

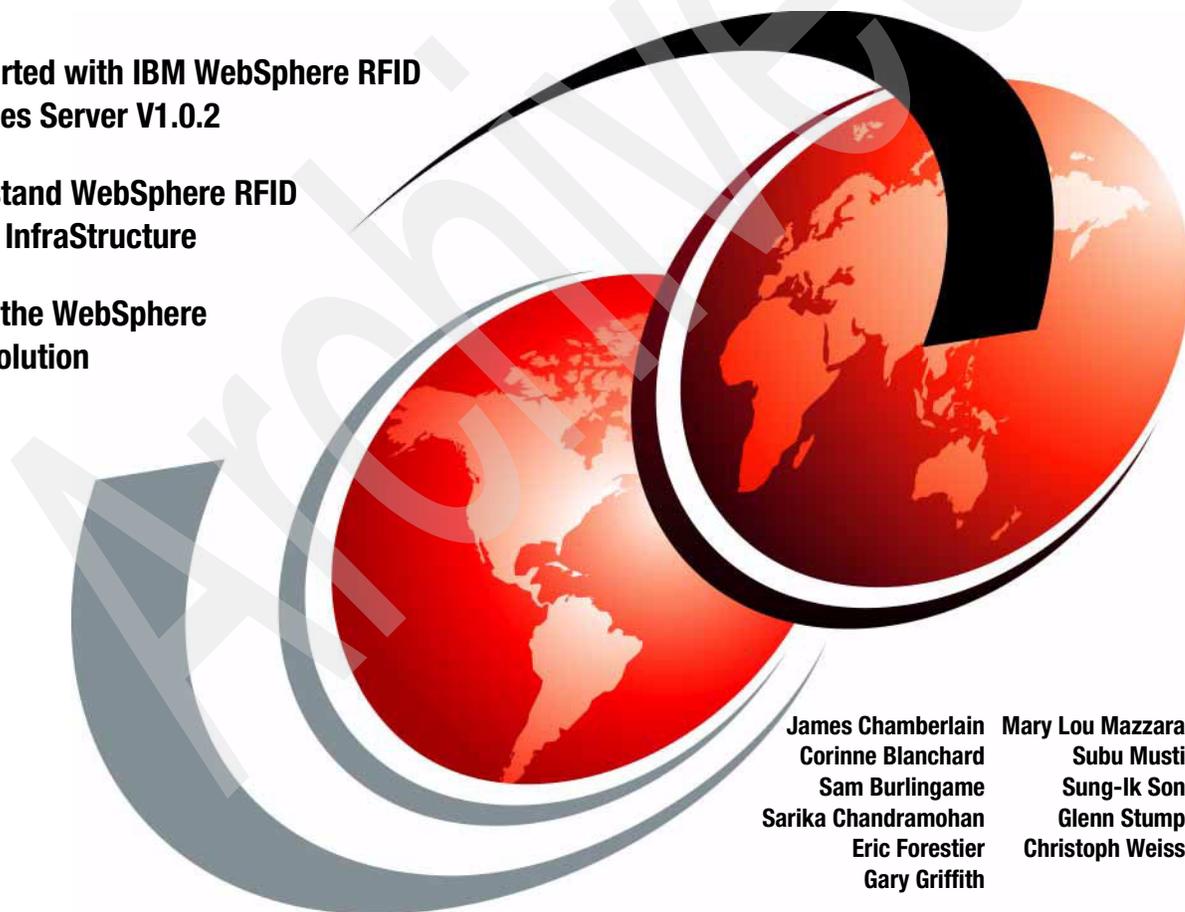
IBM WebSphere RFID Handbook

A Solution Guide

Get started with IBM WebSphere RFID
Premises Server V1.0.2

Understand WebSphere RFID
Device InfraStructure

Set up the WebSphere
RFID Solution



James Chamberlain
Corinne Blanchard
Sam Burlingame
Sarika Chandramohan
Eric Forestier
Gary Griffith

Mary Lou Mazzara
Subu Musti
Sung-Ik Son
Glenn Stump
Christoph Weiss



International Technical Support Organization

IBM WebSphere RFID Handbook: A Solution Guide

May 2006

Archived

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

First Edition (May 2006)

This edition applies to Version 1.0.2 of IBM WebSphere RFID Premises Server.

© Copyright International Business Machines Corporation 2006. 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

Figures	ix
Tables	xiii
Examples	xv
Notices	xvii
Trademarks	xviii
Preface	xix
The team who wrote this book	xx
Become a published author	xxiv
Comments welcome	xxiv
Part 1. Concepts and architecture	1
Chapter 1. Overview of Radio Frequency Identification	3
1.1 RFID basics	4
1.1.1 What is RFID	4
1.1.2 Uses of RFID technology	4
1.1.3 Automatic Identification Systems	8
1.1.4 Antenna designs and functionality	13
1.1.5 RFID tags	15
1.2 Transponders (RFID tags)	20
1.2.1 Passive tag	20
1.2.2 Active tag	20
1.2.3 Semi-active (semi-passive) tag	22
1.2.4 Active or passive tags data access capabilities	22
1.2.5 Surface acoustic wave (SAW) tag	23
1.2.6 Elements of an RFID tag	23
1.3 Exploring RFID at the IBM Wireless Center of Excellence	26
1.4 RFID solution design considerations	30
Chapter 2. Introduction to IBM RFID solutions	35
2.1 IBM RFID Solution Domain Model	36
2.2 Implementing IBM RFID solutions	39
2.2.1 IBM software	39
2.2.2 IBM Integration and Consulting Services	43
2.2.3 IBM Business Partners	43

2.3	Introduction to the WebSphere RFID solution	45
2.3.1	WebSphere RFID Premises Server	47
2.3.2	WebSphere RFID Device Infrastructure	48
2.3.3	IBM Business Integration Server	49
2.3.4	WebSphere RFID Dock Door Receiving Starter Kit (Kimono)	49
2.4	A peek inside the WebSphere RFID solution	50
Chapter 3. IBM WebSphere RFID Premises Server		53
3.1	WebSphere RFID Premises Server overview	54
3.1.1	Key features	56
3.1.2	Premises Server software bundle	56
3.2	WebSphere RFID Premises Server Architecture	64
3.2.1	MicroBroker Bridge	65
3.2.2	RFID Event Server	65
3.3	Event processing	65
3.3.1	Concepts	66
3.3.2	Message flow	68
3.3.3	Dock Door Starter Kit tag_read event example	69
Chapter 4. WebSphere RFID Device Infrastructure		73
4.1	Overview of the WebSphere RFID Device Infrastructure	74
4.2	Key features	77
4.3	Core technologies	80
4.3.1	Workplace Client Technology, Micro Edition	81
4.3.2	WebSphere Sensors and Actuators Base Infrastructure	85
4.3.3	WebSphere Identification Tracking Kit	90
4.4	Internal architecture and operation	91
4.4.1	Agent architecture	91
4.4.2	Operations overview	92
4.5	Dock Door Receiving Starter Kit	97
4.5.1	Example agents	97
4.5.2	Reference implementation	98
Part 2. Setting up the WebSphere RFID solution		99
Chapter 5. Planning your WebSphere RFID solution		101
5.1	Platform support	102
5.1.1	Premises Server hardware requirements	102
5.1.2	Premises Server software requirements	102
5.1.3	Edge Controller requirements	103
5.1.4	Supported RFID Edge Domain devices	103
5.2	Defining your RFID network topology	105
5.2.1	Edge scalability	105
5.2.2	Device scalability	106

5.2.3 Supported layout	106
5.3 Pre-installation checklist	107
Chapter 6. Installing the WebSphere RFID solution.	111
6.1 Installing the Premises Server software	112
6.1.1 Copy the Premises Server CD to a local directory	113
6.1.2 Install DB2 Universal Database	114
6.1.3 Create the IBMRFID database	114
6.1.4 Import the IBMRFID database script files	114
6.1.5 Install WebSphere Application Server.	117
6.1.6 Install WebSphere Application Server Fix Pack	117
6.1.7 Install WebSphere MQ	120
6.1.8 Install WebSphere MQ CSD	121
6.1.9 Configure the environment variables	122
6.1.10 Modify the Event Server configuration properties	124
6.1.11 Run the RFID installation script.	125
6.1.12 Install the SMF stack as a Windows service.	127
6.2 Verifying the Premises Server installation.	130
6.2.1 General purpose verifications	130
6.2.2 SMF stack verification	132
6.2.3 WebSphere Application Server verifications	134
6.2.4 Premises Server verifications	135
6.2.5 Using KimonoPremisesTest bundle	138
6.2.6 Edge Controller configuration verification	140
6.3 Defining network topology	142
6.4 Installing Edge Controller software	143
6.5 Uninstalling the Premises Server software	143
6.6 Defining administrative roles and security.	144
Chapter 7. Administering the WebSphere RFID solution	147
7.1 Before you begin	148
7.2 Administrative Console overview	148
7.2.1 Getting started with the Administrative Console	148
7.3 Defining your RFID network topology	150
7.3.1 Locations.	151
7.3.2 Readers.	156
7.3.3 Agents	161
7.3.4 Controllers.	162
7.4 Viewing configuration and tag data	166
7.4.1 Configuration variables	166
7.4.2 Tags	167
7.5 Configuring your RFID solution extensions.	169
7.5.1 Output channels	170

7.5.2 Event services (templates)	174
7.5.3 Tasks (event handlers)	177
Chapter 8. Running the Dock Door Receiving scenario	181
8.1 Before you begin	182
8.2 Overview of the scenario	182
8.2.1 Using the default network topology	184
8.2.2 Configuring our network topology	185
8.2.3 Simulating a back-end system	190
8.2.4 Using our Dock Door Receiving scenario	195
8.2.5 Behind the scenes	199
8.2.6 Extending the Dock Door Receiving scenario	200
Part 3. Advanced deployment topics	201
Chapter 9. Performance	203
9.1 RFID system performance	204
9.1.1 System performance breakdown	204
9.1.2 System performance test results	206
9.2 Premises Server performance	207
9.2.1 Running SMF as a service	207
9.2.2 Reducing and eliminating logging	207
9.2.3 Known issues and limitations	210
9.3 Edge Controller performance	210
9.4 RFID Reader performance	211
Chapter 10. Monitoring	213
10.1 Tivoli Enterprise Console	214
10.1.1 Installing the Logfile Adapters	214
10.1.2 Monitoring the RFID environment using Tivoli Enterprise Console	217
10.2 Monitoring Other components	217
Chapter 11. Edge Controller Software installation and management	219
11.1 Installing WebSphere Everyplace Device Manager	220
11.1.1 WebSphere Everyplace Device Manager prerequisites	220
11.1.2 Installing WebSphere Everyplace Device Manager	222
11.2 Using WebSphere Everyplace Device Manager	227
11.2.1 The Device Manager console	228
11.2.2 Preparing the WebSphere Everyplace Device Manager Server	229
11.3 Enrolling and configuring the Edge Controller	233
11.3.1 Configuring the OSGi Agent on the Edge Controller	234
11.3.2 Starting SMF on the Edge Controller	235
11.3.3 Configuring the Edge Controller parameters	237

11.4 Verifying the Edge Controller setup	239
Part 4. Appendixes	243
Appendix A. Supported device matrix	245
A.1 IBM WebSphere RFID V1.0.2 matrix	246
A.2 IBM WebSphere RFID PVS Starter Kit V1.0.2 matrix.	247
Appendix B. Supported software matrix.	249
B.1 IBM WebSphere RFID Premises Server V1.0.2 matrix	250
B.2 IBM WebSphere RFID Premises Server PVS Starter Kit V1.0.2 matrix	251
Appendix C. Agents, properties, and values	253
C.1 Reader agents	254
C.1.1 AlienReaderAgent	254
C.1.2 IntermecReaderAgent	255
C.1.3 MatricsReaderAgent.	256
C.1.4 SamsysReaderAgent	257
C.1.5 TagSysReaderAgent	258
C.1.6 SymbolReaderAgent	260
C.1.7 FeigUHFReaderAgent	261
C.2 Printer agents	262
C.2.1 ZebraPrinterAgent	262
C.2.2 PrintronixPrinterAgent	263
C.3 Controller agents	264
C.3.1 FilterAgent	264
C.3.2 ArcomIoDkReaderAgent.	265
C.3.3 ControllerAgent.	266
C.3.4 PrinterControllerAgent	266
C.4 Other device agents	266
C.4.1 LightTreeAgent.	266
C.4.2 MotionSensorAgent	267
C.4.3 SwitchAgent	267
C.5 Other agents	267
C.5.1 DutyCycleAgent	267
C.5.2 SelfTestAgent.	268
Glossary	269
Abbreviations and acronyms	271
Related publications	273
IBM Redbooks publications	273
Other publications	273

