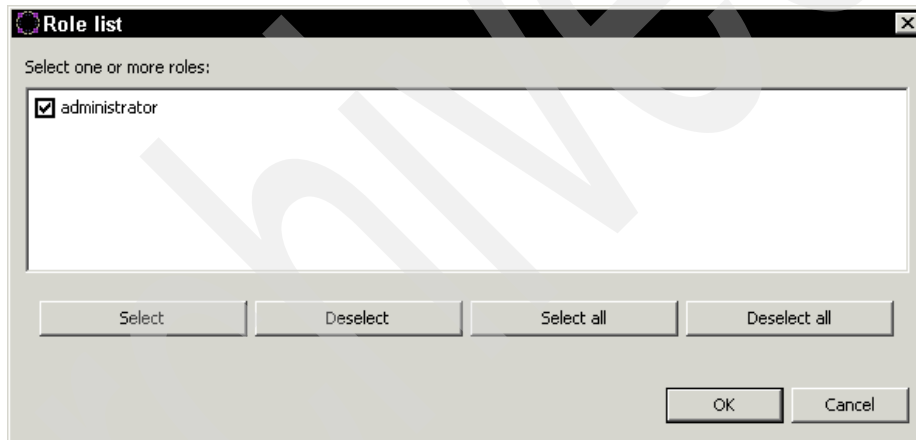


Figure 10-43 Role list

8. Select **administrator** and click **OK**.
9. Close the Users and roles management window (Figure 10-44)



Figure 10-44 User admin with administrator role

10. Right-click the server again in the Interchange Components management view and choose **Edit Configuration**. An editor to configure the server is displayed.
11. Click the **Security - RBAC** tab as shown in Figure 10-45.

Figure 10-45 Server Configuration - Security-RBAC tab

12. In the tab, apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Set **Server start user** to admin.
  - c. Set **Server start password** to the password for the admin user.
  - d. Check all other settings in the tab and modify if needed
  - e. Save the configuration to the server by pressing the save icon, or choosing **File → Save**.
13. Reboot WebSphere InterChange Server to activate security.

**Note:** The adapter agent defaults to the guest account if the RBAC is activated for WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions on how to install adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they need to be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end and performance testing. Before moving the upgraded InterChange Server system from development to production it is recommended to perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 10-4 shows the scenarios that were tested in this migration, and the results found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 10-4 Test scenarios results

Test scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated

Test scenario	Results
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated

A failed flow is considered to be successfully migrated if the event can be resubmitted and completed after migration. Table 10-5 shows the results of the migration of Failed Flows.

Table 10-5 Migrated failed flows results

Failed flows tested	Results
Failed flow for IDL	Failed because of hierarchical Business Object structure
Failed flow for MQ	Failed because of hierarchical Business Object structure
Failed flow for JMS	Failed because of hierarchical Business Object structure
Transactional collaboration in transactional status (compensation call failed)	Successfully migrated with flat Business Object structure
Transactional collaboration failed flow (compensation call successful)	Successfully migrated with flat Business Object structure
LLBP waiting asynchronous inbound call	Failed because of hierarchical Business Object structure
LLBP failed flow	Failed because of hierarchical Business Object structure
Collaboration group failed at Destination 1	Successfully migrated with flat Business Object structure
Collaboration group failed at Destination 2	Successfully migrated with flat Business Object structure

It can be seen from Table 10-5 that the majority of the Failed Flows were not able to be successfully migrated. In response to this the Change Request CR60367 was raised. The cause of this problem is that in WebSphere InterChange Server 4.2.2.4 a Child BO Verb with value *null* is set to *CxIgnore* which is not expected by WebSphere InterChange Server 4.3.0. and is therefore rejected when resubmitted. WebSphere InterChange Server 4.3.0 is changed to handle the

*CxIgnore* case in a future version. It is recommended that Failed Flows are handled before migration until this problem has been resolved.

Check the support site to see if a fix for the problems mentioned in this section are available.

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that the WebSphere InterChange Server is re-started in Production mode.

### Uninstall (optional)

This section describes the steps to uninstall the previous version of WebSphere InterChange Server and the related software. The following steps list the order of uninstalling relevant software

1. Uninstall JDK as the root user using `smi t`, only if it is not required by other products. For this release, the JDK is typically Java 2 SDK v.1.3.1. Contact the system administrator for further assistance.
2. Uninstall WICS using the uninstaller located in the product directory once it is certain that the previous version is no longer needed. After the uninstaller has completed, the product directory can be deleted.

## 10.4 Without in-place database migration of V4.2.2 to V4.3 on AIX

This section describes a without in-place database migration scenario and the migration of Version 4.2.2.4 to Version 4.3.0. The 4.3.0 WebSphere InterChange Server is started with a new and empty repository. Repository content (component definitions) are exported from the original WebSphere InterChange Server 4.2.2 version and deployed in WebSphere InterChange Server Version 4.3.0 environment.

**Tip for advanced users:** Appendix A, “WebSphere InterChange Server migration checklist” on page 419 provides a checklist of the migration steps.

### 10.4.1 Pre-migration steps

The environment must be put into a quiescent state before the migration is attempted. It is also necessary to take a backup of the system. The steps for performing these tasks are described in this section.

**Important:**

- ▶ Before starting the migration, ensure that the current WebSphere InterChange Server is running at least at V4.1.1.9.
- ▶ Ensure that all components follow the Naming Conventions. Refer to the appropriate V4.3 Development Guide for each component. For example, business object names and attributes can use only the US English character set (umlauts cannot be migrated).
- ▶ WebSphere InterChange Server names that contain dots (.) are no longer supported. See Chapter 5, “Technical impact of major design changes” on page 67. The Installation Guide for UNIX has more information about other UNIX platforms.

**Put the environment in a quiescent state**

The WebSphere InterChange Server is in a quiescent state when all of the following conditions exist:

- ▶ All working queues are emptied.
- ▶ All collaborations are paused so that no new data can be written to the cross-reference tables.
- ▶ All data is consistent between the integrated applications.

To put the WebSphere InterChange Server into a quiescent state:

1. Stop all adapters from polling the event tables by setting the Poll Frequency property of the adapter to **No** using the System Manager.
2. Resolve transactional status of transactional collaborations using the Flow Manager. All status information is lost during the migration.
3. Resubmit failed events or discard the events using Flow Manager. Failed flows are lost after migration.
4. When all events have been processed by the system, stop all collaborations. To confirm that all events are processed, check the collaboration statistics in System Manager. Be sure that the in-process and pending event counter is 0.

**Backing up and shutting down components**

This section describes the steps for shutting down and taking a backup. It is a recommended practice to take a backup which allows the recovery of any data that might be inadvertently overwritten during the installation of WebSphere InterChange Server 4.3.0.



Additional information about backing up the system can be found in the *System Administration Guide*. There are two types of data in the WebSphere InterChange Server system: static data and dynamic data.

Static data includes:

- ▶ WebSphere InterChange Server repository (except for relationship tables)
- ▶ Custom collaborations components, such as Java class files and message files
- ▶ Custom connectors
- ▶ Map components, including: map design files, Java class files, and message files

Dynamic data includes:

- ▶ Relationship database tables
- ▶ Work-in-progress tables and transaction database tables
- ▶ WebSphere MQ message data
- ▶ Log files (as desired for historical information)

These steps perform the backup:

1. Back up the current WebSphere InterChange Server repository using the **repos\_copy** utility.

For example, suppose the InterChange Server instance is *AIX422* and has the default login of *admin* and the password of *null*. The following **repos\_copy** command creates a backup of the repository objects in a file called *Repository422.jar*. This file is created in the directory where **repos\_copy** command is running.

```
repos_copy -sAIX422 -uadmin -pnull -oRepository422.jar
```

To export existing user projects, connect System Manager running on a connected Windows machine to the existing WebSphere InterChange Server instance and take the following steps in System Manager:

- a. Right-click **User Projects** and select **Export Solution**.
- b. Select all user projects that are to be exported and enter a destination path.

**Restriction:** At the time of this writing, a product defect causes the Export Solution utility to not work properly. This defect has been reported and assigned tracking number CR60456. Check the support site to see whether a fix for this problem number is available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

To circumvent this problem, export the integration component libraries and user projects from System Manager. Right-click the integration component library and choose **Export to Repository File**. The *Implementation Guide for WebSphere InterChange Server* has more information about this process. The user projects must be recreated from the integration component library after the migration.

2. Shut down the WebSphere InterChange Server. Type **ics\_manager -stopgraceful** at the AIX command line.
3. Arrange with the database administrator to back up the database. This must be a complete backup, including schema information, and stored procedures.
4. Stop the Persistent Name Server.
5. Arrange with the WebSphere MQ administrator to back up the WebSphere MQ environment used by the WebSphere InterChange Server.
6. Clear WebSphere MQ queues of any outstanding messages if the same queue manager is reused for Version 4.3.0. Old messages in the queues may be considered as new events and may cause data inconsistencies. Refer to the WebSphere MQ documentation for more information.

## 10.4.2 Migration steps

This section describes the main steps required for the migration:

- ▶ Upgrading related software to the required level
- ▶ Installing and configuring WebSphere InterChange Server 4.3.0
- ▶ Completing the upgrades of WebSphere InterChange Server components

### Preparing the database system

Prepare an empty database for the 4.3.0 repository: Create a new database following the steps in the Installation Guide, and ensure that it has the proper configuration for WebSphere InterChange Server 4.3.0.

- ▶ Check the *IBM WebSphere InterChange Server Installation Guide for UNIX* for the proper database settings for V4.3. For example, when using DB2, a change in the database settings is that the `applheapsz` parameter must be set to at least 4096.

- ▶ Additionally, make sure that the C-compiler is installed and configured for stored procedures using DB2.

To confirm that the C-compiler is installed and configured, use the DB2 command line processor to connect to any database and run this command:

```
create procedure foo() language SQL begin end
```

A message is displayed indicating that the stored procedure has been created successfully, at which time it can be removed with the following command:

```
Drop procedure foo()
```

A message is displayed indicating that the stored procedure has been removed successfully. Refer to the DB2 documentation for more information.

- ▶ For DB2, it is recommended that you catalog the database system to route using TCP/IP as documented in the System Installation Guide for Unix.

Make sure that the changed database settings are effective. This can be achieved by restarting the database.

## Upgrade WebSphere MQ

WebSphere MQ must be upgraded to 5.3.0.2 Fix Pack 7. Refer to the WebSphere MQ documentation for detailed instructions at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

## Installation

At this point, install WebSphere InterChange Server 4.3. Refer to Chapter 6, “Installation of WebSphere InterChange Server 4.3” on page 99 for instructions.

It is recommended that WebSphere InterChange Server Version 4.3.0 be installed with a different user ID and to a different directory than the previous version. The previous installation is not modified during the install process.

During the install process, the WebSphere InterChange Server name may be changed. If so, the adapters and queues may have to be configured. Use the same WebSphere InterChange Server name if possible.

The WebSphere InterChange Server name can be changed after the installation process. Edit the file `$CROSSSWORLDS/bin/ics_manager` and the file `$ADAPTER_RUNTIME/bin/connector_manager`. Change the parameter `SERVERNAME` to the desired WebSphere InterChange Server name.

## Configure WebSphere InterChange Server

At the end of the installation with the graphical installer, the user is asked to configure the WebSphere InterChange Server. The following pictures and examples explain how to configure the WebSphere InterChange Server. The IBM WebSphere InterChange Server Configuration wizard can also be run after the installation has finished, by invoking:

```
$CROSSWORLDS/bin/ICConfig.sh
```

**Note:** If WebSphere InterChange Server is installed using the silent installer, all the configuration settings have to be entered in the file `Options.txt` which is delivered with the installer. For detailed information, refer to the *Installation Guide for UNIX* (page 69).

The following figures show the configuration values used by the redbook team during migration:

1. Figure 10-46 shows the InterChange Server tab, where the location and name of the log file and the locale for WebSphere InterChange Server are specified.

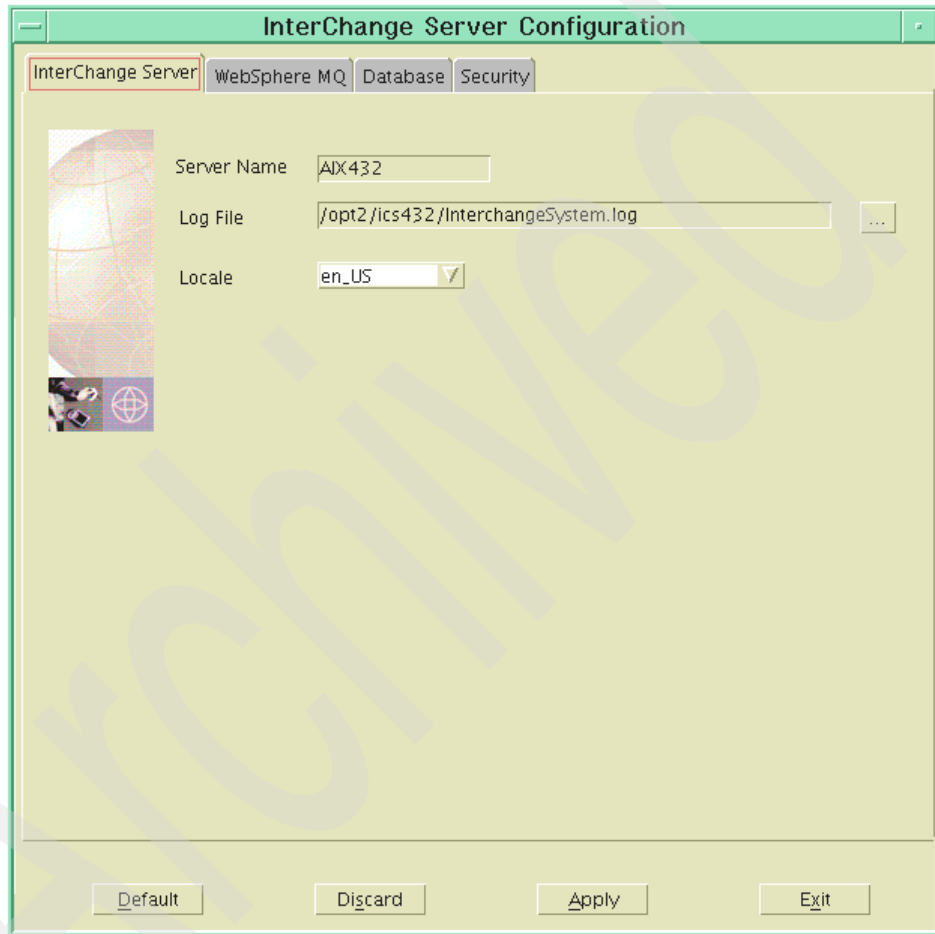
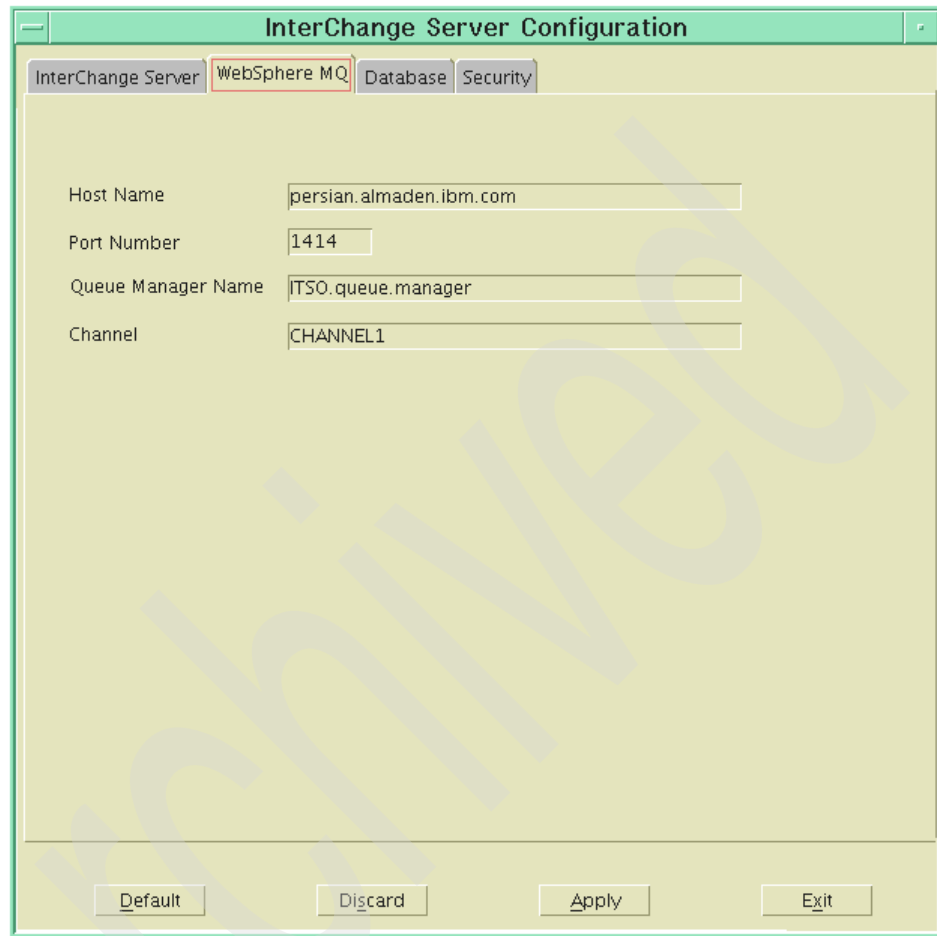


Figure 10-46 InterChange Server Configuration: InterChange Server tab

2. Figure 10-47 shows the WebSphere MQ configuration tab.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'WebSphere MQ' tab. The dialog has a title bar with the text 'InterChange Server Configuration'. Below the title bar, there are four tabs: 'InterChange Server', 'WebSphere MQ' (which is selected and highlighted with a red border), 'Database', and 'Security'. The main area of the dialog contains four labeled text input fields: 'Host Name' with the value 'persian.almaden.ibm.com', 'Port Number' with the value '1414', 'Queue Manager Name' with the value 'ITSO.queue.manager', and 'Channel' with the value 'CHANNEL1'. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Field	Value
Host Name	persian.almaden.ibm.com
Port Number	1414
Queue Manager Name	ITSO.queue.manager
Channel	CHANNEL1

Figure 10-47 InterChange Server Configuration: WebSphere MQ tab

3. The third tab is for database configuration (Example 10-48). Make sure to enter the connection information for the empty database created in the previous step and apply the following settings:
  - Max Connections: unlimited, if possible. The number of connections depends on the repository content, specifically the use of connection pools and relationships.
  - Max Pools: 50.