Online resources	273
How to get IBM Redbooks	274
Help from IBM	274
Index	275

Archived

Figures

1-1	Example of an Ameritech Wireless RFID toll tag card	4
1-2	Examples of RFID transponders	5
1-3	Encapsulated agricultural and pet RFID tag	6
1-4	Intellitag RFID by Intermec Technologies	6
1-5	RFID frequency allocation standards	7
1-6	Static Barcode label	8
1-7	Example of a key fob security token	9
1-8	Example material characteristics	10
1-9	Simplified RFID system	11
1-10	Visual representation of an RFID system	12
1-11	Graphical representation of perfect broadcast	13
1-12	Typical RF broadcast pattern	14
1-13	Reader location for optimum performance	15
1-14	Intermec mini dipole RFID tag insert	16
1-15	Individual case tag locations	17
1-16	Product tags per case	17
1-17	Illustration of the complexity for pallet, case, and product tags	18
1-18	Visual representation of a pallet of cases	18
1-19	Absorption of the electromagnetic field when water is the medium	19
1-20	Saturation chart with typical characteristics of frequency and samples	19
1-21	Example of a passive tag - Intermec Butterfly RFID tag insert	20
1-22	Example of an active tag - RF Code Mantis tag	21
1-23	Elements of an RFID tag	23
1-24	Example of a chip-size RFID tag compared to a U.S. postage stamp	24
1-25	Example of an RFID tag insert	25
1-26	IBM Wireless Center of Excellence RFID testing facility	26
1-27	IBM dock door RFID portals at the Wireless Center of Excellence	27
1-28	The team in the Wireless Center of Excellence	28
1-29	The Wireless Center of Excellence state-of-the-art conveyor system	29
1-30	Performing an RFID site survey	30
1-31	Radio Frequency interference factors	32
1-32	Sample site survey form	33
2-1	The IBM RFID Solution Domain Model	36
2-2	Mapping the IBM RFID Solution Domain	40
2-3	Software and tooling for complete RFID solution implementations	42
2-4	IBM RFID Consulting and Services and IBM Business Partners roles	44
2-5	WebSphere RFID solution implementation	45
2-6	WebSphere RFID solution: components and functions	46

2-7	Arcom Viper Edge Controller	48
2-8	A peek inside the WebSphere RFID Solution	50
2-9	WebSphere RFID solution: dock door example flow	51
3-1	The Premises Server in context of IBM RFID Solution Domain Model	54
3-2	Premises Server in the context of RFID solution architecture	55
3-3	WebSphere RFID Premises Server software bundle.	57
3-4	RFID Event Server component overview.	64
3-5	RFID Event Server message flow	69
4-1	RFID Domain Model - Edge Domain	74
4-2	WebSphere RFID Device Infrastructure pervasive overview.	75
4-3	Edge Controller overview.	77
4-4	WebSphere RFID Device Infrastructure overview	80
4-5	Workplace Client Technology Micro Edition overview	82
4-6	Service Management Framework architecture	83
4-7	WebSphere Sensors and Actuators Base Infrastructure	86
4-8	OSGi Application Framework Architecture	87
4-9	WebSphere Connection Server Micro Edition Architecture	88
4-10	WebSphere Connection Server Micro Edition Application Framework	90
4-11	WebSphere Identification Tracking Kit.	90
4-12	Agent Architecture overview	91
4-13	Edge configuration overview	92
4-14	MicroBroker Bus and MicroBroker Bridge Configuration operation	94
4-15	Device communications.	95
4-16	Additional extension options	96
4-17	Dock Door Receiving reference implementation	98
5-1	Edge Domain physical layout.	104
5-2	RFID solution topology.	107
6-1	Connect to IBMRFID database using DB2 Command Center	115
6-2	Importing RFID application DDL file.	116
6-3	Removing WebSphere MQ tray icon	117
6-4	Update installation wizard window	118
6-5	WebSphere Application Server set to start automatically	119
6-6	WebSphere MQ Setup settings	121
6-7	IVEHOME variable.	122
6-8	MQ_JAVA_DATA_PATH variable	123
6-9	MQ_JAVA_INSTALL_PATH variable	123
6-10	IBM_RFID_HOME variable	123
6-11	IBM WebSphere RFID Premises Server license agreements	126
6-12	IBMSMFService registry key	128
6-13	Adding a multi-string value key	128
6-14	Modifying registry key DependOnService	129
6-15	Setting multi-string key DependOnService	129
6-16	New registry key DependOnService	130

6-17	IBM.RFID.QM status on WebSphere MQ console	131
6-18	Premises Server defined WebSphere MQ queues	131
6-19	Premises Server WEB applications	135
6-20	Premises Server Administrative Console	136
6-21	KimonoPremisesTest bundle	138
6-22	Edge Controller configuration on the Premises Server	141
7-1	IBM WebSphere RFID Premises Server Administrative Console	149
7-2	Sample RFID network topology	151
7-3	Locations	152
7-4	Locations contacts	153
7-5	Edit Location details	154
7-6	Create new Location	154
7-7	Display location details	156
7-8	Readers page	157
7-9	An example of a reader definition	157
7-10	Reader types	158
7-11	Create a new Reader page	158
7-12	Reader search	160
7-13	Edit Reader details page	160
7-14	Edit Agent Properties page	161
7-15	Controllers	162
7-16	Create new Controller	163
7-17	Controller locations	163
7-18	Controller readers	164
7-19	Reload Config	165
7-20	Restart verification	165
7-21	Configuration variables	166
7-22	Tags	167
7-23	Tag search	168
7-24	View Tag Details page	168
7-25	View Tag History page	168
7-26	View Tag Metadata page	169
7-27	Relationship between tasks, event templates and output channels . .	169
7-28	Output channels	170
7-29	Create new Email Output Channel	171
7-30	Modify JMS Output Channel Details	174
7-31	Event Templates	175
7-32	Create Event Template	176
7-33	Event Template channels	176
7-34	Event Template Detail	177
7-35	Tasks	178
7-36	Create new Task	179
7-37	Edit Task Details	180

8-1	Dock Door Receiving Scenario	183
8-2	End-to-end message flow	184
8-3	ITSO lab locations	185
8-4	ITSO lab location contacts	186
8-5	ITSO lab readers	186
8-6	Alien reader pin settings	187
8-7	ITSO lab controllers	189
8-8	Activating the edge devices	196
8-9	Reading valid tags	197
8-10	Reading an invalid tag	198
8-11	Viewing tag data	199
9-1	System performance breakdown	205
9-2	WebSphere Application Server Diagnostic Trace settings	208
11-1	Setting the JAVA_HOME environment variable	221
11-2	Configuring the WebSphere Everyplace Device Manager Database	223
11-3	Setting Global Security	225
11-4	WebSphere Everyplace Device Manager and the RFID solution	227
11-5	Verifying the OSGi Device Class in the Device Manager Console	228
11-6	Confirming job creation	233
11-7	Viewing enrolled devices	236
11-8	Viewing jobs and job status	240
11-9	Monitoring job progress	241
11-10	Device specific job progress	241

Tables

3-1	RFID Event Server queues	67
5-1	RFID Network Topology Concepts	105
5-2	Software planning and hardware prerequisite checklist	107
5-3	Software planning - needed software packages checklist	109
5-4	Software planning - needed topology items checklist	110
6-1	IBM WebSphere RFID Premises Server CD package layout	112
6-2	KimonoPremisesTest - testing components	138
6-3	Event Simulator Events	139
6-4	Event handler event messages	139
6-5	Command simulator commands	140
7-1	Types of Output Channels	170
7-2	Premises Server Predefined Output Channels	171
7-3	Field definitions for various types of output channels	172
8-1	Locations and contacts for the ITSO lab controllers and readers	185
9-1	Stages in system performance breakdown	205
9-2	Alerts generated for the basic events necessary to read a single tag	211
11-1	Variables	230
11-2	Variables	232
11-3	OSGi Agent properties	234
11-4	Variables	238
11-5	IBM WebSphere RFID V1.0.2 supported Edge Controllers	246
11-6	IBM WebSphere RFID V1.0.2 supported RFID readers	246
11-7	Starter Kit supported Edge Controllers	247
11-8	Starter Kit supported RFID readers	247
11-9	Starter Kit supported RFID printers	248
B-1	IBM WebSphere RFID Premises Server V1.0.2 required software	250
B-2	Everyplace Device Manager V5.0 FP 1 required software	250
B-3	Premises Server PVS Starter Kit required software	251
C-1	AlienReaderAgent properties	254
C-2	IntermecReaderAgent properties	255
C-3	MatricsReaderAgent properties	256
C-4	SamsysReaderAgent properties	257
C-5	TagSysReaderAgent properties	258
C-6	SymbolReaderAgent properties	260
C-7	FeigUHFReaderAgent properties	261
C-8	ZebraPrinterAgent properties	262
C-9	PrintronixPrinterAgent properties	263
C-10	FilterAgent properties	264

C-11	ArcomIoDkReaderAgent properties	265
C-12	LightTreeAgent properties	266
C-13	MotionSensorAgent properties	267
C-14	DutyCycleAgent properties	267
C-15	SelfTestAgent properties	268

Archived

Examples

3-1	Sample XML message that represents the reading of an RFID tag . . .	66
3-2	MQTT message from the Edge Controller	70
3-3	MicroBroker Bridge transformed message	70
3-4	Message selector for the Dock Door Receiving Event Handler MDB . .	71
6-1	RFID Installation script log - part 1	125
6-2	RFID installation script log - part 2	126
6-3	Installation log phases	127
6-4	SMF as a service installation log	127
6-5	SMF stack starting log	132
6-6	SMF installed bundles in the Premises Server	136
6-7	Premises Server uninstallation log	144
8-1	Default values for AlienReaderAgent	187
8-2	IntermecReaderAgent settings	188
8-3	Setting tag persistence	189
8-4	Premises Server SMF console	190
8-5	XML message example with type='tag_read'	192
8-6	premises-test.properties file	192
8-7	premises-test.properties file (modification 1)	193
8-8	premises-test.properties file (modification 2)	194
8-9	Output from starting the simulator	194
11-1	startDMS.bat	226
11-2	stopDMS.bat	226
11-3	createWedmSoftware.xml	230
11-4	createWedmJobs.xml	231
11-5	OSGiAgent.properties.bak	234
11-6	S99startKimono	236
11-7	updateParameters.xml	237
11-8	Creating jobs with xmlConfig.bat	239

Archived

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

Trademarks

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

Redbooks (logo) ™

xSeries®

AIX®

CICS®

Domino®

DB2 Universal Database™

DB2®

Everyplace®

Informix®

IBM®

IMS™

Lotus®

MQSeries®

Redbooks™

RS/6000®

Tivoli Enterprise™

Tivoli Enterprise Console®

Tivoli®

WebSphere®

Workplace™

Workplace Client Technology™

The following terms are trademarks of other companies:

EJB, Java, JDBC, JVM, J2EE, J2ME, Solaris, Sun, Sun Microsystems, Ultra, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, 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.

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

Ameritech® is a registered trademark of Transcore in the United States and other countries.

Alien® and Alien Technology are trademarks or registered trademarks of Alien Technology Corporation in the United States and other countries.

Intermec® is a trademark or registered trademark of Intermec Technologies Corporation in the United States and other countries.

Arcom® is a trademark or registered trademark of Arcom, Inc. in the United States and other countries.

ThingMagic® and Mercury4™ are trademarks or registered trademarks of ThingMagic in the United States and other countries.

Printronix®, SL5000e™, SL5000r™, and RFID_Smart™ are trademarks or registered trademarks of Printronix in the United States and other countries.

Zebra Technologies is a trademark or registered trademark of Zebra Technologies in the United States and other countries.

OSGi® is a registered trademark of The OSGi Alliance in the United States and other countries.

PATLITE® is a trademark or registered trademark of PATLITE in the United States, Japan, and other countries.

Loftware® is a trademark or registered trademark of Loftware in the United States and other countries.

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

Preface

This IBM® Redbooks publication provides an overview of Radio Frequency Identification (RFID) technology and will help get you up and running with the WebSphere® RFID solution, a new offering that enables you to connect the edge of your environment with your enterprise and to enhance your on demand business capabilities. This book explains the key products and components that are included in the solution, with a focus on WebSphere RFID Premises Server and Device Infrastructure. You will learn how to install and to configure the Premises Server, to connect and to communicate with your edge devices, and to implement the Dock Door Receiving Starter Kit.