The screenshot shows the 'InterChange Server Configuration' window with the 'Database' tab selected. The window has four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Database' tab is active and contains several sections for configuration:

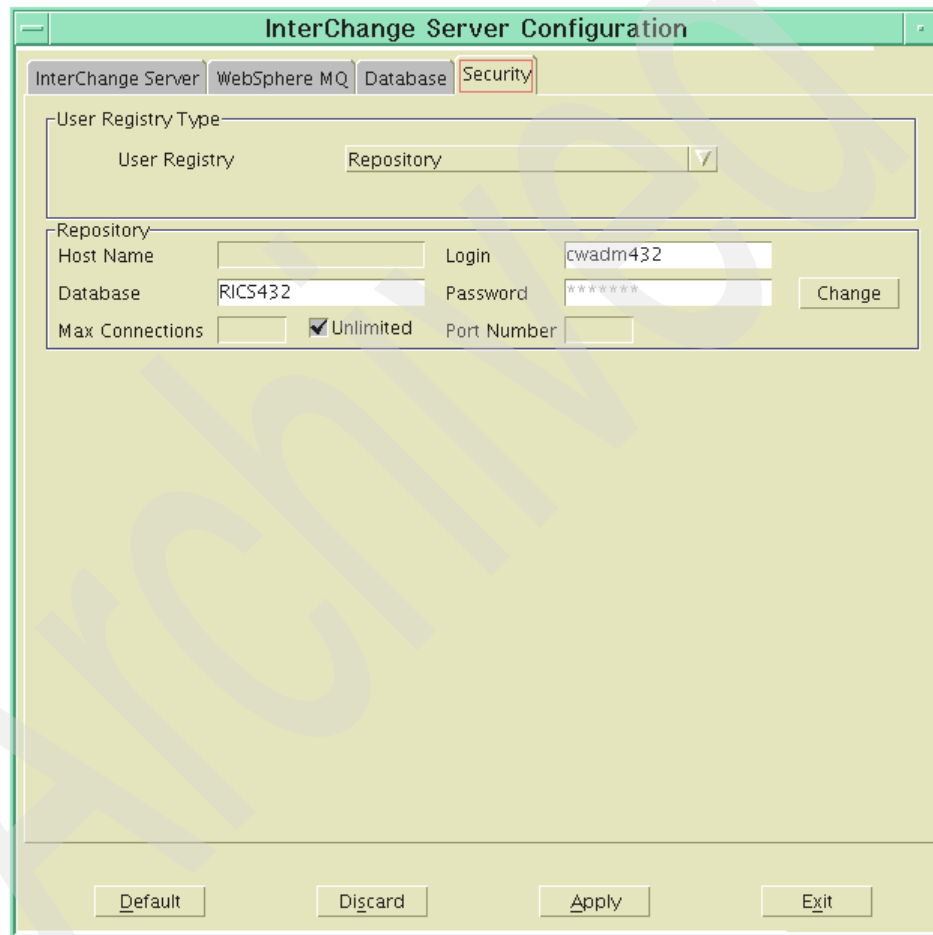
- Database Connectivity:** Includes a dropdown for 'Database Driver' set to 'DB2'. To the right, 'Max Connections' is set to 'Unlimited' (checked), 'Max Pools' is set to '50', and 'Idle Timeout' is set to '2'.
- Event Management:** Includes fields for 'Host Name', 'Database' (set to 'RICS432'), 'Login' (set to 'cwadm432'), and 'Password' (masked with asterisks). There is a 'Change' button. 'Max Connections' is set to 'Unlimited' (checked), and 'Port Number' is empty.
- Transactions:** Similar to Event Management, with 'Database' set to 'RICS432', 'Login' set to 'cwadm432', and 'Max Connections' set to 'Unlimited' (checked).
- Repository:** Similar to Event Management, with 'Database' set to 'RICS432', 'Login' set to 'cwadm432', and 'Max Connections' set to 'Unlimited' (checked).
- Flow Monitoring:** Includes fields for 'Host Name', 'Database' (set to 'RICS432'), 'Login' (set to 'cwadm432'), 'Schema Name', and 'Password' (masked). There is a 'Change' button. 'Max Connections' is set to 'Unlimited' (checked).

At the bottom of the window are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

Figure 10-48 InterChange Server Configuration: Database tab

4. The fourth tab is for security configuration (Figure 10-49). Introduced with Version 4.3 of WebSphere InterChange Server, this must be filled with valid information.

For User Registry, choose either **Repository** or **LDAP**. Do not choose LDAP unless an LDAP server is available and properly configured. If Repository is chosen, enter the appropriate database and connection information. You can use the same database and connection information from the Database tab for the Security database.



The image shows a screenshot of the 'InterChange Server Configuration' dialog box, specifically the 'Security' tab. The dialog has a title bar with the text 'InterChange Server Configuration'. Below the title bar, there are four tabs: 'InterChange Server', 'WebSphere MQ', 'Database', and 'Security'. The 'Security' tab is selected and highlighted. The main content area of the 'Security' tab is divided into two sections. The top section is labeled 'User Registry Type' and contains a dropdown menu with 'Repository' selected. The bottom section is labeled 'Repository' and contains several input fields: 'Host Name' (empty), 'Login' (containing 'cwadm432'), 'Database' (containing 'RICS432'), 'Password' (containing '\*\*\*\*\*'), 'Max Connections' (empty), 'Unlimited' (checked), and 'Port Number' (empty). There is a 'Change' button next to the 'Password' field. At the bottom of the dialog, there are four buttons: 'Default', 'Discard', 'Apply', and 'Exit'.

User Registry Type	
User Registry	Repository

Repository	
Host Name	
Login	cwadm432
Database	RICS432
Password	*****
Max Connections	<input checked="" type="checkbox"/> Unlimited
Port Number	

Figure 10-49 InterChange Server Configuration: Security tab



## Preparing the environment for the first WebSphere ICS startup

Prior to starting the new installation of 4.3.0, additional steps are required to prepare the environment. These are outlined as follows.

1. Apply any custom configuration settings to `$CROSSWORLDS/InterchangeSystem.cfg`.

For example, to disable `DEADLOCK_DETECTOR_CHECK` for collaborations, edit the config file excerpt in Example 10-4 using a suitable editor for XML.

### *Example 10-4 Disable DEADLOCK\_DETECTOR\_CHECK*

---

```
<tns:property>
<tns:name>DEADLOCK_DETECTOR_CHECK</tns:name>
<tns:value xml:space="preserve">true</tns:value>
<tns:isEncrypted>false</tns:isEncrypted>
<tns:updateMethod>system restart</tns:updateMethod>
<tns:location>
<tns:reposController>false</tns:reposController>
<tns:reposAgent>false</tns:reposAgent>
<tns:localConfig>true</tns:localConfig>
</tns:location>
</tns:property>
```

---

Change `true` to `false` in the line starting with `<tns:value`.

2. Include external code and .jar files in `CLASSPATH`. Copy any custom Java classes to a location accessible by the new WebSphere InterChange Server 4.3.0 installation.

In this migration scenario, the external jar files are included in `$CROSSWORLDS/bin/CWSharedEnv.sh`. The external classes are first added to a new variable, `EXT_LIB`, which is then appended to the `CWCLASSES` variable. See the following excerpt from `CWSharedEnv.sh`:

```
# external classes added here
EXT_LIB=${CROSSWORLDS}/lib/Sample.jar

CWCLASSES=${WLICENSE}:${CORECW}:${DATAHANDLERS}:${DATADIRECT}:${DATAMANAGER}:${ITLM}:${JAVAMAIL}:${WEBSERVICES_JARS}:${EXT_LIB}
export CWCLASSES
```

3. Modify and verify the environment variables as documented in the Installation Guide for UNIX.

For example, make sure the path to the `JAVA 1.4.2` compiler precedes the `JAVA 1.3.1` compiler if the `JAVA 1.3.1` compiler is still in the path. This excerpt represents part of the `$HOME/.profile` from one of our test servers:

```
#Start DB Environment
. /home/db2inst1/sqllib/db2profile
export EXTSHM=ON
```

```
# End DB Environment

# WBI environment
export PATH=/usr/java14/bin:$PATH
. /opt2/ics43/bin/CWSharedEnv.sh
#END WBI environment
```

Make sure the latest settings are in effect. To confirm that the right Java Compiler is activated, run the **which javac** command.

The following output is returned for AIX, which is the installation path for Java 1.4.2 JDK:

```
/usr/java14/bin/javac
```

4. Install the WebSphere Business Integration Toolset on a supported Windows client. If the tools are to be installed on a Windows client that currently hosts an older version of tools, the old tools must be uninstalled prior to the installation of the new tools. It is recommended to take a backup of the Windows client before uninstalling and installing the new tools.

During installation, a prompt for the location of the Object Request Broker is displayed. Enter the host name or IP address of the machine hosting the Persistent Name Server that the WebSphere InterChange Server is using. Also enter the port number if it is different from the default, 14500. To change settings after installation, edit %CROSSWORLDS%\bin\CWSharedEnv.bat.

```
set ORB_PORT=14500
set ORB_HOST=persian.almaden.ibm.com
```

## Starting the new server

When the server is started for the first time after installation, the WebSphere InterChange Server repository tables are created. This section describes the steps to prepare and start the server for the first time.

1. Make sure that the following applications are started and running:
  - WebSphere MQ queue manager and Listener
  - Database
2. Start the Persistent Name Server. In a new console, type:
 

```
$CROSSWORLDS/bin/PersistentNameServer.sh
```
3. Start the WebSphere InterChange Server in design mode:
 

```
$CROSSWORLDS/bin/ics_manager -start -design
```

The WebSphere InterChange Server logs messages to the InterChangeSystem.log file located in the product directory. The message Interchange Server is Ready indicates that the WebSphere InterChange Server has started successfully.

## Migrating repository content to WebSphere ICS 4.3.0

Migrating the content of the WebSphere InterChange Server repository is one of the most important steps in the migration process. This section covers the steps required to complete this task.

1. Import the previous repository into the tooling:
  - a. Start the System Manager.
  - b. Import the Solution created in the backup step.

**Note:** If Export Solution is not used in the backup procedure, import the component libraries using exported integration component library files.

1. Create a new integration component library.
2. Right-click the newly created integration component library and select **Import Repository from File**. Choose the repository file that was created in the backup procedure. Repeat these steps if there are multiple integration component libraries to import. The *Implementation Guide for WebSphere InterChange Server* contains further information about this process.

3. Include custom .jar files in the CLASSPATH for the WebSphere Business Integration Toolset. Click **Windows** → **Preferences**, then navigate to **System Manager** → **Compiler** to view the CLASSPATH for the toolset. Add the location of the custom .jar files to this list (Figure 10-50).

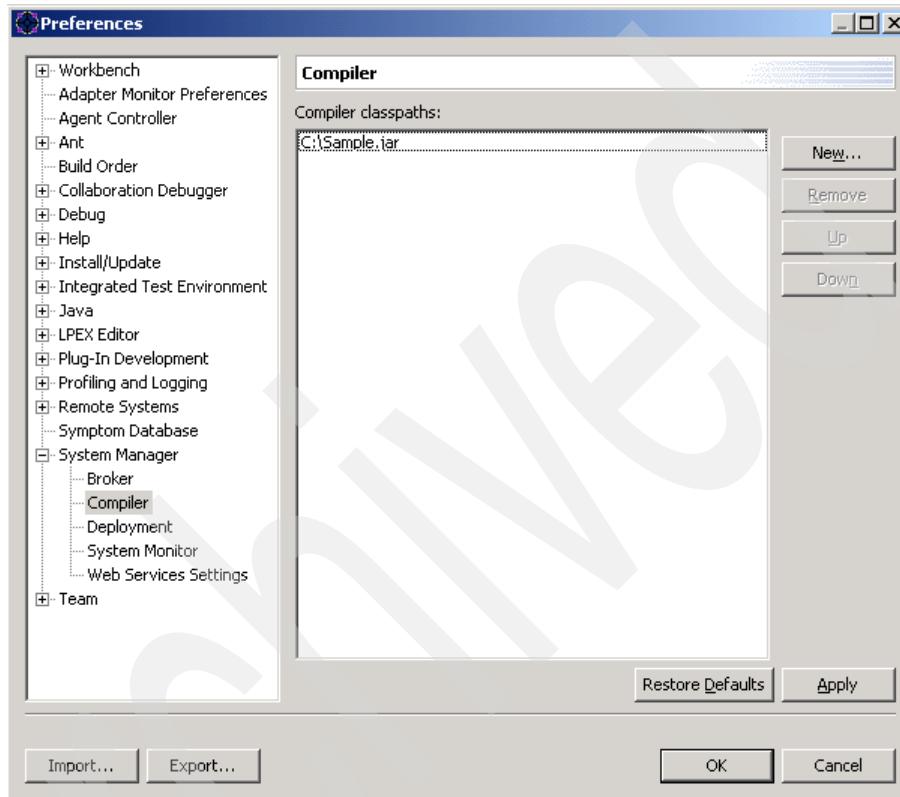


Figure 10-50 Setting the classpath for the toolset

4. Set the compliance of the compiler to Java 1.4. Click **Windows** → **Preferences**, then navigate to **Java** → **Compiler**. Under the JDK Compliance tag, change the Compiler compliance level to **1.4**. Figure 10-51 shows an example. A dialog box is displayed indicating that the compiler settings have changed. Choose **No** to not rebuild files at this time.

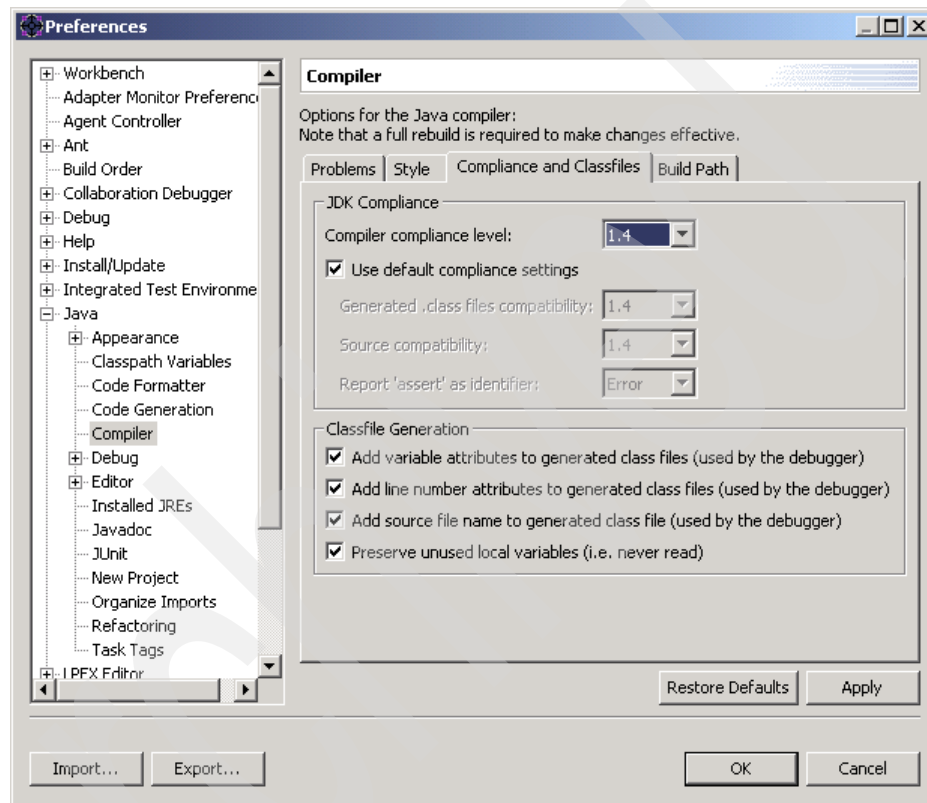


Figure 10-51 Setting the Compiler compliance level

5. If there is any custom code in the maps or collaboration templates, examine it for Java 1.4 compliance and make the appropriate changes. For example, one change between Java 1.3 and Java 1.4 is the use of the import statement. In Java 1.4, the import statement must be followed by a packaged class, but Java 1.3 allowed the import of unpackaged classes. The compliance is checked during compilation because the compiler compliance level was set correctly in the previous step.
6. Compile all maps and templates in the integration component libraries and examine the console for any error messages.

7. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Examine and reconfigure any collaboration templates that use a Long Lived Business Process Asynch Inbound call. In the Template Definitions examine the Port and Triggering Events tab. For the **BO Type** that represents the object used in the Asynch Inbound call, ensure that the scenarios are set to <None> for all verbs.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
8. Deploy the user project.
  - a. Right-click the user project and select **Deploy user project**.
  - b. Click **Select the Destination Server** and choose the server name.
  - c. Select the components to be deployed.
  - d. Select **Perform Compilation on Interchange Server machine**.
  - e. If the location of the relationship tables has changed, select **Create Schema**.
  - f. Click **Finish** to start the deployment.

**Tip:** If the database has been upgraded, the stored procedures required for these relationships may have been broken. If the integrations that use dynamic relationships are not working, redeploy the relationship with the **Create Schema** option. This re-creates the stored procedures and does not overwrite existing data.

9. Restart the ICS if prompted.

10. Completing relationship migration: If a new relationship location is used, the relationship data from the 4.2.2.4 relationship tables can be migrated.
  - a. For both static and dynamic relationships, copy the data from each relationship table to the new relationship table. (Consult with the database administrator.) The steps to complete this process using DB2 are:
    - i. In the Control Center, right-click the original relationship table and choose Export (Figure 10-52).

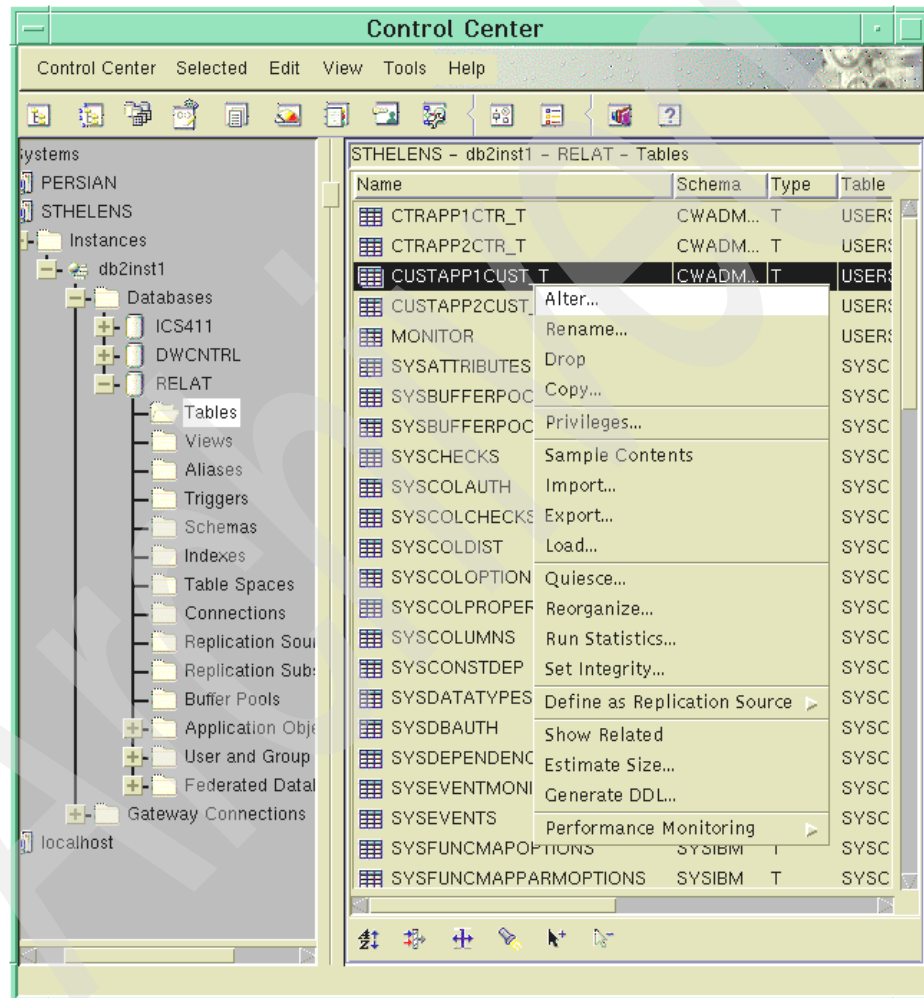


Figure 10-52 Exporting a table

- ii. In the new window, check for output file name and location and choose an export type. Click **OK** to export the data.

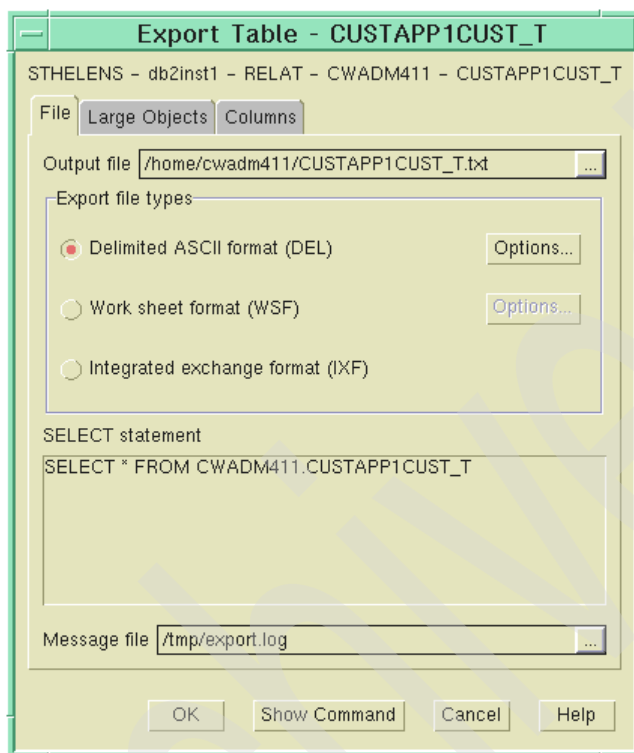


Figure 10-53 Completing the table export



- iii. In the Control Center, right-click the newly created relationship table and choose **Import** as shown in Figure 10-54.

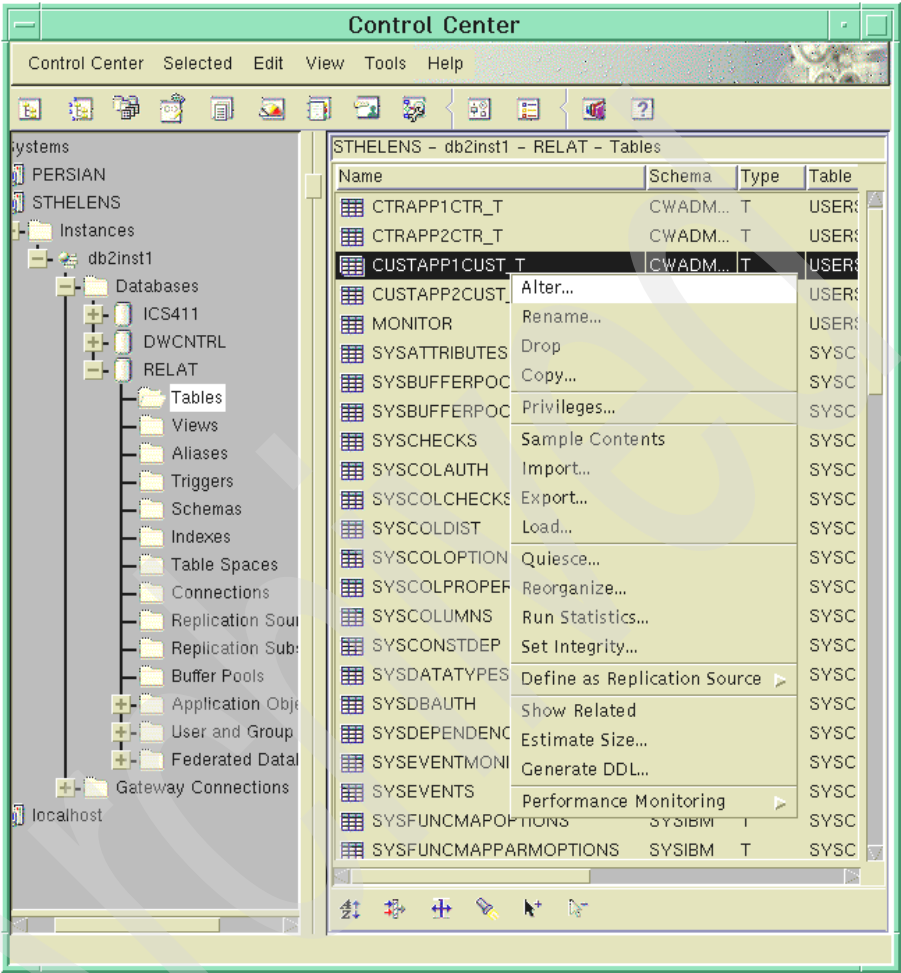


Figure 10-54 Control Center after right-clicking a table

- iv. Click **Import**. The window in Figure 10-55 opens. Browse to the export file by clicking the ... button next to Import file and set the correct Import file type. Click **OK** to import the data.

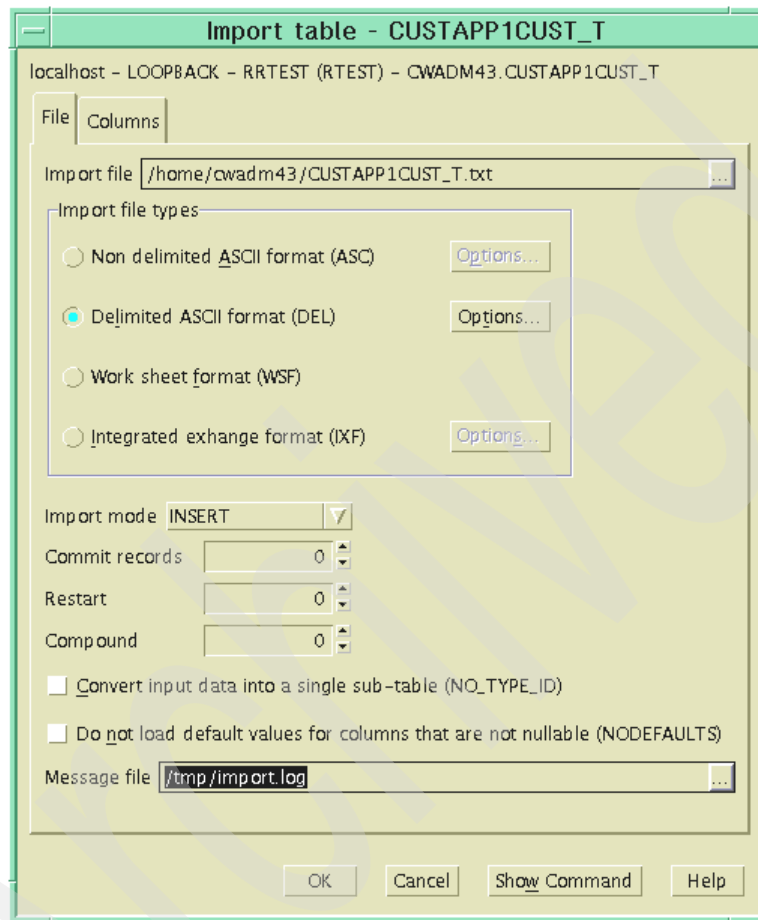


Figure 10-55 Importing the data

- b. Additional steps are needed to set the sequence number for each relationship. Here are the steps to set the sequence number when the relationships are stored in DB2:
  - i. Find the current highest ID for any relationship by examining the INSTANCEID column for any participant table in the relationship.

```
SELECT MAX(INSTANCEID) FROM DB2ADMIN.CUSTAPP1CUST_T
```

In this example, the query has returned the value 41 as the highest INSTANCEID.

- ii. Find the sequence variable name. The following command generates a SQL script that provides information about the relationship table. This information can also be obtained in the Control Center by right-clicking a database table and choosing **Generate DDL**.

```
db2look -dRELAT -tCUSTAPP1CUST_T -a -e -c -oOUTPUT.SQL
```

Examine the *OUTPUT.SQL* file to obtain the sequence variable name. Search for the text CREATE SEQUENCE.

```
CREATE SEQUENCE "DB2ADMIN"."CUSTOMER_1_0_0" AS INTEGER  
MINVALUE 1 MAXVALUE 2147483647 START WITH 1 INCREMENT BY 1  
CACHE 20 NO CYCLE NO ORDER;
```

In the previous example, the sequence variable name is CUSTOMER\_1\_0\_0. The sequence name is derived from the name of the relationship. In this case the relationship name is CUSTOMER.

- iii. Set the sequence variable to be at least one larger than the current number. In this example, the current highest INSTANCEID is 41 so the sequence is set to 42 with the following command.

```
alter sequence Customer_1_0_0 restart with 42
```

- c. Verify that the relationship has been moved successfully to the new database by examining the relationship using Relationship Manager.

### 10.4.3 Post-migration steps

This section describes the steps to take after migration completes and the WebSphere InterChange Server has been started successfully. This includes setting up the new security on the WebSphere InterChange Server and testing that the migration was completed successfully.

#### Security

To achieve the same level of security that was available before the migration, it is necessary to create the admin user with the previously used password, assigned to the administrator role. The following steps and figures show how to do this:

1. In System Manager, connect to the WebSphere InterChange Server.

2. Right-click the server name in the Interchange component management view and select **Users and Roles**.

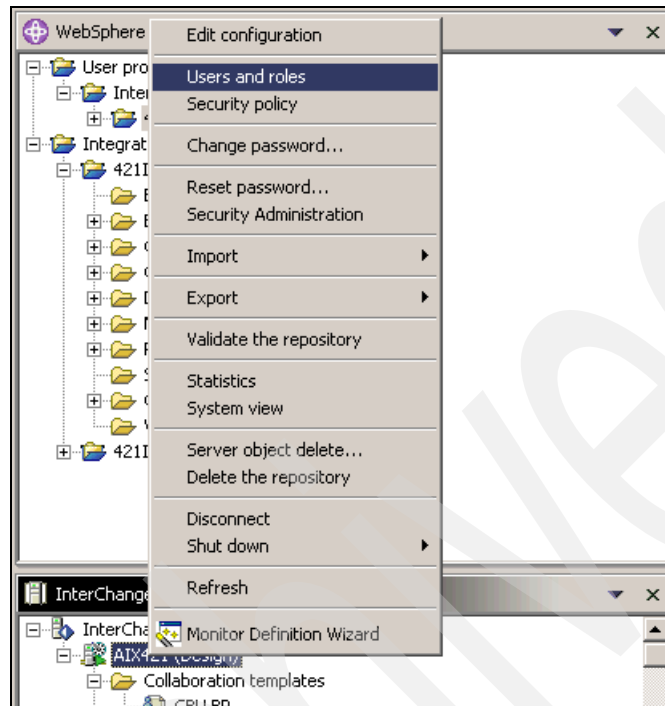


Figure 10-56 Selecting Users and roles

3. A new window called Users and roles management is displayed. Right-click **Users** and select **New User**.

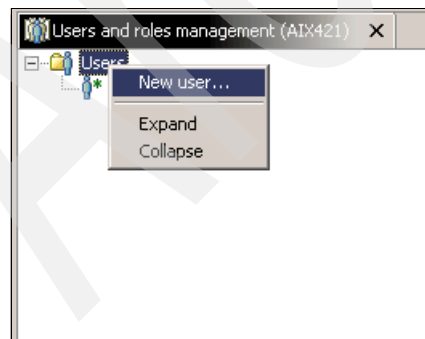


Figure 10-57 Selecting new user

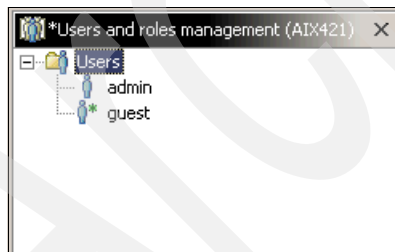
4. The Create user window opens. Enter the name admin and the previous password, plus additional information if needed. Click **OK**.



A screenshot of a 'Create user' dialog box. It has a title bar with a close button. The dialog contains several text input fields: 'User name' (containing 'admin'), 'Password' (containing '\*\*\*\*'), 'Confirm password' (containing '\*\*\*\*'), 'Distinguished name' (empty), 'Common name' (empty), and 'Surname' (empty). At the bottom right are 'OK' and 'Cancel' buttons.

*Figure 10-58 Creating a user*

5. The new user is added to the Users and roles management window. This is shown in Figure 10-59.



*Figure 10-59 The newly created admin user*

6. Right-click the newly created **admin** user and select **Add roles**.

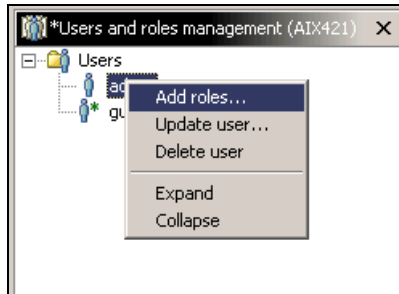


Figure 10-60 Add roles

7. The Role list window is displayed. Select **administrator** and click **OK**.

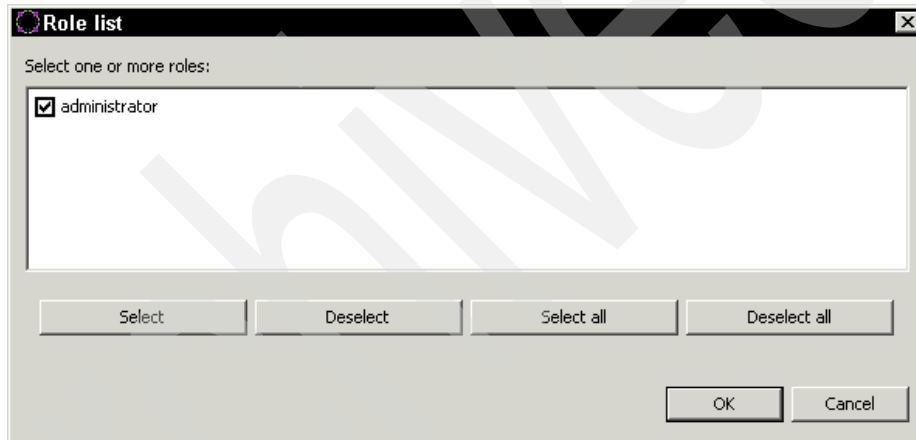


Figure 10-61 Role list

8. Close the Users and roles management window.

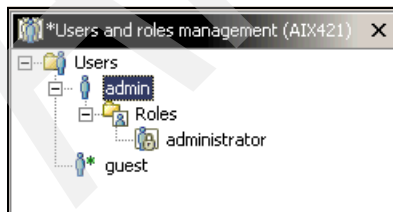


Figure 10-62 User admin with administrator role

9. Right-click the server again in the InterChange Component management view and select **Edit Configuration**. An editor opens to configure the server.

10. Click the **Security-RBAC** tab and apply the following settings:
  - a. Select **Enable RBAC**.
  - b. Set **Server start user** to admin.
  - c. Set **Server start password** to the password for the admin user.
  - d. Check all other settings and modify if necessary.
  - e. Save to the server by clicking the save icon or selecting **File** → **Save**.

Figure 10-63 Server Configuration: Security-RBAC tab

11. Reboot WebSphere InterChange Server to activate security.

**Note:** The adapter agent defaults to the guest account if RBAC is activated for WebSphere InterChange Server and no user name/password is provided.

## Enabling the WebSphere InterChange Server for adapters

To enable WebSphere Business Integration Adapters to work with the WebSphere InterChange Server 4.3.0, version 2.6 of a WebSphere Business Integration Adapter must be installed. This requires the Adapter Framework 2.6 to be installed on every machine hosting an adapter. This includes the Visual Test Connector. For more detailed instructions about installing adapters, refer to the individual adapter guides.

## Server Access Interface

If any access clients use the Server Access Interface, they must be updated to work with WebSphere InterChange Server 4.3.0. Refer to Chapter 5, “Technical impact of major design changes” on page 67.

## Testing

Testing is an important part of the migration process. Be sure to perform unit, end-to-end, and performance testing. Before moving the upgraded WebSphere InterChange Server system from development to production, you should perform tests on every interface and every business process in production. Refer to the *Implementation Guide for WebSphere InterChange Server* for further information about testing a migrated system.

Table 10-6 shows the scenarios that were tested in this migration and the results that were found. These correlate with the test scenarios that are described in Chapter 7, “Project environment setup” on page 109.

A test scenario is considered to be successfully migrated if there is no loss or change of functionality from the previous version when tested with WebSphere InterChange Server 4.3.0.

Table 10-6 Test scenarios results

Test scenario	Results
Long Lived Business Process	Successfully migrated
CustomerSync Collaboration including relationships	Successfully migrated
Connection Pool	Successfully migrated
Collaboration Group	Successfully migrated
Polymorphic Map	Successfully migrated
Server Access Interface	Successfully migrated
Transactional Collaboration	Successfully migrated
Failed Flows with different transports	Successfully migrated
Schedules	Successfully migrated
Imported external code	Successfully migrated
Web-based System Monitor	Successfully migrated



**Note:** After all the system tests are performed and the InterChange Server is ready for production, ensure that WebSphere InterChange Server is re-started in Production mode.

### Uninstall (optional)

This section describes the steps to uninstall the previous version of WebSphere InterChange Server and the related software. The following steps list the order of uninstalling relevant software

12. Uninstall JDK as the root user using **smit**, only if it is not required by other products. For this release, the JDK is typically Java 2 SDK V.1.3.1. Contact the system administrator for further assistance.
13. Uninstall the IBM CrossWorlds InterChange Server using the uninstaller located in the product directory when it is certain that the previous version is no longer needed. After the uninstaller has completed, the product directory can be deleted.





## Part 3

# Appendixes





# WebSphere InterChange Server migration checklist

This appendix provides the a checklist of the migration steps for each version covered in this book. The details of the individual steps can be found in the relevant migration chapter.

This chapter has three sections, one section for each WebSphere InterChange Server (WebSphere ICS) version. Each section is then sub-divided by platform, and then by without in-place database migration and in-place database migration to mirror the structure of the migration chapters. Refer to the relevant section for the migration version, platform, and method required.

# Migration checklist for WebSphere ICS 4.1.1

This section lists the steps that are required to migrate WebSphere InterChange Server 4.1.1.9 to WebSphere InterChange Server 4.3.0. It is divided into sections for Windows and AIX and for in-place database migration and for without in-place database migration.

## Windows: in-place database migration

This section provides the steps necessary to perform an in-place database migration of WebSphere InterChange Server 4.1.1.9 to WebSphere InterChange Server 4.3.0 on Windows.

### Pre-migration steps

Table A-1 details the pre-migration steps for an in-place database migration of WebSphere InterChange Server 4.1.1 on Windows.

Table A-1 Windows in-place database pre-migration steps for 4.1.1

1. Put the environment in a quiescent state.

a. Stop all adapters from polling.

b. Resolve transactional status of transactional collaborations.

c. Resubmit failed events or discard the events.

d. Stop all collaborations.

2. Backing Up and Shutting Down Components.

a. Back up the current InterChange Server repository using the repos\_copy utility.

b. Shut down the InterChange Server.

c. Back up the database.

d. Shut down the VisiBroker Service.

e. Back up WebSphere MQ environment.

f. Save environment settings.
-

Migration steps

Table A-2 details the migration steps for an in-place database migration of WebSphere InterChange Server 4.1.1 on Windows.

Table A-2 Windows in-place database migration steps for 4.1.1

- 1. Preparing the Database System.
  - a. Upgrade Database.
  - b. Check the database parameters.
  - c. For DB2 test the c-compiler.
  - d. Restart database.
- 2. Upgrade WebSphere MQ.
- 3. Uninstall Software.
  - a. Uninstall Windows services (if any are present for example: WebSphere Business Integration Adapters, WebSphere InterChange Server).
  - b. Uninstall the Object Request Broker (VisiBroker).
  - c. Uninstall Java Development Kit (JDK) if this is not required for other products.
  - d. Uninstall WebSphere InterChange Server.
- 4. Install and Configuration.
  - a. Install WebSphere InterChange Server 4.3.
  - b. Configure the WebSphere InterChange Server with ICSCfg.bat.
- 5. Preparing the environment for the first WebSphere InterChange Server startup.
  - a. Copy DLMS and collaboration directories to new installation.
  - b. Apply any custom configuration settings to %CROSSWORLDS%/InterchangeSystem.cfg.
  - c. Modify any custom startup or other scripts that reference VisiBroker ORB.
  - d. Copy external jar files and include them in the WebSphere InterChange Server CLASSPATH.


- e. Modify and verify environment variables: CROSSWORLDS, MQ\_LIB, PATH.
6. Start the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.
  - c. Start the PeristentNameServer.bat.
  - d. Run the DDL script in Appendix B, "Sample code" on page 465.
  - e. Start the WebSphere InterChange Server in design mode.
7. Create a new integration component library and import the components from the server.
8. Include custom .jar files in the Toolset CLASSPATH.
9. Set the compliance of the compiler to Java 1.4.
10. Examine custom code in maps or collaboration templates for Java 1.4 compliance.
11. Compile all maps and templates in the integration component libraries.
12. Modify Adapters and queues for JMS delivery transport.
  - a. Examine/Modify queue names in Connector Configurator.
  - b. Create new queues in the queue manager.
  - c. Save the configuration file to the project, and save a copy to file.
13. Reconfigure components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.



- 14.Create a new user project and populate it the component from the integration component library.
- 15.Resolve any collaborations that are transactional status.
- 16.Delete the repository.
- 17.Deploy the user project.
  - a. Perform compilation on WebSphere InterChange Server machine.
  - b. (Optional) Select Create Schema option if the relationship location has changed or is not working.
  - c. Restart the WebSphere InterChange Server if prompted.


Post-migration steps

Table A-3 details the post-migration steps for an in-place database migration of WebSphere InterChange Server 4.1.1 on Windows.

Table A-3 Windows in-place database post-migration steps for 4.1.1

- 1. Security.
  - a. Create the admin user.
  - b. Use previous password.
  - c. Assign to Administrator role.
- 2. Update any Server Access Interface clients.
- 3. Install Adapter Framework 2.6 and upgrade adapters.
- 4. Test the WebSphere InterChange Server.


# Windows: without in-place database migration

This section provides the steps necessary to perform a without in-place database migration of WebSphere InterChange Server 4.1.1.9 to WebSphere InterChange Server 4.3.0 on Windows.

## Pre-migration steps

Table A-4 details the pre-migration steps for without in-place database migration of WebSphere InterChange Server 4.1.1 on Windows.

Table A-4 Windows without in-place database pre-migration steps for 4.1.1

1. Put the environment in a quiescent state.

a. Stop all adapters from polling.

b. Resolve transactional status of transactional collaborations.

c. Resubmit failed events or discard the events.

d. Stop all collaborations.
2. Backing Up and Shutting Down Components.

a. Back up the current InterChange Server repository using the repos\_copy utility.

b. Shut down the WebSphere InterChange Server.

c. Back up the database.

d. Shut down the VisiBroker Service.

e. Back up the WebSphere MQ environment.

f. Clear WebSphere MQ queues of any outstanding messages.

g. Save the environment settings.


## Migration steps

Table A-5 details the migration steps for without in-place database migration of WebSphere InterChange Server 4.1.1 on Windows.

Table A-5 Windows without in-place database migration steps for 4.1.1

1. Update the Database System.

a. Upgrade Database.

b. Prepare an empty database for the 4.3.0 repository.




- c. Start the PeristentNameServer.bat.
  - d. Edit the 4.1.1 repository file.
    - i. Connection Pools: Delete the definition if the connectionpool definition has changed.
    - ii. Relationship definition: Delete the entire lines for DatabaseURL, LoginName, LoginPassword, if the location has changed.
    - iii. Delete all schedule entries.
  - e. Start the WebSphere InterChange Server in design mode.
  - f. Load Repository content to WebSphere InterChange Server 4.3.0.
  - g. Load the modified repository file using the repos\_copy command line utility. (If the relationship definition has been edited, add the option -r\*).
  - h. Restart the WebSphere InterChange Server after a successful repos\_copy.
- 7. Create a new integration component library and import the components from the server.
  - 8. Include custom .jar files in the Toolset CLASSPATH.
  - 9. Set the compliance of the compiler to Java 1.4.
  - 10. Examine custom code in maps or collaboration templates for Java 1.4 compliance.
  - 11. Compile all the maps and templates in the integration component libraries.
  - 12. Examine/Modify Adapters that use JMS delivery transport.
    - a. Examine/Modify queue names in Connector Configurator.
    - b. Create new queues in the queue manager.
    - c. Save the configuration file to the project, and save a copy to file.
  - 13. Reconfigure any of the other components that require modification.


- a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
14. Create a new user project and populate it the component from the integration component library.
  15. Resolve any collaborations that are transactional status.
  16. Delete the repository.
  17. Deploy the user project.
    - a. Perform compilation on WebSphere InterChange Server machine.
    - b. (Optional) Select **Create Schema** option if the relationship location has changed or is not working.
    - c. Restart the WebSphere InterChange Server if prompted.
  18. Completing relationship migration.
    - a. Copy the data from each relationship table to the new relationship table.
    - b. Set the sequence number.
    - c. Verify the relationship with Relationship Manager.


## Post-migration steps

Table A-6 details the post-migration steps for without in-place database migration of WebSphere InterChange Server 4.1.1 on Windows.

*Table A-6 Windows without in-place database post-migration steps for 4.1.1*

1. Security.
  - a. Create the admin user.
  - b. Use previous password.
  - c. Assign to Administrator role.


2. Install Adapter Framework 2.6 and upgrade adapters.
3. Update any Server Access Interface clients.
4. Test the WebSphere InterChange Server.


## AIX: in-place database migration

This section provides the steps required to perform an in-place database migration of WebSphere InterChange Server 4.1.1.9 to WebSphere InterChange Server 4.3.0. on AIX.

### Pre-migration steps

Table A-7 details the pre-migration steps for an in-place database migration of WebSphere InterChange Server 4.1.1 on AIX.

*Table A-7 AIX in-place database pre-migration steps for 4.1.1*

1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling.
  - b. Resolve transactional status of transactional collaborations.
  - c. Resubmit failed events or discard the events.
  - d. stop all collaborations.
2. Backing Up and Shutting Down Components.
  - a. Back up the current InterChange Server repository using the repos\_copy utility.
  - b. Shut down the WebSphere InterChange Server.
  - c. Back up the database.
  - d. Shut down the Inprise VisiBroker Agent.
  - e. Back up the WebSphere MQ environment.


### Migration steps

Table A-8 details the migration steps for an in-place database migration of WebSphere InterChange Server 4.1.1 on AIX.

*Table A-8 AIX in-place database migration steps for 4.1.1*

1. Upgrade the Database System.

--







Post-migration steps

Table A-9 details the post-migration steps for an in-place database migration of WebSphere InterChange Server 4.1.1 on AIX.

Table A-9 AIX in-place database post-migration steps for 4.1.1

- 1. Security.
  - a. Create the admin user.
  - b. Set previous password.
  - c. Assign to Administrator role.
- 2. Install Adapter Framework 2.6 and upgrade adapters.
- 3. Update any Server Access Interface clients.
- 4. Test the WebSphere InterChange Server.
- 5. Uninstall (optional).
  - a. Uninstall the ORB (VisiBroker).
  - b. Uninstall JDK as the root user using `smi t`, only if it is not required by other products.
  - c. Uninstall WebSphere InterChange Server 4.1.1 once it is no longer needed.


AIX: without in-place database migration

This section provides the steps necessary to perform a without in-place database migration of WebSphere InterChange Server 4.1.1.9 to WebSphere InterChange Server 4.3.0.

Pre-migration steps

Table A-10 details the pre-migration steps for without in-place database migration of WebSphere InterChange Server 4.1.1 on AIX.

Table A-10 AIX without in-place database pre-migration steps for 4.1.1

- 1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling.
  - b. Resolve transactional status of transactional collaborations.
  - c. Resubmit failed events or discard the events.