Part 1 starts with a broad picture of RFID technology and explains how the WebSphere RFID solution fits into that overall technology. It details the features and functions of the Premises Server and includes information about the PVS Starter Kit technology preview.

Part 2 gives step-by-step instructions for installing the WebSphere RFID Premises Server and related software. It explains the Administrative Console and the tasks that are required to get your system started. You will learn how to use the Administrative Console to configure the Premises Server, Edge Controllers, RFID readers, and the components that enable them to communicate.

Part 3 explains what the book team did to configure the WebSphere RFID system and to run the Dock Door Receiving sample scenario. It shows you the variables that we set, the tasks that we performed, and the results that we obtained. It also offers suggestions for extending this reference implementation to meet the needs of your enterprise.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.

James Chamberlain is a Senior Software Engineer and certified Senior IT Specialist. He is a project leader at the ITSO, Raleigh Center. He has over 24 years of experience in the IT industry and specializes in pervasive computing technologies. His areas of expertise include e-commerce, pervasive computing, portals, AIX®, Linux®, and Java™ programming. He also architects, designs, and develops solutions using J2EE™, XML, Web Services, and IBM software products, including WebSphere and DB2®. Before joining the ITSO, James worked for IBM Global Services on e-commerce system development for IBM Business Partners. He majored in Computer Science at Iowa State University.

Corinne Blanchard is a Business Consulting Services (BCS) IT Specialist in France. She has 18 years of experience in the application development field. She has been working extensively with WebSphere RFID Premises Server and tools for the last year. She helped customers such as Sernam, Renault, and IBM-ISC Montpellier to implement, tune, and customize RFID applications using IBM middleware.

Sam Burlingame is a Software Engineer with IBM Software Group, Application and Integration Middleware Software. He has worked in Sensors and Actuators RFID Middleware Test for the past year and a half, specializing in functional and performance testing as well as test automation. He has been involved with the testing and improvement of the IBM WebSphere RFID solution from both product and customer specific standpoints. Sam holds a Bachelor's degree in Electrical Engineering from Tufts University.

Sarika Chandramohan is a Solutions Architect with the India Software Labs (ISL) in Bangalore, India. She has five years of experience in the IT industry and has worked on domains that span mobile solutions, retail and distribution, and RFID. Her skill sets include C/C++, J2ME™, and J2EE. At ISL she works with the Distribution Sector Solutions team and is involved with activities that include pre-sales, solution development, and customer pilots. Sarika has a degree in Computer Science and Engineering from the University of Bangalore, India.

Eric Forestier is an IT Architect and works at the e-business Solutions Center in IBM La Gaude, France. He currently is working with the IBM Pervasive Computing Division and provides EMEA advanced technical support to Independent Software Vendors, assessing and enabling IBM Partners to include the pervasive computing assets into their solutions. He has also been a system software developer, working in various areas such as networking, telephony, and Internet.

Gary Griffith is an Enterprise IT Systems Engineer for The Boeing Company in Seattle, Washington. He has 24 years experience in the IT industry, specializing in large-scale and distributed systems integration. He has co-authored *Using Tivoli's ARM Response Time Agents*, SG24-2124 in 1998 and *IBM WebSphere Host On-Demand: Version 5 Enhancements*, SG24-5989 in 2001.

Mary Lou Mazzara is a software engineer with the Application Integration Middleware (AIM) Demo Solutions Development team in Research Triangle Park, NC. She has 25 years of experience in field of software development, 16 of them with IBM. Her areas of expertise include technical communication, user interface design, and Web development. She has worked extensively on WebSphere products, starting with the very first release of WebSphere Application Server. She holds a Master of Science degree from Rensselaer Polytechnic Institute in Troy, NY, and a Bachelor of Arts degree from Vassar College in Poughkeepsie, NY.

Subu Musti is an RFID solution Architect with the Wireless Applications practice of IBM Global Services, USA. He is also an IBM Certified IT Specialist in Business Applications. He has 10 years of IT industry experience, implementing customer solutions in a broad range of technologies and industry verticals. He is a founding member of the IBM WebSphere RFID solution team and has been actively engaged in implementing RFID solutions for the last three years. He holds a Master's degree from the North Carolina A&T State University and a Bachelor's degree from the Indian Institute of Technology (IIT).

Sung-Ik Son is a Senior Software Engineer for IBM Software Services at the IBM Raleigh Lab. He has been a key developer for more than 12 years in IBM system and application software development organizations. He has worked for the past six years on the WebSphere Enablement team. Sung-Ik's focus area has been wireless, mobile, and pervasive applications, which his current focus and responsibilities including RFID. He holds a B.S. in Computer Science from the University of New Brunswick in Canada and an M.S. in Computer Science from Purdue University.

Glenn Stump is a Senior Solutions Architect for IBM WebSphere Business Partner Enablement team in RTP, NC. He has 23 years of experience in networking software and has been a member of the IBM Pervasive Computing product team since 2000. Glenn has advocated mobile and wireless solutions within IBM and has developed and administered wireless solutions to support the IBM Pervasive software sales team. Most recently, Glenn has been engaged in new Sensors and Actuators business at IBM, working with IBM Business Partners who use the WebSphere RFID solution. He holds a Bachelors degree in Computer Science from Pennsylvania State University.

Christoph Weiss is a IT specialist in Wireless Broadband and Sensing Solutions EBO., IBM Deutschland GmbH. He has been working for IBM since 1999. Within

his current role, he is responsible for the design and integration of embedded IT solutions with special emphasis on sensor networks and RFID. In the last year he was for example involved in the METRO Future Store Initiative project. He studied business information technology at the University of Cooperative Education in Stuttgart.

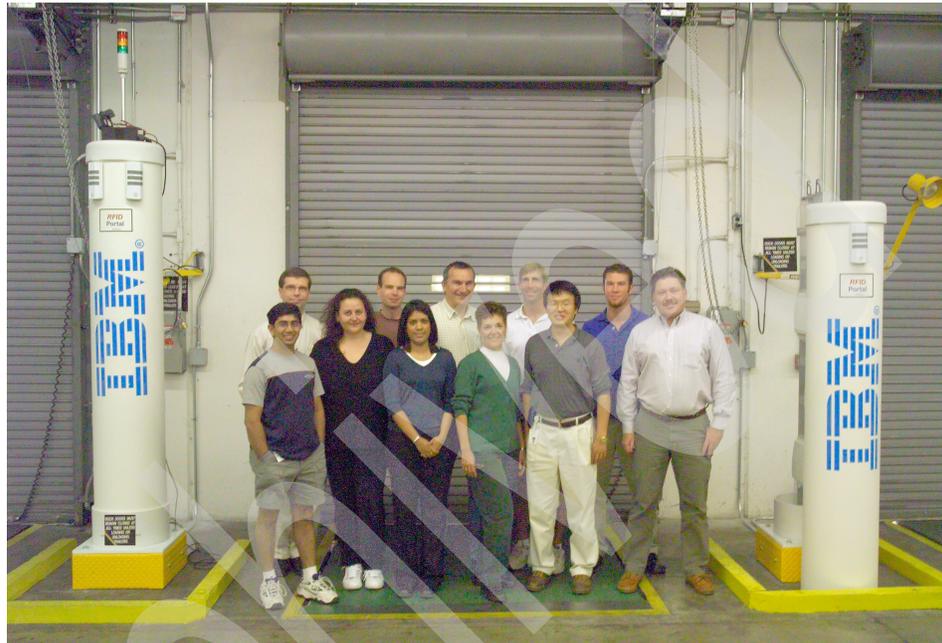


Figure 1 The team for this book in the dock door receiving area that is equipped as an RFID portal at the IBM Wireless Center of Excellence in Research Triangle Park, North Carolina

Front row, from left to right: Subu Musti, Corinne Blanchard, Sarika Chandramohan, Mary Lou Mazzara, Sung-Ik Son, Gary Griffith; Back row, from left to right: James Chamberlain, Christoph Weiss, Eric Forestier, Glenn Stump, Sam Burlingame

Thanks to the following people for their invaluable contributions that made this project possible:

Joshua Barnhardt
Deidre Lenderking
Peter Rossi
James Rutledge

IBM Global Services, Integrated Technology Services

Samuel Camut
Kevin T. Chu
Scott de Deugd
Steven De Gennaro
Yi-kuan Lee
Fred Rowe
Anne I. Ryan
John Senegal
Dave Soroka
Allen Smith
Greg Smith
Leslie Wiggins

IBM Software Group, Application and Integration Middleware Software

Faye Holland
IBM Global Services, Headquarters

Aldo Eisma
IBM Global Services, IBM Business Consulting Services

John Baker
Douglas Hunt
Kimberly Krostue
IBM Global Services, Application Innovation Services

Nicolas Comete
Joël Viale
IBM Sales and Distribution, Operations

And a special thanks to our ITSO support staff at the Raleigh Center:

Margaret Ticknor
Jeanne Tucker
Tamikia Barrow
Linda Robinson

Thanks to our ITSO management:

Jere Cline

And a special thanks to our IBM Pervasive Computing sponsor:

Mary Fisher

Jim Toohey

Become a published author

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

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

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

redbook@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400



Part 1

Concepts and architecture

In this part, we give you a broad picture of RFID technology and explain how the WebSphere RFID solution fits into that overall technology. It details the features and functions of the Premises Server and includes information about the PVS Starter Kit technology preview.

Archived



Overview of Radio Frequency Identification

In this chapter, we give you the basics of understanding Radio Frequency Identification (RFID) and describe some of the challenges that RFID technology faces.

1.1 RFID basics

This section gives you a brief overview of RFID. It also describes possible uses of RFID technology.

1.1.1 What is RFID

RFID is a term used to describe generically a system that transmits a unique identity that uses radio waves.

1.1.2 Uses of RFID technology

In the early 1970s, the government was looking for different ways to track hazardous nuclear material. Los Alamos National Laboratory developed an RFID model which put a transponder in each vehicle; in turn, this transponder was activated by an antenna at a gate. This model provided the ability to track materials that entered or left the facilities.

Today, the application of this technology is used throughout the world in transportation payment collection systems. Many new roads, bridges, or commuter toll collection systems have been replaced by this type of technology. Figure 1-1 shows the toll tag card that allows a transportation authority to collect payment on specific tollway.



*Figure 1-1 Example of an Ameritech Wireless RFID toll tag card
Photograph courtesy of Transcore*

The technology provides increased efficiency or flow of traffic because the vehicle is not required to stop. This automation also reduces costs.

Agricultural applications include placing an encapsulated tag (transponder) underneath the skin of livestock (including pets). This application has proven to be very useful for tracking historical information, such as inoculations or the owner's contact information in the case of a lost pet. Figure 1-2 shows various examples of RFID transponders that are available from Texas Instruments.



*Figure 1-2 Examples of RFID transponders
Photograph courtesy of Texas Instruments*

Figure 1-3 illustrates an encapsulated passive tag. One application of this type of tag, which is just a few millimeters in size, is that it can be inserted under the skin of a young puppy. The chip includes the dog's registered name, birth date, recent inoculations, and ownership information. Even though the animal's physical characteristics might change during growth to adulthood, the tag can provide positive identification with 100% accuracy in the event that the pet becomes lost.

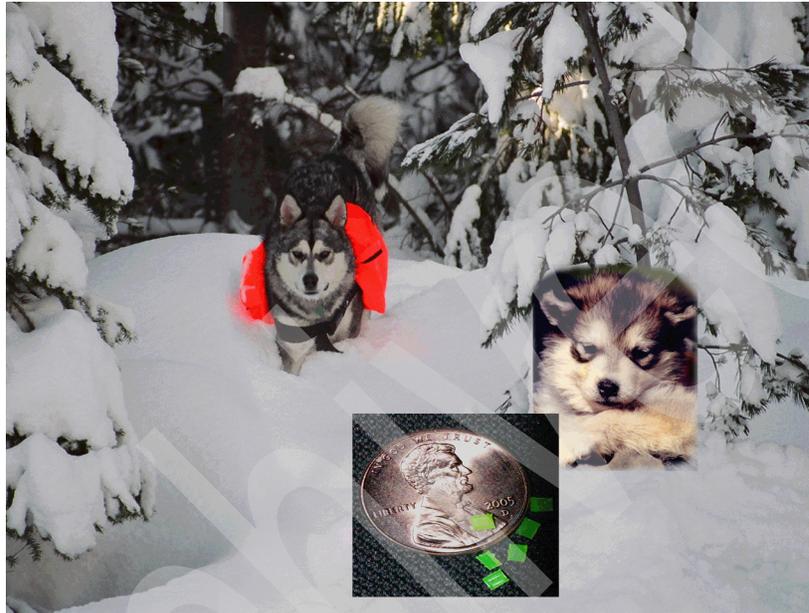


Figure 1-3 Encapsulated agricultural and pet RFID tag

Security access or access control credentials that many of us use today provide different examples of how RFID technology has been exploited. Access can be granted, restricted, or removed without collection of the physical card. Many financial institutions embed this type of technology into the cards for identification purposes (Figure 1-4).