- d. Stop all collaborations.
2. Backing Up and Shutting Down Components.
  - a. Back up the current InterChange Server repository using the **repos\_copy** utility.
  - b. Shut down the WebSphere InterChange Server.
  - c. Back up the database.
  - d. Back up the WebSphere MQ environment.
  - e. Clear WebSphere MQ queues of any outstanding messages.


## Migration steps

Table A-11 details the migration steps for without in-place database migration of WebSphere InterChange Server 4.1.1 on AIX.

*Table A-11 AIX without in-place database migration steps for 4.1.1*

1. Upgrade the Database System.
  - a. Upgrade Database.
  - b. Prepare an empty database for the 4.3.0 repository.
  - c. Check the database parameters.
  - d. For DB2 test the c-compiler.
  - e. For DB2 Catalog database with Loopback Node.
  - f. Restart database.
2. Upgrade WebSphere MQ.
3. Installation.
  - a. Install WebSphere InterChange Server 4.3.
  - b. Configure the WebSphere InterChange Server with ICSCfg.sh.
4. Preparing the environment for the first WebSphere InterChange Server startup.
  - a. Copy DLMS and collaboration directories to new installation.


- b. Apply any custom configuration settings to `$CROSSWORLDS/InterchangeSystem.cfg`.
  - c. Modify any custom startup or other scripts that reference VisiBroker ORB.
  - d. Copy external jar files and include them in the WebSphere InterChange Server CLASSPATH.
  - e. Modify and verify the environment variables and make sure the latest settings are in effect.
  - f. Install the Toolset on a supported Windows client.
5. Start the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.
  - c. Start the Persistent Name Server.
  - d. Edit the 4.1.1 repository file.
    - i. Connection Pools: Delete the definition if the connectionpool definition has changed.
    - ii. Relationship definition: Delete the entire lines for DatabaseURL, LoginName, LoginPassword, if the location has changed.
    - iii. Delete all schedule entries.
  - e. Start the WebSphere InterChange Server in design mode.
  - f. Load the modified repository file using the `repos_copy` command line utility (If the relationship definition has been edited, add the option `-r*`).
  - g. Restart the WebSphere InterChange Server after a successful `repos_copy`.
6. Create a new integration component library and import the components from the server.
7. Include custom .jar files in the Toolset CLASSPATH.
8. Set the compliance of the compiler to Java 1.4.

9. Examine custom code in maps or collaboration templates for Java 1.4 compliance.
10. Compile all the maps and templates in the integration component libraries.
11. Examine/Modify Adapters that use JMS delivery transport.
  - a. Examine/Modify queue names in Connector Configurator.
  - a. Create new queues in the queue manager.
  - a. Save the configuration file to the project, and save a copy to file.
12. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface.
  - c. Reconfigure any custom code or property that references a specific queue manager, queue name, or file location that has changed.
13. Create a new user project and populate it the component from the integration component library.
14. Resolve any collaborations that are in transactional status.
15. Delete the repository.
16. Deploy the modified components.
  - a. Perform compilation on WebSphere InterChange Server machine.
  - b. (Optional) Select **Create Schema** option if the relationship location has changed or is not working.
  - c. Restart the WebSphere InterChange Server if prompted.
17. Completing relationship migration.
  - a. Copy the data from each relationship table to the new relationship table.


- b. Set the sequence number.
- c. Verify the relationship with Relationship Manager.


### Post-migration steps

Table A-12 details the post-migration steps for a without in-place database migration of WebSphere InterChange Server 4.1.1 on AIX.

Table A-12 AIX without in-place database post-migration steps for 4.1.1

- 1. Security.
  - a. Create the admin user.
  - b. Set previous password.
  - c. Assign to Administrator role.
- 2. Install Adapter Framework 2.6 and upgrade adapters.
- 3. Update any Server Access Interface clients.
- 4. Test the WebSphere InterChange Server.
- 5. Uninstall (optional).
  - a. Uninstall the ORB (VisiBroker).
  - b. Uninstall JDK as the root user using `smit`, only if it is not required by other products.
  - c. Uninstall WebSphere InterChange Server 4.1.1 once it is no longer needed.


# Migration checklist for WebSphere ICS 4.2.1

This section provides the steps required to migrate WebSphere InterChange Server 4.2.1.7 to WebSphere InterChange Server 4.3.0. It is divided into different sections for Windows and AIX and for in-place database migration and for without in-place database migration.

## Windows: in-place database migration

This section provides the steps necessary to perform an in-place database migration of WebSphere InterChange Server 4.2.1.7 to WebSphere InterChange Server 4.3.0 on Windows.

### Pre-migration steps

Table A-13 details the pre-migration steps for an in-place database migration of WebSphere InterChange Server 4.2.1 on Windows.

Table A-13 Windows in-place database pre-migration steps for 4.2.1

1. Put the environment in a quiescent state.

a. Stop all adapters from polling.

b. Resolve transactional status of transactional collaborations.

c. Resubmit failed events or discard the events using Flow Manager.

d. Stop all collaborations.

2. Backing Up and Shutting Down Components.

a. Back up InterChange Server repository.

b. Back up Custom collaborations components.

c. Back up Custom adapters.

d. Back up Map components.

e. Back up Relationship database tables.

f. Back up Log files.

g. Back up user projects.

h. Shut down the WebSphere InterChange Server.

i. Back up the database.
- 
- 436 WebSphere InterChange Server Migration Scenarios

- j. Shut down the VisiBroker ORB Service.
- k. Back up the WebSphere MQ environment and message data.
- l. Stop WebSphere MQ.
- 3. Uninstallation.
  - a. Uninstall Windows services (if any are present for example: WebSphere Business Integration Adapters, WebSphere InterChange Server).
  - b. Uninstall Java Development Kit (JDK) if not used by other applications.
  - c. Uninstall VisiBroker ORB.
  - d. Uninstall WebSphere InterChange Server.


## Migration steps

Table A-14 details the migration steps for an in-place database migration of WebSphere InterChange Server 4.2.1 on Windows.

*Table A-14 Windows in-place database migration steps for 4.2.1*

- 1. Upgrade the Database System.
  - a. Upgrade Database.
  - b. Check the database parameters.
  - c. For DB2 test the c-compiler.
  - d. Restart Database.
- 2. Upgrade WebSphere MQ.
- 3. Installation.
  - a. Install WebSphere InterChange Server 4.3.
  - b. Configure the WebSphere InterChange Server with ICSCfg.bat.
  - c. Copy DLMs and collaboration directories to new installation.
  - d. Apply any custom configuration settings to %CROSSWORLDS%/InterchangeSystem.cfg.






- c. Restart WebSphere InterChange Server if prompted.

--

### Post-migration steps

Table A-15 details the post-migration steps for an in-place database migration of WebSphere InterChange Server 4.2.1 on Windows.

Table A-15 Windows in-place database post-migration steps for 4.2.1

1. Security.
  - a. Create user admin/null.
  - b. Assign Administrator role to user admin.
  - c. Enable Role-based Access Control in WebSphere InterChange Server.
  - d. Restart WebSphere InterChange Server for this to become active.
2. Install Adapter Framework 2.6 and upgrade adapters.
3. Update any Server Access Interface clients.
4. Test the WebSphere InterChange Server.


## Windows: without in-place database migration

This section provides the steps necessary to perform a without in-place database migration for WebSphere InterChange Server 4.2.1.7 to WebSphere InterChange Server 4.3.0.

### Pre-migration steps

Table A-16 details the pre-migration steps for without in-place database migration for WebSphere InterChange Server 4.2.1 on Windows.

Table A-16 Windows without in-place database pre-migration steps for 4.2.1

1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling.
  - b. Resolve transactional status of transactional collaborations.
  - c. Resubmit failed events or discard the events using Flow Manager.
  - d. Stop all collaborations.


2. Backing Up and Shutting Down Components.
  - a. Back up InterChange Server repository using repos\_copy utility.
  - b. Back up Custom collaborations components.
  - c. Back up Custom adapters.
  - d. Back up Map components.
  - e. Back up Relationship database tables.
  - f. Back up Log files.
  - g. Back up user projects.
  - h. Shut down the WebSphere InterChange Server.
  - i. Back up the database.
  - j. Shut down the VisiBroker ORB Service.
  - k. Back up the WebSphere MQ environment and message data.
  - l. Stop WebSphere MQ.
3. Uninstallation
  - a. Uninstall Windows services (if any are present for example: WebSphere Business Integration Adapters, WebSphere InterChange Server).
  - b. Uninstall Java Development Kit (JDK) if not used by other applications.
  - c. Uninstall VisiBroker ORB.
  - d. Uninstall WebSphere InterChange Server.


### Migration steps

Table A-17 details the migration steps for without in-place database migration for WebSphere InterChange Server 4.2.1 on Windows.

Table A-17 Windows without in-place database migration steps for 4.2.1

1. Upgrade the Database System.
  - a. Upgrade Database.


- b. Create an empty database for the 4.3.0 repository.
  - c. Check the database parameters.
  - d. For DB2 test the c-compiler.
  - e. Restart Database.
2. Upgrade WebSphere MQ.
3. Installation
  - a. Install WebSphere InterChange Server 4.3.
  - b. Configure the WebSphere InterChange Server with ICSCfg.bat.
  - c. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.
  - d. Modify any startup or custom scripts that reference VisiBroker ORB.
  - e. Include external code and jar files in CLASSPATH.
  - f. Modify and verify the environment variables: CROSSWORLDS, MQ\_LIB, PATH.
4. Starting the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.
  - c. Start the Persistent Name Server.
  - d. Start the WebSphere InterChange Server in design mode.
5. Import the previous repository into the tooling.
6. Toolset: Include custom .jar files in the CLASSPATH.
7. Set the Compiler compliance level to JDK 1.4.
8. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.


- b. Reconfigure any custom code that uses Server Access Interface.
  - c. Examine and set the triggering events for LLBP Asynch Inbound calls to <None>.
  - d. Reconfigure any custom code or property that references a changed property.
9. Resolve any collaborations that are in transactional status.
10. Deploy modified components.
  - a. Redeploy all components to have all administrative features later.
  - b. (Optional) Select **Create Schema** option if the relationship location has changed or is not working.
  - c. Restart WebSphere InterChange Server if prompted.
11. Complete relationship migration.
  - a. Copy the data from each relationship table to the new relationship table.
  - b. Set the sequence number.
  - c. Verify the relationship with Relationship Manager.


### Post-migration steps

Table A-18 details the post-migration steps for without in-place database migration for WebSphere InterChange Server 4.2.1 on Windows

*Table A-18 Windows without in-place database post-migration steps for 4.2.1*

1. Security.
  - a. Create user admin/null.
  - b. Assign Administrator role to user admin.
  - c. Enable Role-based .Access Control in WebSphere InterChange Server
2. Install Adapter Framework 2.6 and upgrade adapters.
3. Update any Server Access Interface clients.
4. Test the WebSphere InterChange Server.


# AIX: in-place database migration

This section provides the steps necessary to perform an in-place database migration for WebSphere InterChange Server 4.2.1.7 to WebSphere InterChange Server 4.3.0 on AIX.

## Pre-migration steps

Table A-19 details the pre-migration steps for an in-place database migration for WebSphere InterChange Server 4.2.1 on AIX.

Table A-19 AIX in-place database pre-migration steps for 4.2.1

- 1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling.
  - b. Resolve transactional status of transactional collaborations.
  - c. Resubmit failed events or discard the events using Flow Manager.
  - d. Stop all collaborations.
- 2. Backing Up and Shutting Down Components.
  - a. Back up of WebSphere InterChange Server repository.
  - b. Back up Custom collaborations components (Java class files, message files, Custom connectors).
  - c. Back up Map components (map design files, Java class files, and message files).
  - d. Back up Relationship database tables.
  - e. Back up Log files.
  - f. Back up user projects.
  - g. Shut down WebSphere InterChange Server.
  - h. Back up the database.
  - i. Stop Persistent Name Server.
  - j. Back up WebSphere MQ environment and message data.
  - k. Stop WebSphere MQ.


## Migration steps

Table A-20 details the migration steps for an in-place database migration for WebSphere InterChange Server 4.2.2 on AIX.

*Table A-20 AIX in-place database migration steps for 4.2.1*

1. Upgrading the Database System.
  - a. Upgrade database.
  - b. Check the database parameters.
  - c. For DB2 test the c-compiler.
  - d. For DB2 Catalog database with Loopback Node.
  - e. Restart database.
2. Upgrade WebSphere MQ.
3. Installation WebSphere InterChange Server.
  - a. Create new icsadmin user.
  - b. Add to group mqm/dbadmins.
  - c. Install to new directory.
  - d. Apply old settings to configuration from WebSphere InterChange Server 4.2.2.4
  - e. Apply security settings as needed.
  - f. Copy DLMs and collaboration directories to new installation.
  - g. Apply custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.
  - h. Modify custom scripts to reflect new IBM ORB settings.
  - i. Include external code and jar-files in CLASSPATH.
  - j. Modify and verify the environment variables.
  - k. Install the WebSphere Business Integration Toolset.
4. Starting the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.

[illegible]

- c. Start the Persistent Name Server.
  - d. Start the WebSphere InterChange Server in design mode.
5. Import the previous repository into the tooling.
6. Toolset: Include custom .jar files in the CLASSPATH.
7. Setting the Compiler compliance level to JDK 1.4.
8. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Reconfigure any custom code that uses Server Access Interface.
  - c. Examine and set the triggering events for LLBP Asynch Inbound calls to <None>.
  - d. Reconfigure any custom code or property that references a changed property.
9. Resolve any collaborations that are transactional status.
10. Deploy modified components.
  - a. (Optional) Redeploy all components to have all administrative features later.
  - b. Restart WebSphere InterChange Server if necessary.


## Post-migration steps

Table A-21 details the post-migration steps for an in-place database migration for WebSphere InterChange Server 4.2.1 on AIX.

*Table A-21 AIX in-place database post-migration steps for 4.2.1*

1. Security.
  - a. Create user admin/password null in administrator role.
  - b. Enable Role-based Access Control.
  - c. Restart WebSphere InterChange Server for this to become active.
2. Enable WebSphere InterChange Server for Adapters.


3. Install Adapter Framework 2.6.
4. Modify Server Access Interface client settings.
5. Test the WebSphere InterChange Server.
6. Uninstall (optional).
  - a. JDK 1.3.1. (if not needed by other applications).
  - b. WebSphere InterChange Server 4.2.1.
  - c. VisiBroker.


## AIX: without in-place database migration

This section provides the steps necessary to perform a without in-place database migration for WebSphere InterChange Server 4.2.1.7 to WebSphere InterChange Server 4.3.0.

### Pre-migration steps

Table A-22 details the pre-migration steps for without in-place database migration for WebSphere InterChange Server 4.2.1 on AIX.

*Table A-22 AIX without in-place database pre-migration steps for 4.2.1*

1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling the event tables.
  - b. Resolve transactional status of transactional collaborations.
  - c. Resubmit failed events or discard the events.
  - d. Stop all collaborations.
2. Backing Up and Shutting Down Components.
  - a. Back up WebSphere InterChange Server repository.
  - b. Back up Custom collaborations components (Java class files, message files, custom connectors).
  - c. Back up Map components (map design files, Java class files, message files).
  - d. Back up Relationship database tables.
  - e. Log files (as desired for historical information).




- f. Back up user projects.
- g. Shut down the WebSphere InterChange Server.
- h. Back up the database.
- i. Stop PersistentNameServer.
- j. Back up the WebSphere MQ environment.


## Migration steps

Table A-23 details the migration steps for without in-place database migration for WebSphere InterChange Server 4.2.1 on AIX.

*Table A-23 AIX without in-place database migration steps for 4.2.1*

1. Upgrading the Database System.
  - a. Upgrade database.
  - b. Prepare an empty database for the 4.3.0 repository.
  - c. Check the database parameters.
  - d. For DB2 test the c-compiler.
  - e. For DB2 Catalog database with Loopback Node.
  - f. Restart database.
2. Upgrade WebSphere MQ.
3. Installation of WebSphere InterChange Server 4.3.0.
  - a. Create new WebSphere InterChange Server admin user.
  - b. Make user member of group mqm/dbadmins group.
  - c. Install to new directory with new WebSphere InterChange Server admin user.
  - d. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.
  - e. Modify custom scripts to reflect new IBM ORB settings.
  - f. Include external code and jar-files in CLASSPATH.
  - g. Modify and verify the environment.
  - h. Install the WebSphere Business Integration Toolset.




- c. Verify the relationship with Relationship Manager.

--

### Post-migration steps

Table A-24 details the post-migration steps for a without in-place database migration of WebSphere InterChange Server 4.2.1 on AIX.