Figure 1-4 Intellitag RFID by Intermec Technologies
Photograph courtesy of Intermec Technologies

The largest area of growth for RFID technology is supply chain inventory tracking. Manufacturers have the potential to track inventory from the point the item was produced to when it was consumed. This method of tracking inventory differs significantly from the Universal Product Code (UPC) barcode, which is an industry-wide as a product identifier.

Using RFID, the product can be tracked throughout the supply chain, which provides advantages for inventory control (just in time) or identifies the location of the unique product that might have an expiration date or needs to be recalled.

Global governing agencies are defining standards so that RFID technology can be used throughout the world.

Concerns or issues such as non-duplication or counterfeiting of tags are concerns from the local, national, and global perspectives. Included in these concerns are standards for frequency's use on an inter-continent perspective. Figure 1-5 indicates the typical frequency used in a geographical area.

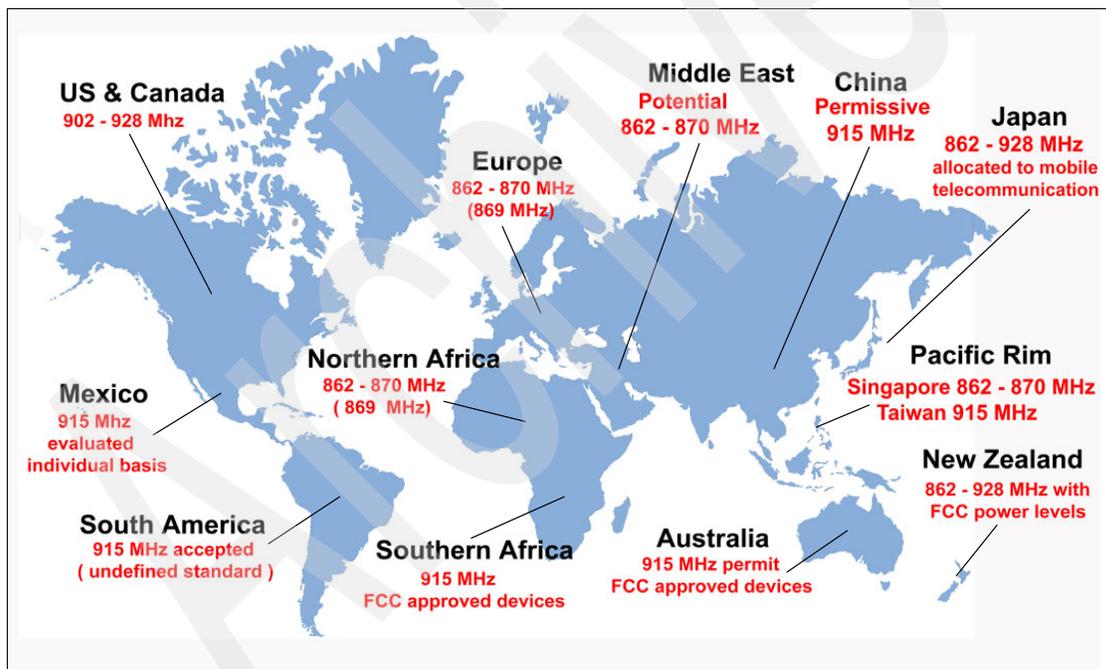


Figure 1-5 RFID frequency allocation standards

1.1.3 Automatic Identification Systems

RFID is a type of *Automatic Identification System*. In this section, we contrast and compare barcodes with RFID technology.

Barcodes

A barcode is a series of machine readable lines of varying widths (Figure 1-6). When translated, it represents a series of alphanumeric characters. Barcodes are a static representation of a specific product number or Universal Product Code (UPC). This code, when scanned or input into a collection point, is transmitted to a delivery system. The delivery system, depending on the application, translates and provides the appropriate response for that product.



Figure 1-6 Static Barcode label

Barcodes have several limitations in comparison to RFID tags:

- ▶ The information about a barcode cannot be changed dynamically and must be read as a individual unit.
- ▶ The barcode label does not contain product information. Encoded in the barcode are alphanumeric numbers that correspond to a specific product. It requires an external delivery system to maintain and to use this information.
- ▶ Barcodes labels are very inexpensive to generate but the trade-off is the limitation of actual data that is maintained at the product level. The label does not lend itself easily to automated tracking and requires an interface to back-end systems.

Smart Cards

Smart Cards are usually about the size of a typical credit card and come in two varieties. The simplistic model is a card with memory and uses the reader for functionality. The reader also provides the security element to ensure the integrity of the card.

The second type of Smart Card, in the form of a key fob (Figure 1-7), relies on an embedded microprocessor. Its function can be similar to a mobile computer but significantly smaller. It can add, delete, or modify information that is stored on the card and has built-in security functions.



Figure 1-7 Example of a key fob security token

Interfaces for these types of cards are accomplished in several ways. They might use a Smart Card *reader* where the cards are inserted into the device. Another approach is *proximity* card or device — when the card is in the general vicinity of a reader it executes the transaction in a wireless mode. A *dual combination* card combines the functionality of both types into one card.

Radio Frequency ranges

Radio Frequency (RF) is electromagnetic waves that have a frequency range of 30 Hz to 300 GHz. RFID typically uses a frequency range between 30 Hz and 5.8 GHz.

The frequency break down into four categories that provide unique qualities and can be used in a variety of ways. The waves that are generated at these different frequencies pass through many types of materials but exhibit different characteristic or results. The categories are:

- ▶ Low-Frequency (LF) 125 KHz 134 KHz
- ▶ High-Frequency (HF) 13.56 MHz
- ▶ Ultra™-High Frequency (UHF) 303.8 MHz 433 MHz 868 MHz
- ▶ Microwave Frequency (MW) 915 MHz

Impact of material characteristics with regard to RF

Figure 1-8 represents how RF waves transmit through different types of materials. When selecting RFID solutions to support a delivery system, you should consider these material characteristics when selecting the optimum frequency. Figure 1-8 does not address limitation of distance or speed. Materials that are lucent react very well in all frequency ranges. The RF waves travel through the object with significant distortion. Absorbent materials react differently to the higher frequency ranges.

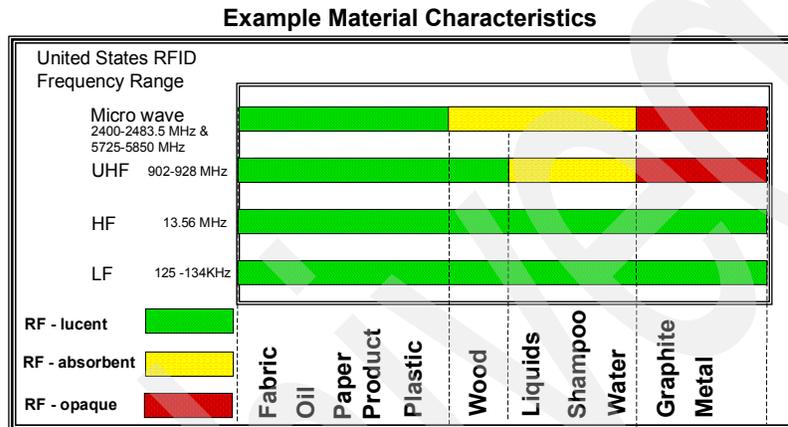


Figure 1-8 Example material characteristics

Simplified RFID system

A simplified RFID system is composed of two main components (Figure 1-9). A transceiver (reader or write/read) with one or more attached antennas and a transponder (RFID tag). The integration of several additional components is required to exploit RFID technology effectively. These components are a controller, sensors, annunciators, actuators, and connectivity to back-end business systems.

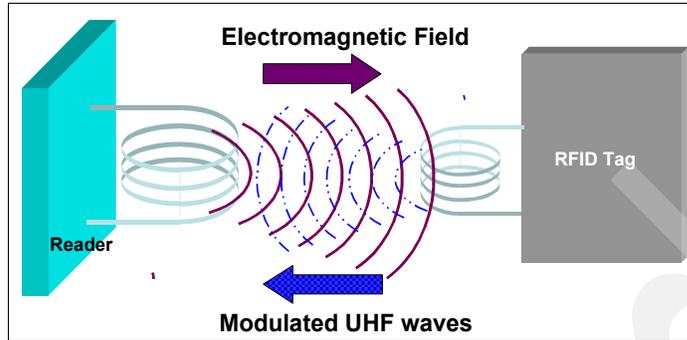


Figure 1-9 Simplified RFID system

The *transceiver* or *reader* is responsible for transmitting and receiving signals in a defined target environment. The originating signal (ac power and clock cycle) in the form of an electromagnetic field is used (in a passive environment) by the transponder or the RFID tag. The tag uses the signal to activate and to generate a modulated wave, which is the corresponding analog signal. This signal is transmitted by the RFID tag that is used during the interrogation by the reader's antenna. The information that was received by the antenna is then processed by the reader and converted into a digital form. This data stream is then forwarded to the controller for additional processing.

The *controller* provides essential communication links between the reader and external entities. The functionality of the controller can be embedded into the reader or can act as an individual component that serves one or more readers. In addition, the controller incorporates different types of communication interfaces (RS232, RS485, 802.x, and so forth).

These interfaces can provide connectivity to sensors, annunciators, and actuators that act as triggers. The response that is generated by the activation of a trigger can enable or control specific reactions. Other interfaces can establish communication to other readers, controllers, or even back-end business systems. Sensors, annunciators, and actuators can provide a certain level of automation to the RFID system.

Figure 1-10 shows an example of an RFID system that might be used in a dock door receiving scenario.

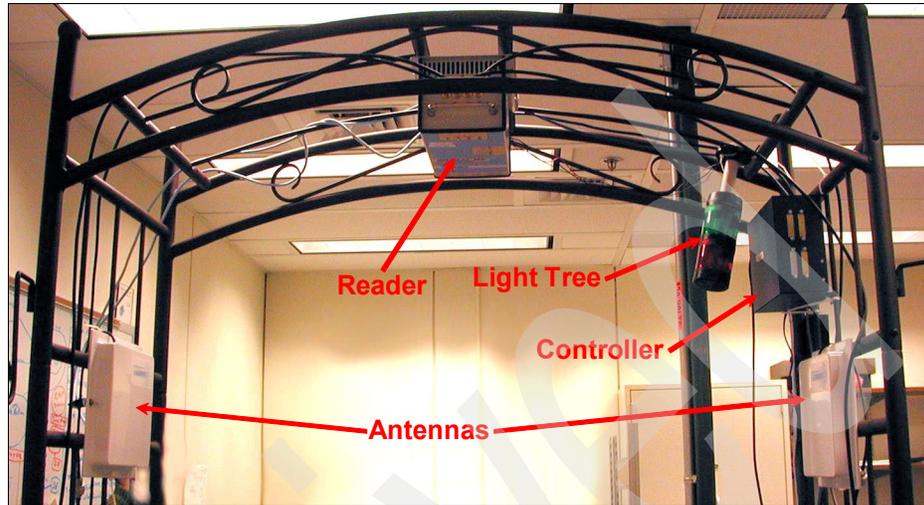


Figure 1-10 Visual representation of an RFID system

In a dock door receiving scenario, which we discuss in Chapter 8, “Running the Dock Door Receiving scenario” on page 181, incoming merchandise activates a motion sensor device. A high-level overview of the process is as follows:

- ▶ The motion sensor creates a response that sets a trigger.
- ▶ In turn, the controller receiving the trigger activates the appropriate reader interrogation zone at a specific dock door.
- ▶ The reader interrogates the contents of the pallet of merchandise.
- ▶ The inventory list that is created by the reader can be reconciled against the invoice manifest of merchandise.
- ▶ At the completion of the interrogation process the controller initiates another trigger to an annunciator or actuator which indicates that the process is complete. This indicator might be in the form of an audible sound, a visual light tree, or even closing the dock door.

Behind the scenes, the controller provides the interface with back-end business systems. Information about the inventory can feed or update different business systems depending on requirements. Depending on the business requirements and deliverables, information flow can be bidirectional from a controller. For example, the controller could provide information to the reader which is then broadcast to the active tag for update. Alternatively, in reverse, the reader provides a data stream to the controller that updates a back-end system with new product information, inventory, and so forth.

1.1.4 Antenna designs and functionality

When designing an interrogation zone, you should take into account the expectations of the RFID solution. Are you designing it for speed, accuracy, or distance? Based on the laws of physics (which govern electromagnetic fields, radio frequencies, and communication), accomplishing all of these simultaneously is not an achievable goal. You can, however, set a goal to optimize any two of these, such as speed and accuracy.

As you might suspect, not all antennas are created equally. Outside influences can have detrimental effects on the accuracy and dependability of the deployment and targeted solution. The basic function of an antenna is fundamentally the same — to transmit and to receive. Then, the laws of physics apply, and length, size, power, frequency, and other factors contribute to the overall design of the antenna.

In an ideal world, the antenna pattern radiates while expanding across the X, Y, and Z axis. In Figure 1-11, the Z axis represents the antenna. The RF wave radiates outward on the X and Y axis. Unfortunately, the actual antenna pattern is somewhat more complex than this simple diagram.

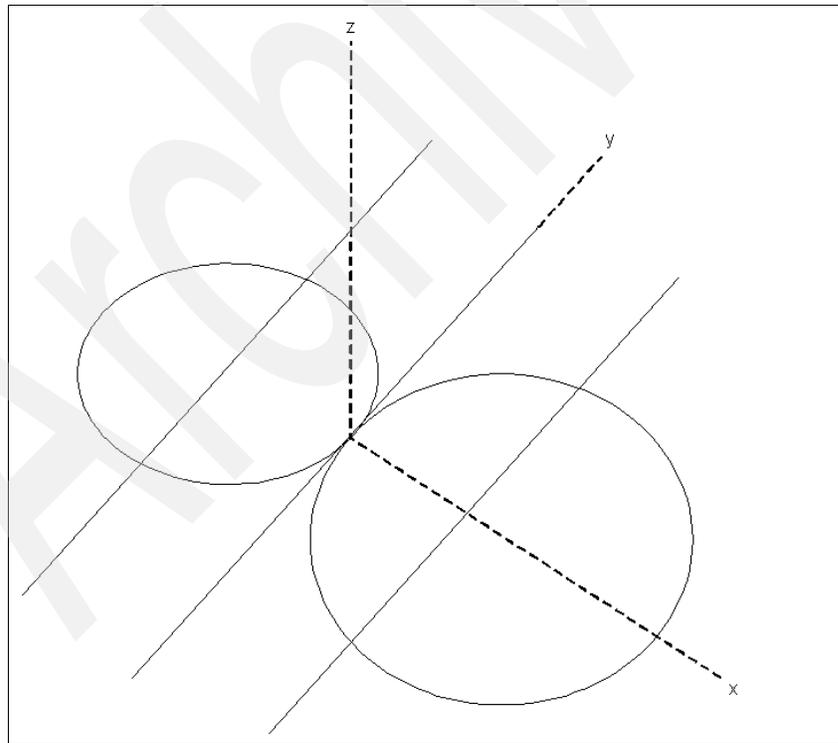


Figure 1-11 Graphical representation of perfect broadcast

Depending on the type of antenna selected, its length, and the power that is applied to the transmission during a broadcast, a typical RF broadcast pattern would look similar to that shown in Figure 1-12.

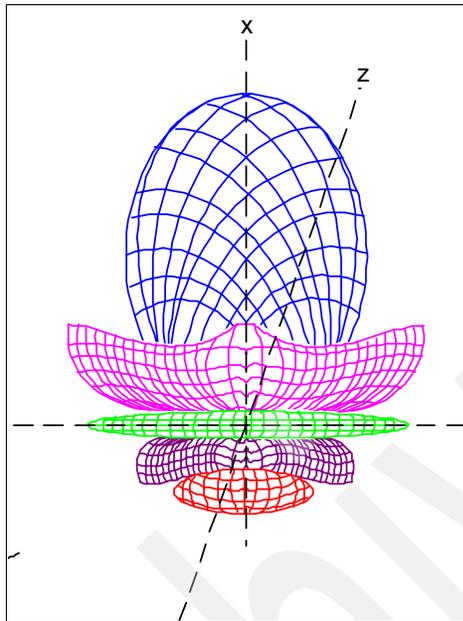


Figure 1-12 Typical RF broadcast pattern

Obstruction or objects can potentially distort this typical RF pattern if introduced. Most manufactures provide a similar broadcast pattern for specific antenna designs. This information should provide a basis for establishing the interrogation zone.

A spectrum analyzer can be used to validate the anticipated coverage and strength of the signal that is broadcast from the antenna.

To understand what influences can impact your design, you should complete a thorough site survey, probably by the architect or integrator. The use of a spectrum analyzer to identify potential RF frequencies that can interfere will eliminate most of the guess work for the design. RF interference, signal strengths, or specific frequency ranges in a given area can be identified with other devices but are usually specialized for specific tasks. These tools can be found as add-on items to notebook computers or amateur radio units.

After a completed survey has identified or located an offending frequency, it is important to determine how the interference can be mitigated to increase the opportunity for a successful and error free deployment.

If the offending frequencies cannot be eliminated, it might require shielding of the readers. The interrogation zone can be shielded by installing an opaque material around the antenna to inhibit stray frequencies from interfering. Another solution is creating an interrogation zone similar to a sound booth. This design would use material that has both absorbent and opaque characteristics to capture and to neutralize stray or interfering RF.

1.1.5 RFID tags

Understanding how an embedded antenna functions in an RFID tag is critical for a successful deployment. The basic and most fundamental problem is ensuring that sufficient power is transmitted to a tag.

There are a wide variety of antenna designs that attempt to provide optimal performance. For passive tag designs, the tag antenna should receive the electromagnetic field that the readers generate at a right angle. Figure 1-13 shows how reader or tag location can impact optimal performance.

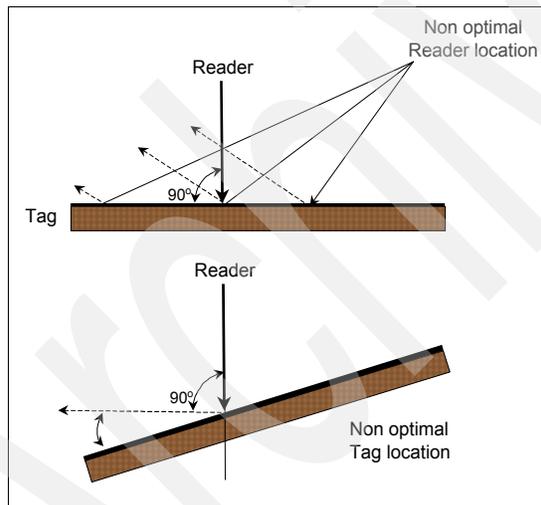
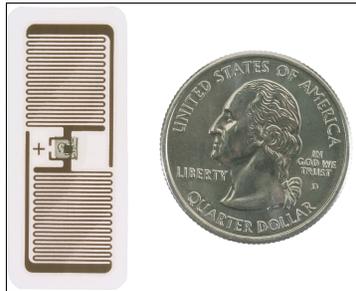


Figure 1-13 Reader location for optimum performance

To compensate for tags that might not be orthogonal (at a right angle) to the wave, some designs have incorporated angles or turn in the dipole antenna (Figure 1-14 on page 16). This design allows the orientation of the tag or tagged component to have greater tolerance and maximizes the coupling characteristics of the tag when passing by a reader.



*Figure 1-14 Intermec mini dipole RFID tag insert
Photograph courtesy of Intermec Technologies*

However, the tag performance is significantly better (power generated and overall distances from the reader) when a straight antenna is used in its proper orientation.

You should address several considerations for tag location on a item when designing a dock door scenario. The optimum functionality for a passive tag is at a right angle to the reader's antenna. This angle can be accomplished by having specific predefined locations for the tag. Alternatives designs could incorporate multiple reader antennas located in such a way that the tagged object cannot be passed through without a successful read.

These tag locations allow the box (case) to be situated in any orientation and result in a successful read (Figure 1-15).

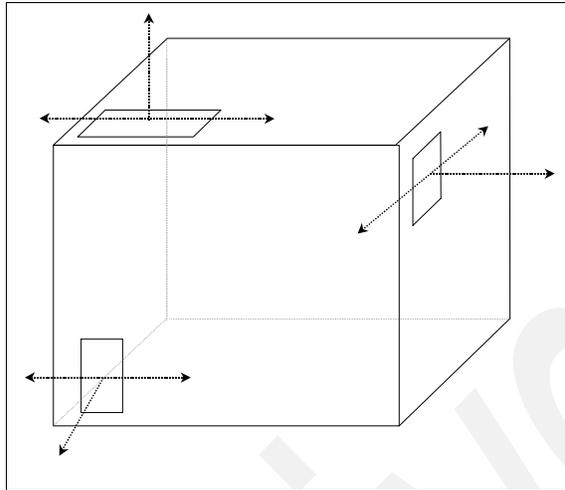


Figure 1-15 Individual case tag locations

In Figure 1-16, the complexity of a successful read is compounded by internal product tags, which are at many orientations and might not be lucent. If these tags were opaque objects, they would affect the external read of the case. If the solution relied on one antenna and that antenna was not able to penetrate the internal objects, the tag could not be read.

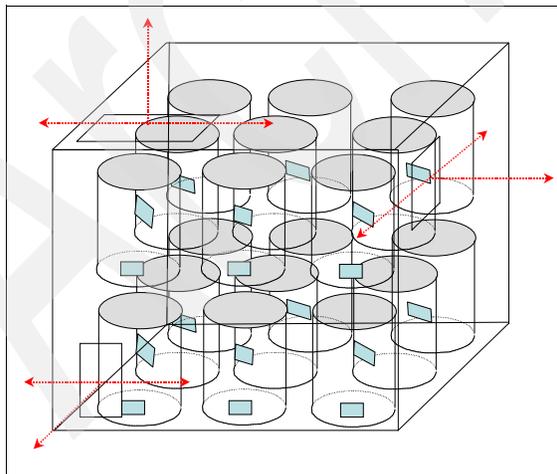


Figure 1-16 Product tags per case

Why stop there? In Figure 1-17, the case that was composed of 18 objects was expanded to a pallet that included 18 shrink-wrapped cases. A successful deployment of a dock door RFID solution could interrogate and identify the contents of each case, indicate how many cases were on the pallet, and identify the individual pallet.

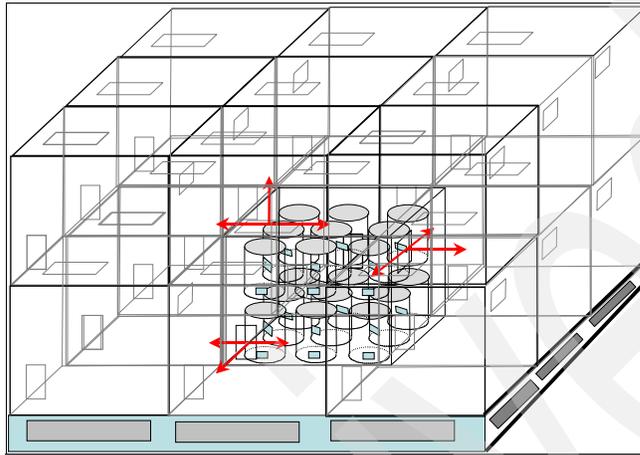


Figure 1-17 Illustration of the complexity for pallet, case, and product tags

Figure 1-18 provides a visual representation of a pallet of cases.



Figure 1-18 Visual representation of a pallet of cases

Dual-dipole is another approach that connects two dipole antennas in different orientations to a chip. This approach increases the chances of a successful read because the second antenna is positioned usually at a right angle to the first.

Antennas can also be tuned to a specific frequency that provides the ability to address different packaged materials, contents, or environmental conditions where tags can be affixed. Depending on the type of target package, penetration or saturation can have different results at a specified frequency.

The absorption of the electromagnetic field by the medium presents additional challenges (Figure 1-19).

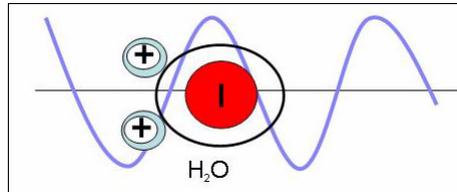


Figure 1-19 Absorption of the electromagnetic field when water is the medium

The absorbing materials reduce the potential power that is available to the transponder. The degradation affects the overall performance and reliability of the tag.