Table A-24 AIX without in-place database post-migration steps for 4.2.1

1. Create basic security to match previous version.
  - a. Create user admin/password null in administrator role.
  - b. Enable Role-based Access Control in WebSphere InterChange Server.
  - c. Restart WebSphere InterChange Server for this to become active.
2. Reconfigure clients using the Server Access Interface.
3. Enable the WebSphere InterChange Server for adapters.
4. Test the WebSphere InterChange Server.
5. Uninstall (optional):
  - a. JDK 1.3.1 if not needed elsewhere
  - b. WebSphere InterChange Server 4.2.1
  - c. VisiBroker


# Migration checklist for WebSphere ICS 4.2.2

This section provides the steps required to migrate WebSphere InterChange Server 4.2.2.4 to WebSphere InterChange Server 4.3.0. It is divided into different sections for Windows and AIX and for in-place database migration and for without in-place database migration.

## Windows: in-place database migration

This section provides the steps necessary to perform an in-place database migration of WebSphere InterChange Server 4.2.2.4 to WebSphere InterChange Server 4.3.0 on Windows.

### Pre-migration steps

Table A-25 details the pre-migration steps for an in-place database migration of WebSphere InterChange Server 4.2.2 on Windows.

Table A-25 Windows in-place database pre-migration steps for 4.2.2

- 1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling.
  - b. Resolve transactional status of transactional collaborations.
  - c. Resubmit failed events or discard the events using Flow Manager.
  - d. Stop all collaborations.
- 2. Backing Up and Shutting Down Components.
  - a. Back up InterChange Server repository.
  - b. Back up Custom collaborations components.
  - c. Back up Custom adapters.
  - d. Back up Map components.
  - e. Back up Relationship database tables.
  - f. Back up Log files.
  - g. Back up user projects.
  - h. Shut down the WebSphere InterChange Server.
  - i. Back up the database.


- j. Stop PersistentNameServer.
- k. Back up the WebSphere MQ environment and message data.
- l. Stop WebSphere MQ.
- 3. Uninstallation.
  - a. Uninstall Windows services (if any are present for example: WebSphere Business Integration Adapters, WebSphere InterChange Server).
  - b. Uninstall Java Development Kit (JDK) if not used by other applications.
  - c. Uninstall WebSphere InterChange Server.


## Migration steps

Table A-26 details the migration steps for an in-place database migration for WebSphere InterChange Server 4.2.2 on Windows.

*Table A-26 Windows in-place database migration steps for 4.2.2*

- 1. Upgrade the Database System.
  - a. Upgrade database.
  - b. Check the database parameters.
  - c. For DB2 test the c-compiler.
  - d. Restart database.
- 2. Upgrade WebSphere MQ.
- 3. Installation.
  - a. Install WebSphere InterChange Server 4.3.
  - b. Configure the WebSphere InterChange Server with ICSCfg.bat.
  - c. Copy DLMS and collaboration directories to new installation.
  - d. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.
  - e. Include external code and jar files in CLASSPATH.


- f. Modify and verify the environment variables: CROSSWORLDS, MQ\_LIB, PATH.
4. Starting the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.
  - c. Start the Persistent Name Server.
  - d. Start the WebSphere InterChange Server in design mode.
5. Import the previous repository into the tooling.
6. Toolset Include custom .jar files in the CLASSPATH.
7. Set the Compiler compliance level to JDK 1.4.
8. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Examine and set the triggering events for LLBP Asynch Inbound calls to <None>.
  - c. Reconfigure any custom code or property that references a changed property.
9. Resolve any collaborations that are in transactional status.
10. Deploy modified components.
  - a. Redeploy all components to have all administrative features later.
  - b. (Optional) Select **Create Schema** option if the relationship location has changed or is not working.
  - c. Restart WebSphere InterChange Server if prompted.


### Post-migration steps

Table A-27 details the post-migration steps for an in-place database migration of WebSphere InterChange Server 4.2.2 on Windows.

Table A-27 Windows in-place database post-migration steps for 4.2.2

1. Security.

--

- a. Create user admin/null.
  - b. Assign Administrator role to user admin.
  - c. Enable Role-based Access Security in WebSphere InterChange Server.
2. Install Adapter Framework 2.6 and upgrade adapters.
3. Update any Server Access Interface clients.
4. Test the WebSphere InterChange Server.


## Windows: without in-place database migration

This section provides the steps necessary to perform a without in-place database migration for WebSphere InterChange Server 4.2.2.4 to WebSphere InterChange Server 4.3.0.

### Pre-migration steps

Table A-28 details the pre-migration steps for without in-place database migration for WebSphere InterChange Server 4.2.2 on Windows.

*Table A-28 Windows without in-place database pre-migration steps for 4.2.2*

1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling.
  - b. Resolve transactional status of transactional collaborations.
  - c. Resubmit failed events or discard the events using Flow Manager.
  - d. Stop all collaborations.
2. Backing Up and Shutting Down Components.
  - a. Back up InterChange Server repository using repos\_copy utility.
  - b. Back up Custom collaborations components.
  - c. Back up Custom adapters.
  - d. Back up Map components.
  - e. Back up Relationship database tables.


- f. Back up Log files.
- g. Back up user projects.
- h. Shut down the WebSphere InterChange Server.
- i. Back up the database.
- j. Stop PersistentNameServer.
- k. Back up the WebSphere MQ environment and message data.
- l. stop WebSphere MQ.
3. Uninstallation.
  - a. Uninstall Windows services (if any are present for example: WebSphere Business Integration Adapters, WebSphere InterChange Server).
  - b. Uninstall Java Development Kit (JDK) if not used by other applications.
  - c. Uninstall WebSphere InterChange Server.


## Migration steps

Table A-29 details the migration steps for without in-place database migration of WebSphere InterChange Server 4.2.2 on Windows.

*Table A-29 Windows without in-place database migration steps for 4.2.2*

1. Upgrade the Database System.
  - a. Upgrade database.
  - b. Create an empty database for the 4.3.0 repository.
  - c. Check the database parameters.
  - d. For DB2 test the c-compiler.
  - e. Restart database.
2. Upgrade WebSphere MQ.
3. Installation.
  - a. Install WebSphere InterChange Server 4.3.




- b. Configure the WebSphere InterChange Server with ICSCfg.bat.
  - c. Copy DLMS and collaboration directories to new installation.
  - d. Apply any custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.
  - e. Include external code and jar files in CLASSPATH.
  - f. Modify and verify the environment variables: CROSSWORLDS, MQ\_LIB, PATH.
4. Starting the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.
  - c. Start the Persistent Name Server.
  - d. Start the WebSphere InterChange Server in design mode.
5. Import the previous repository into the tooling.
6. Toolset Include custom .jar files in the CLASSPATH.
7. Set the Compiler compliance level to JDK 1.4.
8. Reconfigure any of the other components that require modification
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Examine and set the triggering events for LLBP Asynch Inbound calls to <None>.
  - c. Reconfigure any custom code or property that references a changed property.
9. Resolve any collaborations that are in transactional status.
10. Deploy modified components.
  - a. Redeploy all components to have all administrative features later.
  - b. Restart WebSphere InterChange Server if prompted.
11. Complete relationship migration.


- a. Copy the data from each relationship table to the new relationship table.
- b. Set the sequence number.
- c. Verify the relationship with Relationship Manager.


## Post-migration steps

Table A-30 details the post-migration steps for without in-place database migration of WebSphere InterChange Server 4.2.2 on Windows.

*Table A-30 Windows without in-place database post-migration steps for 4.2.2*

1. Security.
  - a. Create user admin/null.
  - b. Assign Administrator role to user admin.
  - c. Enable Role-based Access Control in WebSphere InterChange Server.
  - d. Restart WebSphere InterChange Server for this to become active.
2. Install Adapter Framework 2.6 and upgrade adapters.
3. Update any Server Access Interface clients.
4. Test the WebSphere InterChange Server.


## AIX: in-place database migration

This section provides the steps necessary to perform an in-place database migration of WebSphere InterChange Server 4.2.2.4 to WebSphere InterChange Server 4.3.0.

## Pre-migration steps

Table A-31 details the pre-migration steps for an in-place database migration for WebSphere InterChange Server 4.2.2 on AIX.

*Table A-31 AIX in-place database pre-migration steps for 4.2.2*

1. Put the environment in a quiescent state.
  - a. Stop all adapters from polling.
  - b. Resolve transactional status of transactional collaborations.


- c. Resubmit failed events or discard the events using Flow Manager.
  - d. Stop all collaborations.
- 2. Back up and shutting down components.
  - a. Back up WebSphere InterChange Server repository.
  - b. Back up custom collaborations components (Java class files, message files, Custom connectors).
  - c. Back up map components (map design files, Java class files, and message files).
  - d. Back up Relationship database tables.
  - e. Back up work-in-progress and transaction database tables.
  - f. Back up log files.
  - g. Stop Persistent Name Server.
  - h. Stop WebSphere InterChange Server.
  - i. Back up database.
  - j. Back up WebSphere MQ environment.
  - k. Stop WebSphere MQ.


## Migration steps

Table A-32 details the migration steps for an in-place database migration of WebSphere InterChange Server 4.2.2 on AIX.

*Table A-32 AIX in-place database migration steps for 4.2.2*

- 1. Upgrade the Database System.
  - a. Upgrade Database.
  - b. Check the database parameters.
  - c. For DB2 test the c-compiler.
  - d. For DB2 Catalog database with Loopback Node.
  - e. Restart database.
- 2. Upgrade WebSphere MQ.


3. Installation WebSphere InterChange Server.
  - a. Create new icsadmin user.
  - b. Add to group mqm/dbadmins.
  - c. Install to new directory.
  - d. Apply old settings to configuration from WebSphere InterChange Server 4.2.2.4.
  - e. Apply security settings as needed.
  - f. Copy DLMS and collaboration directories to new installation.
  - g. Apply custom configuration settings to \$CROSSWORLDS/InterchangeSystem.cfg.
  - h. Include external code and jar-files in CLASSPATH.
  - i. Modify and verify the environment variables.
  - j. Install the WebSphere Business Integration Toolset.
4. Starting the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.
  - c. Start the Persistent Name Server.
  - d. Start the WebSphere InterChange Server in design mode.
5. Import the previous repository into the tooling.
6. Toolset: Include custom .jar files in the CLASSPATH.
7. Setting the Compiler compliance level to JDK 1.4.
8. Reconfigure any of the other components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Examine and set the triggering events for LLBP Asynch Inbound calls to <None>.
  - c. Reconfigure any custom code or property that references a changed property.


9. Resolve any collaborations that are transactional status.
10. Deploy modified components.
  - a. (Optional) Redeploy all components to have all administrative features later.
  - b. Restart WebSphere InterChange Server if necessary.


## Post-migration steps

Table A-33 details the post-migration steps for an in-place database migration of WebSphere InterChange Server 4.2.2 on AIX.

*Table A-33 AIX in-place database post-migration steps for 4.2.2*

1. Security
  - a. Create user admin/password null in administrator role.
  - b. Enable Role-based Access Control for WebSphere InterChange Server.
  - c. Restart WebSphere InterChange Server for this to become active.
2. Install Adapter Framework 2.6 and upgrade adapters.
3. Modify Server Access Interface client settings.
4. Enable WebSphere InterChange Server for Adapters.
5. Test the WebSphere InterChange Server.
6. Uninstall (optional).
  - a. JDK 1.3.1. (if not needed by other applications).
  - b. WebSphere InterChange Server 4.2.2.4.


# AIX: without in-place database migration

This section provides the steps necessary to perform a without in-place database migration for WebSphere InterChange Server 4.2.2.4 to WebSphere InterChange Server 4.3.0.

## Pre-migration steps

Table A-34 details the pre-migration steps for a without in-place database migration of WebSphere InterChange Server 4.2.2 on AIX.

Table A-34 AIX without in-place database pre-migration steps for 4.2.2

1. Put the environment in a quiescent state.

a. Stop all adapters from polling the event tables.

b. Resolve transactional status of transactional collaborations.

c. Resubmit failed events or discard the events.

d. Stop all collaborations.
2. Backing Up and Shutting Down Components.

a. Back up WebSphere InterChange Server repository.

b. Back up Custom collaborations components (Java class files, message files, Custom Connectors)

c. Back up Map components (map design files, Java class files, message files).

d. Back up Relationship database tables.

e. Log files (as desired for historical information).

f. Back up user projects.

g. Shut down the WebSphere InterChange Server.

h. Back up the database.

i. Stop PersistentNameServer.

j. Back up the WebSphere MQ environment.


## Migration steps

Table A-35 details the migration steps for without in-place database migration of WebSphere InterChange Server 4.2.2 on AIX.

*Table A-35 AIX without in-place database migration steps for 4.2.2*

1. Upgrade the Database System.
  - a. Upgrade database.
  - b. Prepare an empty database for the 4.3.0 repository.
  - c. Check the database parameters.
  - d. For DB2 test the c-compiler.
  - e. For DB2 Catalog database with Loopback Node.
  - f. Restart database.
2. Upgrade WebSphere MQ.
3. Installation of WebSphere InterChange Server 4.3.0.
  - a. create new WebSphere InterChange Server admin user.
  - b. make user member of group mqm/dbadmins group.
  - c. install to new directory with new WebSphere InterChange Server admin user.
  - d. Apply custom configuration settings to `$CROSSWORLDS/InterchangeSystem.cfg`.
  - e. Include external code and jar-files in CLASSPATH.
  - f. Modify and verify the environment.
  - g. Install the WebSphere Business Integration Toolset.
4. Starting the new Server.
  - a. Start WebSphere MQ Queue Manager and Listener.
  - b. Start Database.
  - c. Start Persistent Name Server.
  - d. Start WebSphere InterChange Server in design mode.
5. Start the System Manager.

[illegible]

6. Import the Solution created in the backup step.
7. Include custom .jar files in the CLASSPATH.
8. Set the compliance of the compiler to Java 1.4.
9. Examine custom code for Java 1.4 compliance.
10. Compile maps and templates in tools.
11. Reconfigure components that require modification.
  - a. Reconfigure or create any connection pools and relationship definitions.
  - b. Examine and set the triggering events for LLBP Asynch Inbound calls to <None>.
  - c. Reconfigure any custom code or property that has changed.
12. Deploy the user project.
13. Restart the WebSphere InterChange Server if prompted.
14. Complete relationship migration.
  - a. Copy the data from each relationship table to the new relationship table.
  - b. Set the sequence number.
  - c. Verify the relationship with Relationship Manager.


### Post-migration steps

Table A-36 details the post-migration steps for without in-place database migration of WebSphere InterChange Server 4.2.2 on AIX.

*Table A-36 AIX without in-place database post-migration steps for 4.2.2*

1. Create basic security to match previous version.
  - a. Create user admin/password null in administrator role.
  - b. Enable Role-based Access Control in WebSphere InterChange Server.
  - c. Restart WebSphere InterChange Server for this to become active.
2. Reconfigure clients using the Server Access Interface.




3. Enable the WebSphere InterChange Server for adapters.
4. Test the WebSphere InterChange Server.
5. Uninstall (optional):
  - a. JDK 1.3.1 if not needed elsewhere
  - b. WebSphere InterChange Server 4.2.2.4


Archived



## Sample code

This appendix provides the code that was used by the team to overcome a problem when migrating WebSphere InterChange Server 4.1.1.9 to WebSphere InterChange Server (WebSphere ICS) 4.3, as documented in Chapter 8, “Migration procedures for WebSphere InterChange Server V4.1.1.9 to V4.3” on page 119.