Different types of metals can influence the distribution of the electromagnetic fields. Depending on the properties of the materials in the target zone, different currents or eddies can result in changes in the wave or potentially blocking it altogether.

As depicted in Figure 1-20, materials such as aluminium packaging (cans for example) can present specific challenges to an electromagnetic fields. Design considerations should identify limitations or future expansion requirements that can introduce new requirements.

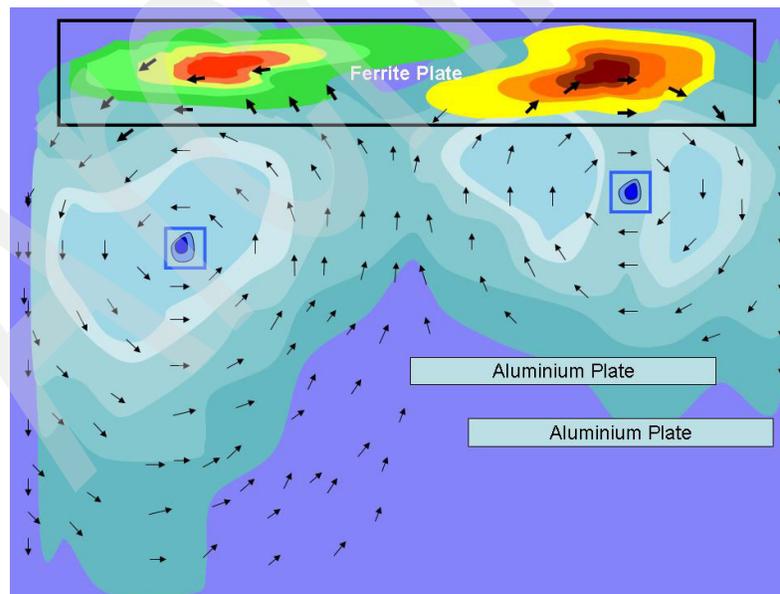


Figure 1-20 Saturation chart with typical characteristics of frequency and samples

1.2 Transponders (RFID tags)

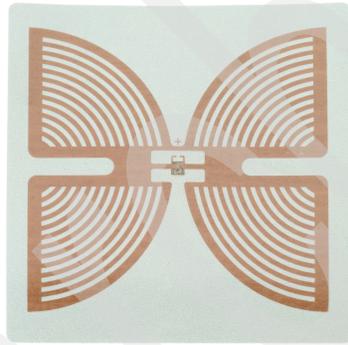
This section contrasts and compares passive and active tag technology.

1.2.1 Passive tag

Passive tags are very simplistic in design; they have no moving parts and no batteries. This type of tag is very well suited for adverse conditions, such as temperature extremes and chemical or corrosive environmental conditions. These tags are typically smaller, less expensive, and have the useful read range of up to 30 feet (approximately 9 meters).

The tag uses the alternating electromagnetic field that is created by the reader antenna coil. It generates a voltage by induction when the electromagnetic field penetrates the cross section of the transponder's antenna coil. This voltage is rectified and acts as the power supply for energizing the microchip and memory in the tag. Then, using load modulation, the transponder transfers encoded data from the tag's memory back to the reader on a modulated UHF wave.

Figure 1-21 shows an example of a passive tag.



*Figure 1-21 Example of a passive tag - Intermec Butterfly RFID tag insert
Photograph courtesy of Intermec Technologies*

1.2.2 Active tag

An active tag usually performs a specialized task. It has an on-board power source (usually a battery) and does not require inductions to provide current as seen in the passive tags. The active tag can be designed with a variety of specialized electronics including microprocessors, different types of sensors, or I/O devices. Depending on the target function of the tag, this information can be processed and stored for immediate or later retrieval by a reader. Active tags in general have an effective reading distance in excess of 100 feet.

To provide a hypothetical example, a new chemical has a specific shelf life. This shelf life is determined by several factors: 1) date manufactured, 2) temperature variation, 3) exposure to light. The micro processors in the active tag can collect information about the environmental conditions and maintain an active log. Depending on the degradation of the new chemical due to the environmental condition, the shelf life can be decreased accordingly. The active tag then transmits the unique identifier along with this updated information to the reader.

There are several types of active tags, all which act differently. One type, referred to as a *transmitter*, takes the lead role and communicates first to the reader. This tag continues to broadcast its data regardless of whether a reader is present.

Another type of active tag is referred to as *transmitter* or *receiver* (transponders). This type of tag enters into a sleep mode (low power state) to conserve resources. The sleep mode is triggered by the lack of interrogation from a reader. Then the tag is awakened from the low power state when the reader broadcasts a specific command.

Other benefits of this type of tag are that they only broadcast if interrogated by the reader. This broadcast reduces the overall RF noise in the environment and conserves the battery life of the tag. Figure 1-22 shows an example of an active tag from RF Code.



Figure 1-22 Example of an active tag - RF Code Mantis tag
Photograph courtesy of RF Code

1.2.3 Semi-active (semi-passive) tag

Semi-active tags or *battery-assisted* tags differ from active tags in several ways. The battery provides energy for the tag's operation or functions and does not transmit to the reader. These tags use the inductive characteristics of a passive tag to generate voltage and to transmit the data to the reader. Because the tags are power assisted by battery, they are capable of reacting faster than a standard passive tag.

Another advantage of battery-assisted tags is in the presence of materials that might inhibit the passive tags from functioning properly. This results in data transmission errors because that tag does not have sufficient power to operate correctly.

Semi-active tags can be read up to approximately 100 feet and a high rate of speed. Ideal conditions use a modulated backscatter scheme with UHF or microwaves.

1.2.4 Active or passive tags data access capabilities

Active or passive tags come with the following data access capabilities.

▶ Read only (RO)

Specific data that is burned permanently onto the microchip during manufacturing stages. Tag manufacturers provide the data for the tag, which is also referred to as *factory programmed*.

▶ Read write (RW)

Depending on the type of deployment scenario, the reader can be used to modify tag data. If this is an active tag, it could use functions within the microchip to collect and to update FLASH or FRAM (Ferroelectric Random Access Memory) memory. This tag is also referred to as *field programmable* or *re-programmable*. Manufacturing is the most expensive of the three types of data access described and poses data security concerns for data integrity. For more information, see the following resources:

- *RFID Essentials*, Print ISBN-13: 978-0-59-600944-1
- *RFID Field Guide: Deploying Radio Frequency Identification Systems*, Print ISBN-13: 978-0-13-185355-3

▶ Write once, read many (WORM)

WORM tags are used widely in the business sector today. They offer the ability for the user to modify or update information rather than the manufacturer of the tag. Compared to the RW tag, they provide a reasonable level of security and a much lower cost per tag.

1.2.5 Surface acoustic wave (SAW) tag

This type of technology is used in cell phones and other consumer electronics. It differs from the microchip-based tag technology because it does not require a direct current (dc) to perform data transmission. Rather it uses low-power RF waves to operate and is in the frequency range of industrial, scientific, and medical (ISM) 2.45 GHz.

The design of the tag uses a dipole antenna that is connected to an inter-digital transducer.

1.2.6 Elements of an RFID tag

The composition of a passive RFID tag is very simplistic in design. The base or substrate layer of the tag is usually a thin plastic film. The antenna and chip are then sandwiched between the substrate and the paper cover (Figure 1-23).

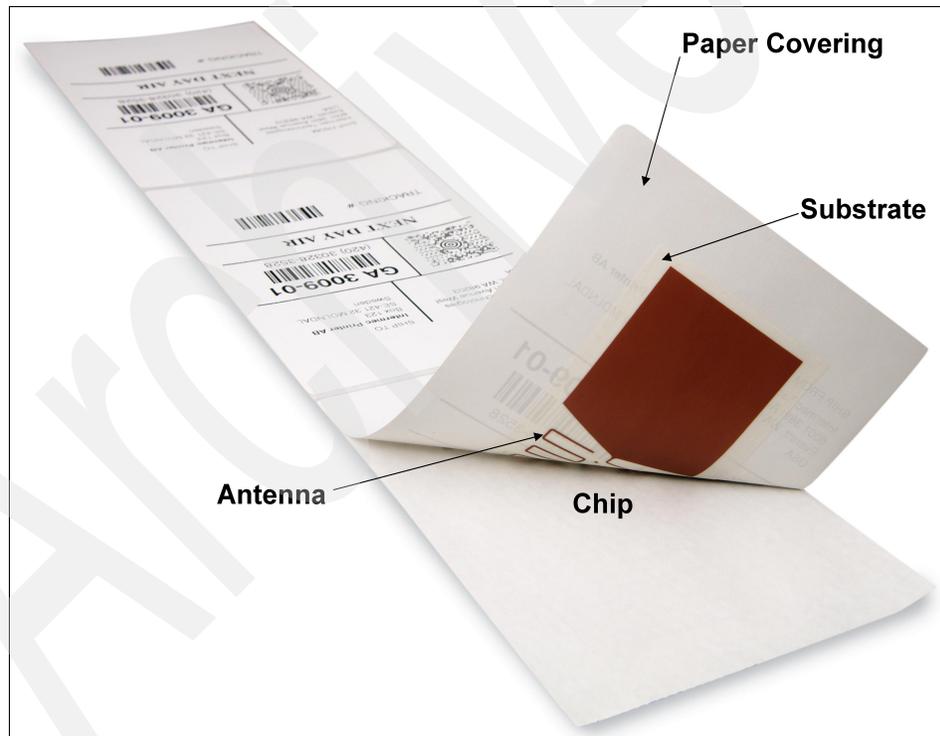


Figure 1-23 Elements of an RFID tag
Photograph courtesy of Intermec Technologies

Figure 1-24 and Figure 1-25 are additional examples of RFID tags.



Figure 1-24 Example of a chip-size RFID tag compared to a U.S. postage stamp
Photograph courtesy of Intermec Technologies



*Figure 1-25 Example of an RFID tag insert
Photograph courtesy of Intermec Technologies*

1.3 Exploring RFID at the IBM Wireless Center of Excellence

During the writing of this book, IBM announced the opening of the Wireless Center of Excellence located on the IBM Research Triangle Park campus in Durham, North Carolina (Figure 1-26). This facility offers potential adopters of RFID technology an opportunity to experience a real-world test environment in a working warehouse. This fully functional, RFID-enabled warehouse provides a demonstration of RFID solutions, capabilities, and emerging technologies while operating as a major distribution center for IBM xSeries® servers.



Figure 1-26 IBM Wireless Center of Excellence RFID testing facility
Mercury4 by ThingMagic reprinted by permission

In addition to the warehouse operations, the center provides a fully integrated lab and testing facility. Customers or IBM Business Partners can use this facility for testing emerging technologies or the latest supported RFID readers, printers, and tags from a large variety of leading manufacturers.

Several dock door scenarios have been constructed that allow clients to use the fully functioning portals. These portals provide opportunities to test or to determine the optimal tag placement or reader combination for a fully loaded pallet (Figure 1-27).



Figure 1-27 IBM dock door RFID portals at the Wireless Center of Excellence Mercury4 by ThingMagic reprinted by permission

Figure 1-28 shows the team with a forklift that is equipped with RFID technology. The forklift has an RFID antenna that is mounted on the front and has an RFID reader and a wireless touch screen mobile PC that is mounted in the cab.



Figure 1-28 The team in the Wireless Center of Excellence

If speed is the name of the game, the facility includes a state-of-the-art conveyor system that uses plastic rollers and nuts and bolts to reduce RF interference is capable of moving product up to 650 feet per minute (Figure 1-29).



Figure 1-29 The Wireless Center of Excellence state-of-the-art conveyor system

The Wireless Center of Excellence provides opportunities for clients to envision the future. Clients can understand how to take advantage of wireless technology strategically and can test manufactured products that they have provided with a large variety of RFID equipment that is available in the lab or warehouse.

While no lab can eliminate completely unforeseen issues in a large warehouse or factory, this center provides an opportunity to simulate a real-world environment and to identify potential obstacles to ensure a successful deployment.

1.4 RFID solution design considerations

When designing your RFID solution, one of the fundamental requirements is conducting a site survey (Figure 1-30). There is no substitute for completing a well-documented survey, which can uncover potential sources of electromagnetic interference when designing the RFID system for the factory, distribution warehouse, or any type of deployment.



*Figure 1-30 Performing an RFID site survey
Photograph courtesy of Intermec Technologies*

The physical obstacles or obstructions can be identifiable and might be viewed on the building layout or blueprints. Although obstructions might cause issues for deployment, electromagnetic or ambient radio frequencies might be present in the environment. This type of interference can play havoc with diagnosing RFID problems. Some frequencies might be generated by extended sources that are not controllable.