## DDL code for WebSphere InterChange Server 4.1.1.9

At the time we wrote this book, a product defect was causing the WebSphere InterChange Server to hang during the first startup. This defect is applicable for WebSphere InterChange Server 4.1.1 only when using DB2 as the repository database. This defect has been reported and assigned tracking number CR60364. Check the support site to see if a fix for this problem number is available:

<http://www.ibm.com/software/integration/wbiserver/ics/support/>

The team created a DDL script to circumvent the problem. This workaround manually creates five database tables and indices. Use the DB2 Command Center or command line processor to load the DDL script as shown below. Change the database name, user, and password to match the environment. This workaround is reviewed and approved by the WebSphere InterChange Server Development team.

```
connect to ICS411 user cwadm411 ;

INSERT INTO CXREPOSVERSIONS (NAME, VERSION) VALUES ('CxCompStateStatsConfig',
'1.0.0');
INSERT INTO CXREPOSVERSIONS (NAME, VERSION) VALUES
('CxPersistentMonitorCollabs', '1.0.0');
INSERT INTO CXREPOSVERSIONS (NAME, VERSION) VALUES
('CxPersistentMonitorConnectors', '1.0.0');
INSERT INTO CXREPOSVERSIONS (NAME, VERSION) VALUES
('CxPersistentMonitorServer', '1.0.0');
INSERT INTO CXREPOSVERSIONS (NAME, VERSION) VALUES ('CxStateLog', '1.0.0');

CREATE TABLE "CXCOMPSTATESTATSCONFIG" (
    "COMPONENTTYPE" INTEGER ,
    "COMPONENTNAME" VARCHAR(80) ,
    "COMPONENTVERSION" VARCHAR(30) ,
    "SNAPSHOTFREQ" INTEGER ,
    "STATEMONITOR" VARCHAR(30) );

CREATE UNIQUE INDEX "COMPSTATESTATSIDX" ON "CXCOMPSTATESTATSCONFIG"
("COMPONENTTYPE", "COMPONENTNAME") CLUSTER ALLOW REVERSE SCANS;

CREATE TABLE "CXPERSISTENTMONITORCOLLABS" (
    "STATTIMESTAMP" TIMESTAMP ,
    "COMPONENTNAME" VARCHAR(80) ,
    "COMPONENTVERSION" VARCHAR(30) ,
    "FAILEDCALLS" INTEGER ,
    "TOTALCALLS" INTEGER ,
```

```

        "FAILEDEVENTS" INTEGER ,
        "TOTALEVENTS" INTEGER ,
        "CURRENTEVENTS" INTEGER );

CREATE INDEX "PERMONCOLLABIDX" ON "CXPERSISTENTMONITORCOLLABS"
("COMPONENTNAME") CLUSTER ALLOW REVERSE SCANS;

CREATE TABLE "CXPERSISTENTMONITORCONNECTORS" (
    "STATTIMESTAMP" TIMESTAMP ,
    "COMPONENTNAME" VARCHAR(80) ,
    "COMPONENTVERSION" VARCHAR(30) ,
    "TOTALBUSOBJSENT" INTEGER ,
    "TOTALBUSOBJRCVD" INTEGER );

CREATE INDEX "PERMONCONNIDX" ON "CXPERSISTENTMONITORCONNECTORS"
("COMPONENTNAME") CLUSTER ALLOW REVERSE SCANS;

CREATE TABLE "CXPERSISTENTMONITORSERVER" (
    "STATTIMESTAMP" TIMESTAMP ,
    "COMPONENTNAME" VARCHAR(80) ,
    "COMPONENTVERSION" VARCHAR(30) ,
    "FAILEDCALLS" INTEGER ,
    "TOTALCALLS" INTEGER ,
    "FAILEDEVENTS" INTEGER ,
    "TOTALEVENTS" INTEGER ) ;

CREATE INDEX "PERMONSVRIDX" ON "CXPERSISTENTMONITORSERVER" ("COMPONENTNAME")
CLUSTER ALLOW REVERSE SCANS;

CREATE TABLE "CXSTATELOG" (
    "STATTIMESTAMP" TIMESTAMP ,
    "COMPONENTTYPE" INTEGER ,
    "COMPONENTNAME" VARCHAR(80) ,
    "COMPONENTVERSION" VARCHAR(30) ,
    "ACTIVITY" VARCHAR(255) );

CREATE INDEX "STATELOGIDX" ON "CXSTATELOG" ("COMPONENTTYPE", "COMPONENTNAME")
CLUSTER ALLOW REVERSE SCANS;

commit;
disconnect current;

```



## Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

### Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG246475>

Alternatively, you can go to the IBM Redbooks Web site at:

<http://www.ibm.com/redbooks>

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246475.

### Using the Web material

The additional Web material that accompanies this redbook includes:

<i>File name</i>	<i>Description</i>
<b>411_430_DLL.zip</b>	Zipped DLL Code for WebSphere InterChange Server

## How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.



# Abbreviations and acronyms

<b>API</b>	Application Programmable Interface
<b>BO</b>	Business Objects
<b>CORBA</b>	Common Object Request Broker Architecture
<b>CSM</b>	CrossWorlds System Monitor
<b>EIS</b>	Enterprise Information Systems
<b>IBM</b>	International Business Machines Corporation
<b>ICL</b>	integration component library
<b>ICS</b>	InterChange Server
<b>IDL</b>	Interface Definition Language
<b>IOR</b>	interoperable object reference
<b>ITE</b>	Integrated Test Environment
<b>ITLM</b>	IBM Tivoli License Manager
<b>ITSO</b>	International Technical Support Organization
<b>J2EE</b>	Java 2 Platform, Standard Edition
<b>JDK</b>	Java Development Kit
<b>JMS</b>	Java Message Service (JMS)
<b>JRE</b>	Java Runtime Environment
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>OAD</b>	Object Activation Daemon
<b>ORB</b>	Object Request Broker
<b>SLES</b>	SUSE LINUX Enterprise Server (SLES)
<b>SLSS</b>	SUSE LINUX Standard Server (SLES)



# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 474. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Implementing and Administering WebSphere Business Integration Server V4.2.2*, SG24-7006
- ▶ *WebSphere Business Integration for SAP®*, SG24-6354
- ▶ *Using Web Services for Business Integration*, SG24-6583

## Online resources

These Web sites and URLs are also relevant as further information sources.

- ▶ WebSphere Information Center documentation:

<http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp>

Here are the paths to the PDFs. All references are to Version 4.3.0.

An online version is accessible through the left navigation bar on this page. There are no page numbers in the online version, but you can navigate by chapter and section.

- *IBM WebSphere InterChange Server System Installation Guide for UNIX*  
**Interchange Server and Toolset → Installation → PDFs → UNIX**
- *IBM WebSphere InterChange Server System Installation Guide for Windows*  
**Interchange Server and Toolset → Installation → PDFs → Windows**
- *WebSphere InterChange Server System Administration Guide*  
**Interchange Server and Toolset → Administration → PDFs → System administration**
- Role-based security in this chapter:

**Administering components of the system → Administering role-based access control (RBAC)**

– *IBM WebSphere InterChange Server Access Development Guide*

**Interchange Server and Toolset → Development → PDFs → Access development**

– *Implementation Guide for WebSphere InterChange Server*

**Interchange Server and Toolset → Development → PDFs → System implementation**

► A large repository of WebSphere MQ product documents is available at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

In the **WebSphere MQ Messaging** menu, select **latest multiplatform books**.

The *System Administration Guide* contains the appropriate information. For this book, the WebSphere MQ version that is required is 5.3.

► *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*

<http://www.ibm.com/developerworks/java/jdk/diagnosis/>

## 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](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## A

- Access Client 75, 116, 170, 270, 366
  - connectivity 74
  - required code changes 75
- activate security 192, 292, 386
- Activity Editor 26, 96–97
- Adapter Framework 102, 141, 169, 192, 219, 245, 269, 341, 365, 387, 413
- adapters 18–19, 102
  - configuring for remote restart capability 81
  - remote restart capability 85
  - upgrading issues 38, 41
- administrative features 45
- administrative messages (Admin) 22
- agent-activation queue 77–78
- AIX
  - WebSphere MQ environment settings 85
- application adapters 18
- application queue 77
- application-specific business object 19, 115
- architecture and software dependency changes 49
- asymmetric security 21
- Asynch Inbound call 240, 261, 286, 309, 336, 357, 381, 404
- auditing 24
- authentication 23
- availability features 44

## B

- back up and shut down components 144, 196, 249, 296, 345, 390
- backing up the database 122, 145, 197, 250, 346
- backing up the product directory 122, 145, 250, 346
- backing up the repository 122, 145, 173, 196, 249, 296, 346, 391
- backing up the system 323
- backing up WebSphere MQ environment 122, 145, 250, 346
- backup 333, 354
  - pre-migration 121, 172, 225–226, 272–274, 297, 323, 368–370, 390, 392
  - with repos\_copy 324
- bidirectional language support 26

- Business Integration System 90
- business logic 117
- Business Object Designer 17, 92
- business object instance 95
- Business Object Probe 94–95
- business objects 18–19, 22, 24–25, 116, 132, 157, 183, 208
  - transformed by maps 19
- business process template 19

## C

- C-compiler 146, 174, 198, 251, 275, 298, 347, 370, 393
- change progression in supported software stack 57
- Code Editor 96
- Code Generation 97
- collaboration 23, 92–93, 114, 116, 135, 186, 225, 272, 286, 295, 323, 343, 345, 368, 380–381, 388
- collaboration debugger tool 96
- collaboration debugging 96
- collaboration directories 130, 153, 180, 204, 235, 281, 332, 376
- collaboration groups 18, 115
- collaboration objects 18, 24, 115, 121, 144, 172
  - and maps 19
- collaboration statistics 121, 144, 196, 248, 390
- collaboration template 16, 18, 24, 32, 96, 132, 134, 157, 159, 183, 185, 208, 210, 239–240, 260–261, 285–286, 308–309, 335–336, 356–357, 381, 403–404
- collaboration template development 17
- collaborations 18, 25–26, 32, 116, 136, 144, 153, 172, 180, 186–187, 195–196, 204, 224–225, 235, 240, 248, 257, 272, 281, 286, 295, 304, 322–323, 332, 336, 344, 353, 368, 376, 381, 390, 420, 424, 431, 438–439, 443, 445–446, 450, 452–453, 456
  - in quiescent state 120
- collaborations components 121, 145, 172, 196, 225, 249, 272, 296, 345, 368, 391
- Common Object Request Broker Architecture (CORBA) object 69
- complete repository 120
- component definitions 120, 171, 224, 271, 322, 367

- component state changes 118
- components
  - development 32
- configuration file changes 91
- configuration files
  - updated to XML 91
- connection pool 111, 114–115, 128, 132, 135, 151, 154, 157, 160, 178, 183, 186, 202, 206, 208, 211, 233, 239, 255, 261, 279, 286, 302, 309, 330, 336, 351, 357, 374, 381, 397
- connection resilience 25
- Connector Configurator 17, 25, 27, 92, 121, 135, 144, 160, 172, 186, 195, 211, 345
- connector definition 135, 160, 186, 211
- Connector Development Kit 32
- connector script file 79
- connectors 24, 32, 132, 157, 183, 208
- Control Center 161–162, 164, 212, 214–215, 261, 263, 310, 312–313, 358, 405, 407, 409
- CORBA
  - non-IBM CORBA 75
- CORBA object 69
- CORBA object references 71
- CORBA objects 70
- CORBA references 70
- CORBA section 73
- CosNameServer\_Dump 71
- CosNaming service 57
- creating the project plan for migrating 60
- CrossWorlds System Manager 92, 121, 144, 172
- CrossWorlds System Monitor 196
- custom code changes 75
- custom collaboration components 323
- custom connectors 121, 145, 172, 196, 225, 249, 272, 296, 323, 345, 368, 391

## D

- data handlers 102
- database configuration 128, 151, 178, 202, 233, 255, 279, 302, 330, 351, 374, 397
- database management system 54
- database migration tools
  - database instance migrate 54
  - database migrate 54
- database, supported versions 32, 38, 41
- databases supported by WebSphere ICS 50
- development features 48
- dynamic and static relationships 117

- dynamic data 122, 145, 173, 196, 226, 249, 272–273, 296, 323, 345, 368–369, 391
- dynamic naming service 57
- dynamic relationships 136, 161, 187, 211, 241, 287, 337, 382, 404
- Dynamic Service Calls 25

## E

- each migration task 60
- Eclipse 91–92
  - plug-ins 92
  - projects 92
  - workbench 92
  - workspace 92
- Eclipse framework
  - source code management 92
- end-to-end security 21–22
- enhancements and changes 33, 39
- Enterprise Information Systems 18
- environment variables 131, 154, 181, 236, 257, 282, 332, 353, 377
  - modifying during installation 122, 146, 250, 347
- Export Solution 227, 237, 250, 259, 273, 283, 306, 324, 333, 346, 354, 369, 378, 401
- Export Solution utility 297, 392

## F

- Fail Events Management API 25
- Failed Event Manager 17, 25, 95
- failed events 121, 144, 196, 225, 248, 272, 295, 323, 345, 368, 390
- failed flow 193, 293, 388
- Failed Flows 343, 388–389
- Failed flows 272, 368, 390
- failed flows 120, 171, 224, 271, 322, 367
- faults 386
- Flow Manager 17, 95, 116, 135, 186, 195, 225, 248, 272, 286, 295, 323, 345, 368, 381, 390

## G

- generic business objects 19

## H

- hardware requirements 55–56
- historical data 118
- historical system information 118

## I

IBM 55  
IBM Business Integration Reference Architecture 12–15  
    business integration functions 12–15  
    implementation acceleration tools 14  
    integration services 14  
    integration with other applications and data 13  
    management tools 14  
    more references 12  
    tools for modeling processes 13  
    transparency of resources 13  
IBM CrossWorlds Full Toolkit 89  
IBM CrossWorlds InterChange Server 177, 201, 221, 319, 415  
    tooling 51  
IBM ORB 30, 37, 56, 68, 130, 153, 180, 204, 235, 257, 281, 304  
    and Server Access Interface 75  
    changing properties 72  
    configuration for Access Client connectivity 74  
    configuring 72–76  
    configuring debug properties 74  
    ORB communication limitation 70  
    setup and configuration 70–75  
IBM Persistent Name Server 70  
IBM Tivoli License Manager 27  
IBM Transient Naming Server 30, 37  
IBM WebSphere Application Server  
    Dynamic Service Calls 25  
IBM WebSphere Business Integration 69  
IBM WebSphere Business Integration Adapters 18  
    applications adapters 18  
    technology adapters 18  
IBM WebSphere Business Integration collaboration 19  
IBM WebSphere Business Integration Toolset 89  
ICS components 132  
IDL  
Import 359, 407  
Import Solution 309  
incompleted flows 117  
individual business object 22  
initiation queue 78  
in-place database migration 54, 117, 171, 271, 421, 445  
    V4.1.1 checklist  
        AIX 428–431  
        Windows 420–423

    V4.1.1.9 to V4.3.0 on AIX 171–194  
    V4.1.1.9 to V4.3.0 on Windows 120–143  
V4.2.1 checklist  
    AIX 443–446  
    Windows 436  
V4.2.1 to V4.3 on AIX 271–294  
V4.2.1 to V4.3 on Windows 224–247  
V4.2.2 checklist  
    AIX 456–459  
    Windows 450–453  
V4.2.2 to V4.3 on AIX 367–389  
V4.2.2 to V4.3 on Windows 322–343  
Inprise VisiBroker 56  
in-process 121, 144, 172, 196, 225, 248, 272, 295, 323, 345, 368, 390  
install process 175, 275, 371  
installation 377  
Integrated Test Environment 90, 92, 95–96  
    collaboration viewer 96  
    remote service support 26  
integration component library 32, 90, 92–93, 132, 134, 156, 183, 208, 210, 237, 239–240, 259–260, 283, 306, 333, 335–336, 354, 356, 369, 378, 401  
    Export Solution utility 324  
    Export Solution utility fix 227, 250, 273, 297, 346, 392  
integration components 90  
integration solution 90  
Integrity (security level) 22  
Integrity Plus Privacy (security level) 22  
Interface Definition Language  
    See IDL  
Internet Inter-ORB Protocol 69  
Inter-ORB Reference 74  
introduction to the new development model 89  
ITE test unit 96  
ITLM  
    See IBM Tivoli License Manager

## J

Java class files 121–122, 145, 172, 196, 225, 249, 272, 296, 323, 345, 368, 391  
Java code development considerations 9  
Java Runtime Environment  
    See JRE  
JDK 21, 31, 38, 41, 124, 131, 147, 154, 229, 236, 239, 252, 258, 333, 335, 353  
JDK compliance 134

JMS delivery transport 135, 160, 186, 211  
JMS transport optimization 25  
JRE 21, 31, 38, 41, 70, 130, 136, 187, 240, 286,  
336, 381

## L

large-object support 24  
LDAP 129, 152, 179, 203, 234, 256, 280, 303, 331,  
352, 375, 398  
LDAP server 100, 234, 331  
Lightweight Directory Access Protocol  
    See LDAP  
log files 122, 145, 173, 196, 226, 249, 273, 296,  
323, 345, 369, 391  
Log Viewer 17  
Long Lived Business Process 114  
Long Lived Business Process Asynch Inbound call  
240, 261, 286, 309, 336, 357, 381, 404