Examples of such interference are all around us. Many are considered nuisances or inconveniences that are tolerated temporarily, such as an old lawn mower that uses a gas driven engine and has a magneto. The lawn mower generates RF that causes an electrical interruption and static on neighborhood TVs.

Today, modern electronics have filtered this type of interruption; however, for RFID technology, it can be devastating. Any type of machinery or communication

devices can generate interference. These RF waves are meaningless (they do not contain useful information or data) and act as inhibitors. RF can also become altered when they are bounced or reflected, resulting in a modified signal that can cause interruption.

Similar to a conversation between two individuals, when ambient noise is introduced as chatter (multiple conversations in the same space with no protocol control), the conversation becomes difficult to understand. This same philosophy applies to RFID. If the communication between the reader and the tags cannot be discontinued between ambient noise and the true signal, the RFID delivery system has failed.

For information about the allocated RF frequencies for the United States, visit the National Telecommunications and Information Administration (NTIA) Web site and navigate to the Office of Spectrum Management (OSM) Web page. There, you can find the United States Frequency Allocation Chart. The URL for this Web site is:

<http://www.ntia.doc.gov/osmhome/allochrt.pdf>

From the chart listed on this Web site, RFID uses a very small allocation of these frequency assignments. It is easy to understand how many frequency ranges are in use today. Transmission from noncompliant devices could cause interference for RFID solution if a frequency is altered or distorted.

Identifying frequencies during a site survey by using a spectrum analyzer is a necessity. It provides the groundwork for a successful deployment by identifying stray frequencies that might exist in the environment. It also helps you build topographies or contours of the optimal performance for a planned area of deployment.

A simple site survey might not be sufficient. These invisible obstacles might not always exist. They can be inactive during the day or a certain periods throughout the day. Thus, the site survey should be conducted over a lengthy period, for example several weeks or perhaps even a month or two. You should consider every aspect of day-to-day operation during the site survey. Here is an example: A power outage requires activation of the back-up electrical systems. Was this included in the assessment of the facility? Not knowing what, if any, interference might be created by the generator, can result in the entire RFID system being inoperable.

During the site assessment phase of the project, you should also carefully consider identifying the frequencies spectrum that are required for the deployment. As discussed in "Impact of material characteristics with regard to RF" on page 10, the types of materials and their properties react differently to specific frequencies.

Various factors can influence RFID deployment efforts and should be considered during initial site survey. Figure 1-31 indicates issues that can influence a successful deployment. There are Radio Frequency interference factors that are predictable or unpredictable as well as factors that are controllable or uncontrollable.

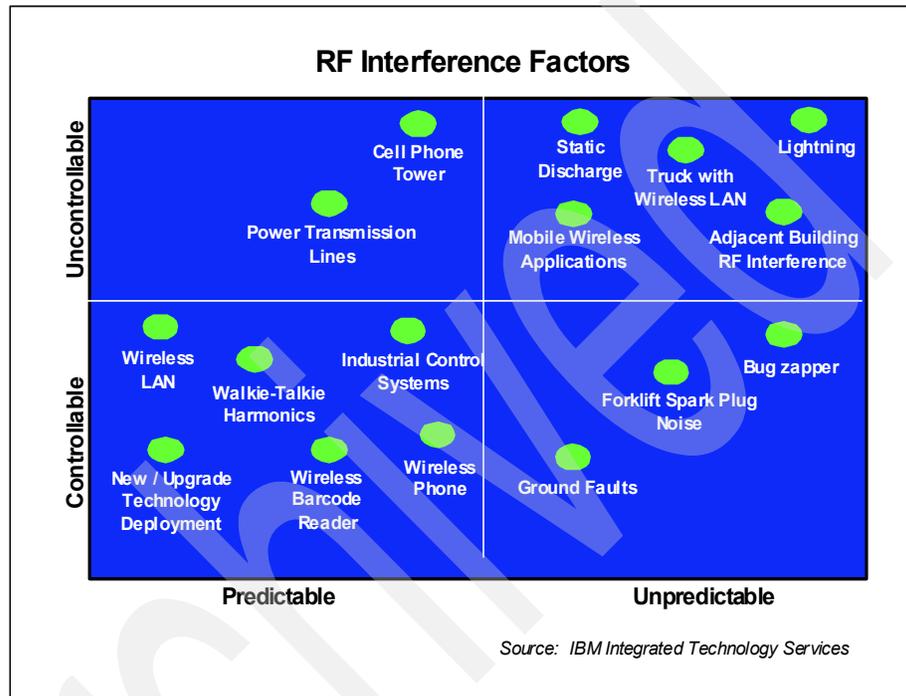


Figure 1-31 Radio Frequency interference factors

Figure 1-32 shows a sample site survey form that you might use to inventory sources of interference.

Sample Site Survey	Identifying the location or zones of operation
Exterior and surrounding area	Property dimensions Physical Building Blueprint Exterior building dimensions Shipping and Receiving (Dock Doors) Trash compactors Power lines Broadcast towers Power Generation units Air conditioning unit Electric Gates
Interior Building	Alarm system Power distribution center Emergency lighting Dividing walls (type) Physical location ref. outer wall Column, row, bay etc Network distribution Intercom systems
Mechanical Equipment	Power lifts Fork Lifts Wrappers Floor sweepers Hand pallet trucks Electric staplers Coffee Pots Label makers Phone system Computers Printers Scanners Wireless Devices Time clocks

Figure 1-32 Sample site survey form

Archived



Introduction to IBM RFID solutions

This chapter discusses how IBM builds RFID solutions. IBM RFID solutions feature IBM software products and technologies and are assembled according to an RFID architectural framework called the *IBM RFID Solution Domain Model*. The WebSphere RFID solution middleware is highlighted as the backbone of IBM RFID solutions. It provides the core infrastructure for connecting RFID devices into enterprise business systems.

2.1 IBM RFID Solution Domain Model

IBM has created an RFID architectural framework to illustrate how complete, end-to-end RFID solutions are modeled. This framework, the RFID Solution Domain Model, groups logically-related RFID solution functions into *component domains*. These component domains enable RFID solutions that support evolving standards, such as EPCglobal and ISO, while insulating technology advancements in one domain from effecting other domains.

Figure 2-1 illustrates this IBM RFID Solution Domain Architecture and includes the domains of Tagged Object, Reader, Edge, Premises, Business Process Integration, Enterprise and Business Application, Systems Management, and Object Directory. Security and Privacy Management issues are also part of the domain model, are distributed among the domains, and are specific to each.

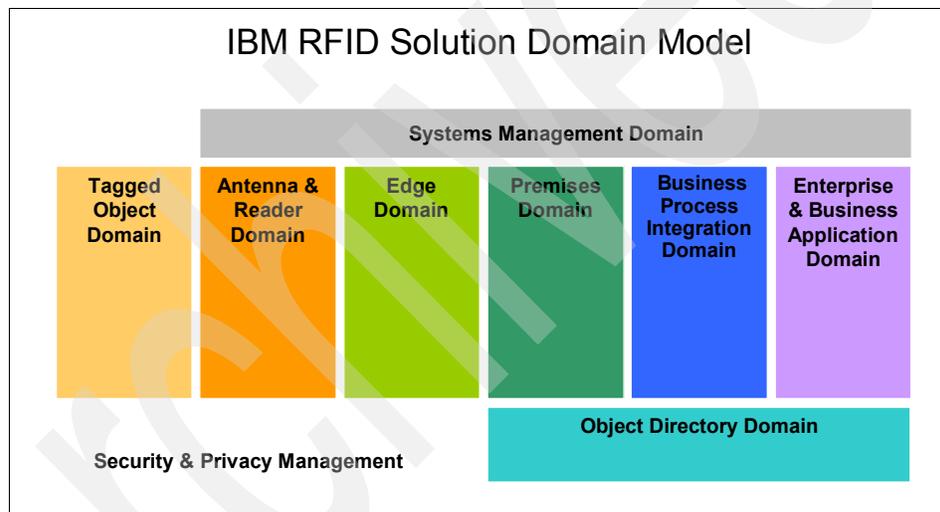


Figure 2-1 The IBM RFID Solution Domain Model

Tagged Object Domain

The Tagged Object Domain contains the tagged products in a supply chain or any other assets or locations that are intended to be tracked or monitored, including the use of sensors on tags. Because object and tag are attached physically, they are considered components of the same domain.

In contrast to other domains, most of the artifacts in the Tagged Object Domain are mobile, that is they can move across different RFID Infrastructures. This restriction imposes strict interoperability requirements on those artifacts that would be addressed ideally through open standards.

Antenna and Reader Device Domain

The Antenna and Reader Domain is the interface between the physical world of objects, tags, radio frequencies, and so forth and the world of software systems. In the future, more sophisticated functionality is expected to move from the Edge Domain down to readers (for example, event filtering), making readers *smarter* and able to handle more tasks programmatically.

Edge Domain

The Edge Domain includes the functionality of filtering and aggregating volumes of data that is provided by the readers, supporting the analysis of data and applying local decision making and intelligence. The architecture of this domain needs to be compatible with readers from multiple vendors and must hide individual reader (and also tag interface) idiosyncrasies effectively from the remainder of the infrastructure.

The Edge Domain functions are implemented typically in a low-cost appliance just upstream of the readers and use embedded software technologies to establish a software stack on the outer edge of the RFID infrastructure. To enable manageability of a production RFID system, deployment aspects are another key aspect that must be addressed. Software updates in the Edge Domain can be deployed automatically.

Premises Domain

The Premises Domain is the intermediary between enterprise applications and the Edge Domain. The Premises Domain filters and aggregates, monitors and escalates RFID events to detect critical business operations, enabling programmatic decision making. It also tracks and logs all important information about products and locations and manages *downstream* components in other domains, such as readers or RFID controllers.

The Premises Domain deals with events *on a higher level* than the Edge Domain, that is events that are important in context of a business operation or process. This domain can store data, and it interacts with enterprise back-end systems through business process integration.

Business logic can be specific for a premises' operational requirements. As an example, a distribution center for food can have different business logic from a distribution center for hardware, although they belong to the same retail chain and are part of the same hierarchy.

Business Process Integration Domain

Business Process Integration Domain connects the RFID Infrastructure to enterprise applications. While other domains provide a reasonable level of functionality as is without modification, Business Process Integration typically

requires customization to match a given enterprise environment. Therefore, this domain is described as a *toolbox* for business integration.

A key feature of the Business Process Integration Domain is its ability to act as a business-to-business hub for automatic transactions between trading partners. As the rest of the RFID Infrastructure notices significant product movement (for example, a shipment), this domain formats and sends a message (for example, an advanced ship notice) to the appropriate trading partner.

The Business Process Integration Domain not only connects to existing applications but can also enable new ways of doing and managing business through Business Process Transformation and Business Performance Management involving RFID-related business analytics and dashboards for monitoring key performance indicators.

Enterprise and Business Applications Domain

The Enterprise and Business Application Domain include the existing applications that require information about product movement that is captured by the RFID infrastructure.

These applications correspond to an organization's unique mix of business requirements. This domain includes systems that help ordering, managing, or supplying goods and that can be enhanced greatly by being able to monitor product movement automatically. Examples of these types of systems are Process Automation, Inventory Management, Enterprise Resource Planning (ERP), Manufacturing Execution Systems, Supply Chain Execution Systems, Warehouse Management Systems, Data Warehouse, Merchandise Management, Store Systems, Work In Process Manufacturing, and so on.

Object Directory Domain

The Object Directory Domain provides information about the physical object that is using its unique ID as the lookup key. It enables the rapid retrieval of product information and also provides a framework for allowing companies to securely share product information with trading partners. The EPCglobal Network consists of the following:

- ▶ Object Naming Service (ONS)
- ▶ Electronic Product Code Discovery Service (EPCDS)
- ▶ Electronic Product Code Information Service (EPCIS)

The EPCglobal Network is one example of an Object Directory Domain.

This domain typically delivers three kinds of information about a product:

- ▶ Core product information
- ▶ Manufacturing time information
- ▶ Life cycle history information

Core product information is usually obtained from product information managers, product catalogs, or data pools.

Systems Management Domain

The Systems Management Domain allows customers to get the RFID systems up and keep them running. It allows you to deploy and manage applications remotely in a distributed environment, including the ability to monitor, configure, or update software and firmware remotely in deployed assets such as antennas, readers, and servers. This domain can include a central dashboard through which it is possible to monitor assets and receive alerts when readers, antennas, and servers break down. Such dashboards help to increase reliability and reduce operations costs. This domain can also deliver operator and user consoles when remote guidance is required.