## M

map 187, 286, 381  
map components 122, 145, 172, 196, 225, 249,  
272, 296, 323, 345, 368, 391  
map definition 32  
map design files 122, 145, 172, 196, 225, 249, 272,  
296, 323, 345, 368, 391  
Map Designer 17, 26–27, 92  
maps 19, 24, 26, 32, 132, 136, 157, 159, 183, 185,  
187, 208, 210, 240, 260, 285–286, 308, 336, 356,  
380–381, 403  
memory fragmentation 24  
message files 121, 145, 172, 196, 225, 249, 272,  
296, 323, 345, 368, 391  
migration  
    after migration 137, 164, 187, 193, 241, 248,  
    265, 337, 361, 382, 388, 390  
    before migration 120, 137, 143, 171, 188, 195,  
    224, 248, 271, 287, 294, 322, 344, 367, 382  
    checklists 61–63  
    configuration values used during migration 126,  
    149, 253, 277, 349, 372, 395  
    creating the project plan 60  
    during migration 176, 200, 231, 300, 328  
    expected time frame 60  
    hardware considerations 55–56  
    human roles 58–60  
    in-place database migration 54  
    methods 54–55

    minimum hardware requirements 55  
    planning 57  
    reasons for migrating 8  
    recommended hardware requirements 56  
    required steps 146, 174, 197, 251, 274, 297,  
    347, 370, 392  
    software considerations 56–58  
    stages 60–63  
    time needed 60–61  
    using the old repository 54  
    without in-place database migration 54–55  
migration checklist  
    V4.1.1  
        AIX 428–435  
        Windows 420–428  
    V4.2.1  
        AIX 443–449  
        Windows 436–442  
    V4.2.2  
        AIX 456–463  
        Windows 450  
migration environment 117  
migration planning  
    tasks to consider 60  
migration process 142, 170, 193, 220, 258, 270,  
292, 318, 354, 366, 387, 414  
migration requirements  
    additional hardware and machines 56  
    CPU capacity 55  
    hard disk drive space 56  
    hard disk storage space 55, 100  
    memory 55–56  
    processor 55–56  
migration scenario 130, 153, 180, 204, 235, 257,  
281, 304, 332, 353, 376, 399  
MQ Delivery Transport 228, 326, 371

## N

Name Server 70  
naming content 71  
Naming Conventions 144, 171, 195, 248, 271, 295,  
344, 367, 390  
naming repository 70–71  
naming server 71  
naming service 69–71  
National Language Support 47, 89  
new features 33, 37, 39  
new security 164, 187, 265, 361, 382



non-IBM CORBA 75

## O

OAPort property 74  
Object Activation Daemon 57, 78  
    replaced in IBM ORB 77  
object definitions 92  
Object Request Broker  
    See ORB  
objects 132, 183, 208  
object-to-object interactions 69  
operating systems supported by WebSphere ICS 49  
ORB 30, 37, 69, 116, 181, 205, 229, 282, 305, 377, 400  
osagent 57  
osagents 70

## P

performance features 46  
Persistent Name Server 154, 258, 354  
Persistent Monitoring 95  
Persistent Name Server 26, 70–71, 181–182, 205–206, 282–283, 305–306, 325, 346, 370, 377–378, 400  
Persistent Naming Server 70  
    starting 71  
    startup file 71  
Poll Frequency 121, 144, 172, 195, 225, 248, 272, 295, 323, 345, 368, 390  
polymorphic map 115  
polymorphic mechanism 115  
post-migration steps 215, 313, 409  
predefined roles 23  
pre-migration step 121, 172, 272, 295, 323, 345, 368  
pre-migration, migration, and post-migration 61  
pre-planning 61  
prerequisite software 50  
Privacy (security level) 22  
process definition 77, 84  
Process Designer 17, 19, 92  
process-definition object 77  
product history of new features 43  
projects 92

## Q

quiescent state 144, 195, 248, 295, 344, 390

## R

rebranding support 47  
recommended hardware configuration 56  
Red Hat Enterprise Linux 28  
Redbooks  
    related 12  
Redbooks Web site 474  
    Contact us xvi  
RELAT databases 114  
Relationship database tables 323  
relationship database tables 122, 145, 173, 196, 226, 249, 273, 296, 345, 369, 391  
relationship definitions 135, 155–156, 160, 186, 206, 208, 211, 239, 261, 286, 309, 336, 357, 381  
Relationship Designer 92  
Relationship Manager 17, 164, 215, 265, 313, 361, 409  
relationship tables 117, 206, 208, 263  
relationships 24, 120, 128, 132, 151, 157, 161, 163, 171, 178, 183, 187, 202, 208, 211, 214, 224, 233, 255, 264, 271, 279, 287, 302, 312, 322, 330, 351, 360, 367, 374, 382, 397, 404, 408  
remote restart capability 81  
remote service support 26  
remote test server 26  
repos\_copy 54–55  
repository 129, 152, 179, 203, 234, 256, 280, 303, 331, 352, 375, 398  
repository content 128, 143, 151, 178, 195, 202, 233, 247, 255, 279, 302, 330, 344, 351, 374, 397  
repository database 131, 156, 182, 208  
repository migration 56  
role-based access 32, 38, 41  
role-based access control 22  
role-based security 95

## S

scalability features 46  
schedule 116  
schema information 122, 145, 197, 250, 346  
security 137, 141, 164, 188, 215, 219, 241, 245, 287, 313, 317, 337, 382, 409, 413  
security configuration 129, 152, 179, 203, 234, 256, 280, 303, 331, 352, 375, 398  
Security database 129, 152, 179, 203, 234, 256,

- 280, 303, 331, 352, 375, 398
- security features 43
- security features (new) 33, 38, 42
- security levels 21
- Server Access Interface 57, 135, 142, 160, 170, 186, 193, 211, 220, 240, 246, 261, 270, 286, 292, 309, 317, 342, 357, 366, 387, 414
- Server Access Interface API 116
- serviceability enhancements 45
- setting up on Windows 79
- static and dynamic relationships 161, 212, 261, 310, 358, 405
- static data 121, 145, 172, 196, 225, 249, 272, 296, 323, 345, 368, 391
- static naming service 57
- stored procedures 122–123, 136, 145–146, 156, 161, 173–174, 187, 197–198, 208, 211, 227–228, 241, 250–251, 274–275, 287, 297–298, 325–326, 337, 346–347, 370, 382, 392–393, 404
- successful migration 61
- SUSE LINUX Enterprise Server (SLES) 28
- SUSE LINUX Enterprise Server (SLSS) 8 49
- SUSE LINUX Standard Server (SLES) 8.1 49
- SUSE LINUX Standard Server (SLSS) 28
- system down time 60
- System Manager 16, 19, 26, 32, 74, 85, 92–93, 132, 135–136, 157, 160, 164, 183, 186–188, 208, 211, 215, 225, 227, 236, 240, 248, 250, 258, 265, 273–274, 287, 296–297, 306, 313, 323–324, 333, 336, 345–346, 354, 361, 368–369, 381–382, 390–392, 401, 409
  - compared to System Monitor 93
  - Import Components wizard 132
  - memory errors 132
  - Windows-based 93
- System Monitor 17, 82, 89, 93–95, 196, 229, 326–327
  - database capacity planning 95
  - key-component management 93
  - prerequisites 94
  - reporting on Business Object Probes 95
  - statistics 93
- system recovery 61

## T

- technology adapters 18
- template class files 187, 286, 381
- Template Definitions 240, 261, 286, 336, 357, 381

- templates 136, 159, 187, 210, 240, 260, 285–286, 308, 336, 356, 380–381, 403
- test environment 90
- test scenario 142, 170, 193, 220, 246, 270, 342, 366, 387, 414
- testcase 116
- testing 142, 170, 193, 220, 246, 270, 318, 342, 366, 387, 414
- third-party software
  - dependencies 56
  - requirements 50
- time considerations 60
- Tivoli License Manager 27
- tnsnameserver 57
- Tomcat
  - Failed Event Manager tool 25
- toolset 93
- transaction database table 122, 145, 173, 196, 226, 249, 273, 296, 345, 369, 391
- transaction database tables 323
- transactional collaboration 116
- transactional collaborations 121, 144, 172, 195, 225, 248, 272, 295, 323, 345, 368, 390
- transactional status 121, 135, 144, 172, 186, 195, 225, 248, 272, 286, 295, 323, 345, 368, 381, 390
- Transient Name Server 71
- Transient Naming Server 30, 37, 69, 71
  - settings 72
- transient naming server (tnsnameserver) 57
- transmission queue 78
- trigger event 78
- trigger message 78
- trigger monitor 77–78, 81–82, 85, 87
- triggering 77–88
  - process-definition object 77
  - setting up 78
  - troubleshooting on WebSphere MQ 87
  - understanding triggering 77
- triggering event 96
- triggering events 261, 286, 309, 357, 381, 404
- troubleshooting
  - triggering 87
  - triggering on WebSphere MQ 87

## U

- uninstall procedures 229, 326
- uninstall the previous version 221
- unresolved dependencies 240, 336

- Unresolved Flows 144
- usability enhancements 45
- User Project 160, 187, 211, 240, 261, 287, 309, 336, 357, 382
  - defined 90
  - deploying 136
- user project 90
  - deploying 187, 211, 287, 382, 404
  - exporting 226, 250, 324, 346
  - Import Solution 309, 357
  - populating 135, 160, 186, 211, 240
- User Projects 273, 296, 369, 391
- User Registry 129, 152, 179, 203, 234, 256, 280, 303, 331, 352, 375, 398
- user registry
  - for storing users and passwords 23

## V

- VisiBroker 57, 69–70, 72, 75, 122, 130, 145, 153, 180, 194, 197, 204, 221, 227, 229, 235, 257, 281, 294, 297, 304, 319
  - uninstalling 147
- Visibroker Agent 173
- VisiBroker naming service (osagent) 57
- VisiBroker ORB 116, 124
  - replaced by IBM ORB 30, 37
- Visual Test Connector 141, 169, 192, 220, 245, 269, 292, 317, 341, 365, 387, 413

## W

- Web servers, supported 94
- Web services 26
- Web-based System Monitor 93, 118
- Webbased System Monitor 118
- WebSphere Application Server 25, 76, 229, 326
- WebSphere Business Integration 184
  - support Web site 20
- WebSphere Business Integration Adapter 141, 169, 192, 219, 245, 269, 317, 341, 365, 387, 413
- WebSphere Business Integration Adapter Development Kit 32
- WebSphere Business Integration Adapter Framework 32, 38, 41
- WebSphere Business Integration Adapters 17, 102, 124, 141, 169, 192, 219, 229, 245, 269, 317, 326, 341, 365, 387, 413
  - uninstalling 147, 252, 348
- WebSphere Business Integration Message Broker

- V5.0.1 16
- WebSphere Business Integration Server
  - about 12
  - components 15–18
  - more references 12
- WebSphere Business Integration System 89
- WebSphere Business Integration Toolset 16–17, 20, 32, 69, 90, 133, 181, 205, 225, 238, 248, 282, 305, 323, 334, 345, 377, 400
  - Activity Editor 26
  - Connector Configurator 27
  - Java source parsing 26
  - JRE and JDK 21
  - Map Designer 27
  - updates in V4.3 26
- WebSphere InterChange Server
  - administrative features 45
  - architecture & software dependency changes 49
  - availability features 44
  - Business Integration Toolset 20
  - business objects 19
  - changes since V4.1.1 30, 33
  - changes since V4.2.2 41
  - collaboration object 19
  - collaborations 18–19
  - components 18–20
  - components in V4.3 15, 101
  - Configuration wizard 125, 147, 175, 199, 252, 348
  - connection resilience with databases 25
  - development features 48
  - development guidelines 9
  - development model 89–91
  - Fail Events Management API 25
  - Failed Event Manager 25
  - installation on UNIX (AIX) 104–108
  - installation on Windows 102–104
  - integration components 90
  - JMS transport 25
  - large-object support 24
  - major changes in 4.3 102
  - maps 19
  - migrating repository content 132, 236, 333
  - National Language Support features 47
  - new features in 4.3.0 20–28
  - performance features 46
  - prerequisite software 50
  - Process Designer 19

- product history 43
- quiescent state 144, 195, 248, 295, 344, 390
- reasons for migrating 8
- rebranding and renaming support 47
- remote 26
- scalability features 46
- security features 43
- security levels 21
- serviceability enhancements 45
- setting up new security 137, 241, 337
- support for Intel 32-bit architectures 28
- supported databases 50
- supported operating systems 49
- testing triggering 86
- third-party software requirements 50
- tooling changes 88–97
- toolset 19
- uninstall procedures 147, 252, 348
- usability enhancements 45
- WebSphere InterChange Server 4.1.1
  - changes since Version 4.1.1 30
  - checklist for migrating on AIX 428–435
  - checklist for migrating on Windows 420–428
- WebSphere InterChange Server 4.2.1
  - changes since Version 4.2.1 37
  - checklist for migrating on AIX 443–449
  - checklist for migrating on Windows 436–442
- WebSphere InterChange Server 4.2.2
  - changes since Version 4.2.2 41
  - checklist for migrating on AIX 456–463
  - checklist for migrating on Windows 450–456
- WebSphere InterChange Server instance 90, 93
- WebSphere InterChange Server repository 23, 121, 131–132, 145, 172, 183, 196, 225–226, 249, 258, 272, 296, 324, 345, 354, 368, 391
  - backing up 173
  - backing up with repos\_copy 122, 145, 173, 196, 249, 296, 346, 391
  - migrating content 132, 156, 207, 236, 283, 306, 333, 378, 401
- WebSphere InterChange Server repository tables 154, 206, 236, 258, 305, 333, 354, 400
- WebSphere MQ 24, 31, 37, 55, 83, 103, 107, 122, 131, 145, 154, 173–175, 177, 182, 201, 206, 227, 236, 250, 258, 274–275, 278, 283, 297–298, 323, 325, 333, 346, 353–354, 370–371, 377, 392–393, 396, 400
  - environment settings for AIX 85
  - upgrading 124, 147, 198, 228, 251, 326, 348
- WebSphere MQ configuration 127, 232, 254, 329, 350, 373
- WebSphere MQ Delivery Transport 124, 174, 275
- WebSphere MQ environment 274, 370
- WebSphere MQ message data 122, 145, 173, 196, 226, 249, 273, 296, 345, 369, 391
- WebSphere MQ objects 84
- WebSphere MQ product documentation 88
- WebSphere MQ trigger monitor 79, 84
- WebSphere MQ trigger monitor daemon 84
- WebSphere MQ triggering 57, 77, 82–83, 87–88
  - batch file for Windows 79
  - confirming configuration 85
  - confirming installation 81
  - connector script file for Windows 79
  - definitions 77
  - on UNIX 84
  - script file for UNIX 83
  - setting up 78
  - setting up on UNIX 83–85
  - setting up on Windows 79–82
  - testing configuration 86
- WebSphere MQ triggering .bat file 80
- WebSphere MQ Workflow V3.5 16
- WebSphere Studio Application Developer Integrated Edition 96
- WebSphere Studio Application Developer Integration Edition 91
- WebSphere Studio Workbench 91, 96
- Windows-based System Monitor 295
- without in-place database migration 54–55, 117, 143, 194, 247, 294, 344, 389, 424
  - V4.1.1 checklist
    - AIX 431–435
    - Windows 424–428
  - V4.1.1.9 to V4.3.0 143–171
  - V4.1.1.9 to V4.3.0 on AIX 194–415
  - V4.2.1 checklist
    - AIX 446–449
    - Windows 439
  - V4.2.1 to V4.3 on AIX 294
  - V4.2.1 to V4.3 on Windows 247–271
  - V4.2.2 checklist
    - AIX 460–463
    - Windows 453
  - V4.2.2 to V4.3 on AIX 389
  - V4.2.2 to V4.3 on Windows 344–367
- workbench 92
- work-in-progress table 122, 145, 173, 196, 226,

249, 273, 296, 323, 345, 369, 391  
workspace 92

## **X**

Xerces 57

Archived





**Redbooks**

# WebSphere InterChange Server Migration Scenarios

(1.0" spine)  
0.875" <-> 1.498"  
460 <-> 788 pages









**Redbooks**

# WebSphere InterChange Server Migration Scenarios

## **Migration of WebSphere InterChange Server 4.1.1 to 4.3**

This IBM Redbook provides a step-by-step migration guide from previous releases of WebSphere InterChange Server to Version 4.3 for AIX and Windows. A comprehensive outline of the new features of WebSphere InterChange Server after Release 4.1.1 is provided. Planning requirements and pre-migration tasks are discussed in addition to the detailed technical steps required for a successful migration.

## **Migration of WebSphere InterChange Server 4.2.1 to 4.3**

WebSphere InterChange Server, a key component of WebSphere Business Integration Server, provides the capability to enable heterogeneous business applications to exchange data. It addresses the need for efficiency and flexibility in automating and synchronizing business activities. WebSphere InterChange Server 4.3 provides an important base level for delivering support to new operating systems, as well as reliability, availability, scalability, performance, and security enhancements, including use of open standards.

## **Migration of WebSphere InterChange Server 4.2.2 to 4.3**

This book is intended for an audience that is familiar with the WebSphere InterChange Server product and wishes to migrate an existing solution on a previous version of WebSphere InterChange Server to Version 4.3. Information is also included for those less familiar with the product or who are interested in an overview of the key new features of WebSphere InterChange Server 4.3.

## **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](http://ibm.com/redbooks)