Security and Privacy Management

As the RFID becomes an integral part of an enterprise-wide data management system, IT systems must be resistant to security and privacy breaches. Effective Security and Privacy Management allows customers to extend the existing security infrastructure to the reader level through to the Electronic Product Code (EPC) Network. The infrastructure must protect stored data as well as data that is in transit.

2.2 Implementing IBM RFID solutions

IBM RFID solutions are implemented using IBM software products, IBM Global Services consulting and services, and supported products and solution components from IBM RFID industry Business Partners.

2.2.1 IBM software

IBM has a broad set of software products and technologies that are relevant to RFID solutions. Some software, such as the WebSphere RFID solution middleware, is designed specifically for RFID solutions. Other IBM software is used to support the development, integration, or management of the RFID solution.

WebSphere RFID solution

IBM WebSphere RFID solution is middleware that provides the core infrastructure and backbone of RFID solutions. It connects and integrates the RFID tags and devices with the business logic and enterprise information systems. The WebSphere RFID solution consists of three components:

- ▶ WebSphere RFID Premises Server
- ▶ WebSphere RFID Device Infrastructure
- ▶ WebSphere Business Integration Server

Atop this WebSphere RFID solution middleware is the custom business logic that embodies the operational requirements for a particular RFID usage scenario. Figure 2-2 illustrates how the IBM RFID Solution Domain maps to WebSphere RFID middleware and best practices applications.

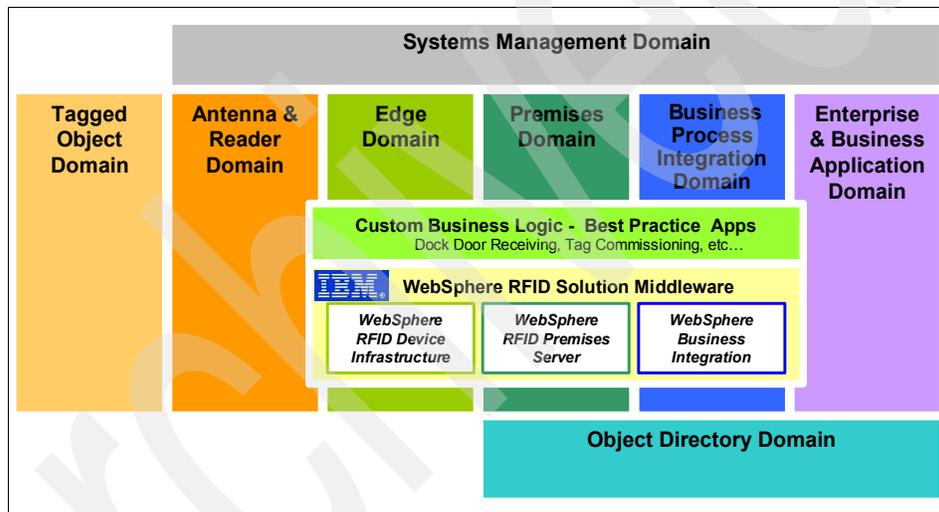


Figure 2-2 Mapping the IBM RFID Solution Domain

WebSphere RFID Premises Server

The WebSphere RFID Premises Server is an IBM software product that is the centerpiece of the IBM RFID solution. It serves as an intelligent intermediary and interpreter between the physical world of RFID devices using the Edge domain and the IT world using the Business Integration domain. The Premises Server is J2EE-based open-standards middleware. It enables RFID solutions to be highly customized and optimized to specific business requirements.

WebSphere RFID Device Infrastructure

The WebSphere RFID Device Infrastructure is a set of associated software technologies that are licensed to manufacturers of RFID programmable controller devices, such as controllers and intelligent tag readers. These devices are generically called *edge controllers* and, in the context of this book, are assumed to be equipped with the WebSphere RFID Device Infrastructure.

The WebSphere RFID Device Infrastructure is Open Services Gateway initiative (OSGi) and J2ME-based open-standards middleware, again enabling the RFID solutions to be highly customized and optimized to specific business requirements.

WebSphere Business Integration Server

The WebSphere Business Integration Server generally represents how the IBM RFID solutions connect the RFID-specific world of the Premises Servers and edge controllers into an enterprise information system. The WebSphere Business Integration family of products include the following:

- ▶ WebSphere Business Integration Server
- ▶ Server Foundation
- ▶ WebSphere Business Integration Express
- ▶ WebSphere Application Server Express

The WebSphere Business Integration family of products are designed to enable enterprises to integrate Web applications with existing business processes and applications.

Custom business logic and best practices applications

Customer business logic and best practices applications represents the software that you must implement to drive the WebSphere RFID solution middleware to meet your operational use and business integration requirements.

Your operational use case requirements are often common across an industry. Therefore, you can benefit by implementing RFID solutions from pre-built applications that embody industry best practices for particular use cases. For example, the IBM RFID Premises Server includes a pre-built dock door receiving sample application that you can use to jump start WebSphere RFID solution implementations.

Supporting IBM software

In addition to the WebSphere RFID solution software components, IBM has other software products that are useful in implementing RFID solutions. Figure 2-3 maps this supporting software to the IBM RFID Solution Domain. This set of supporting software includes the system management software and integrated development tooling that are required for RFID solution development and deployment.

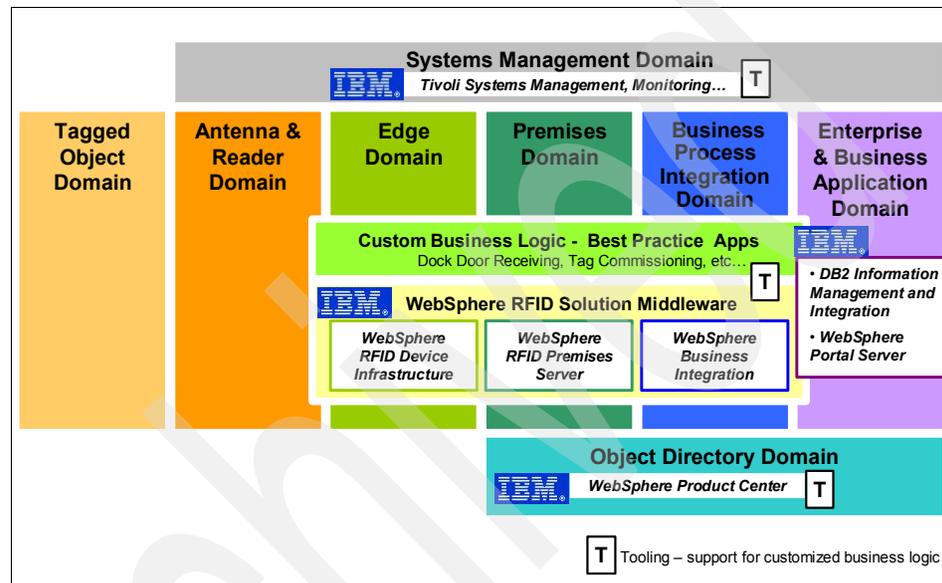


Figure 2-3 Software and tooling for complete RFID solution implementations

IBM DB2 and DB2 Information Integrator

IBM DB2 and DB2 Information Integrator products work with the integration of application servers to help you collect, manage, and to take advantage of enterprise data.

IBM Tivoli monitoring and management products

If you are deploying RFID solutions, you will want to monitor and manage the mission-critical data that flows from the edge of your network to the enterprise. You can use the industry-leading IBM Tivoli® monitoring and management products in conjunction with the integration of an application server in a distributed environment.

WebSphere Product Center

The WebSphere Product Center is a product information management solution that helps companies assemble an accurate, consistent, and central repository. It

facilitates the sharing of product and service information across countless customer, partner, and employee touch points, and it synchronizes information internally with existing enterprise systems and externally with business and trading partners. The WebSphere Product Center is a key component for streamlining supply chain and partner management. You can access the WebSphere Product Center at:

<http://www-306.ibm.com/software/integration/wpc/>

2.2.2 IBM Integration and Consulting Services

The IBM Global Services RFID consulting and services team are experts in developing solutions using the WebSphere RFID solution middleware. They build highly optimized RFID business logic and integrate the solution into your existing business environment. Further, IBM Global Services provides total life cycle RFID solution management for customers — from RFID project consulting, management, and development through deployment and ongoing support. IBM has the expertise to grow RFID solutions from small, narrowly-scoped pilot implementations to large, global deployments.

For small and medium businesses, IBM Global Services also offers Express RFID Services, an affordable *slap and ship* managed service that is designed to enable rapid compliance with RFID tagging mandates. This solution is designed to enable mid-market manufacturers and other suppliers to respond rapidly to the RFID tagging mandates that are issued by the U.S. Department of Defense (DoD), WalMart, Target, and other agencies and retailers. A solution can be deployed in as little as two to three weeks and, unlike competitive offerings, can include remote management and help desk capabilities by IBM.

IBM is a leader in RFID integration and consulting services. In 2004, IBM created a dedicated Sensors and Actuators solutions group, accompanied by a \$250 million (USD) investment in a wide range of RFID technology and services. IBM now has more than one thousand RFID professionals, as well as research and test centers that are located in the United States, LaGaude, France, and Yamato, Japan.

At the Wireless Center of Excellence in Research Triangle Park, North Carolina, for example, our clients can receive hands-on training. In addition, they can test equipment, design networks, and simulate production lines in a working warehouse.

2.2.3 IBM Business Partners

IBM Business Partners work with RFID industry leaders to deliver end-to-end solutions that are based on WebSphere RFID solution middleware. For example,

IBM partners with companies such as Arcom to embed the WebSphere RFID Device Infrastructure into RFID controller and intelligent RFID reader devices. IBM also partners with RFID application providers, such as RFID consumer products and retail software leader OATSystems, Inc., to deliver industry-leading applications and solutions using WebSphere RFID Premises Server.

Figure 2-4 illustrates the roles of IBM RFID Consulting and Services and IBM Business Partners.

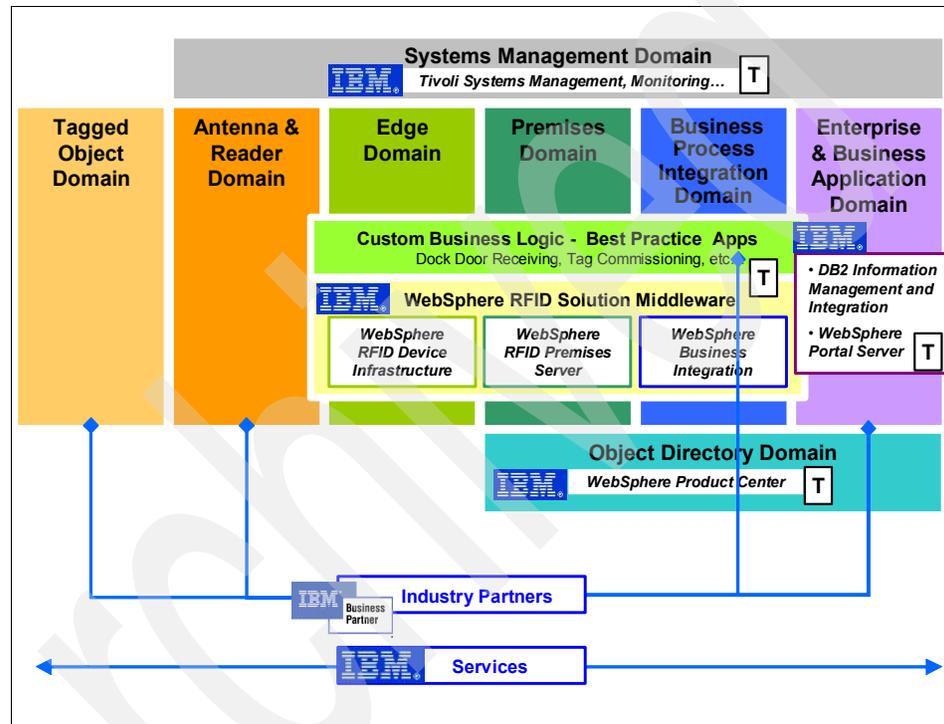


Figure 2-4 IBM RFID Consulting and Services and IBM Business Partners roles

2.3 Introduction to the WebSphere RFID solution

The WebSphere RFID solution, also called *WebSphere RFID*, is a set of IBM WebSphere software products and technologies that are designed specifically to implement the RFID infrastructure for the Edge, Premises, and Business Integration domains, respectively, of the IBM RFID Solution Architecture (Figure 2-5).

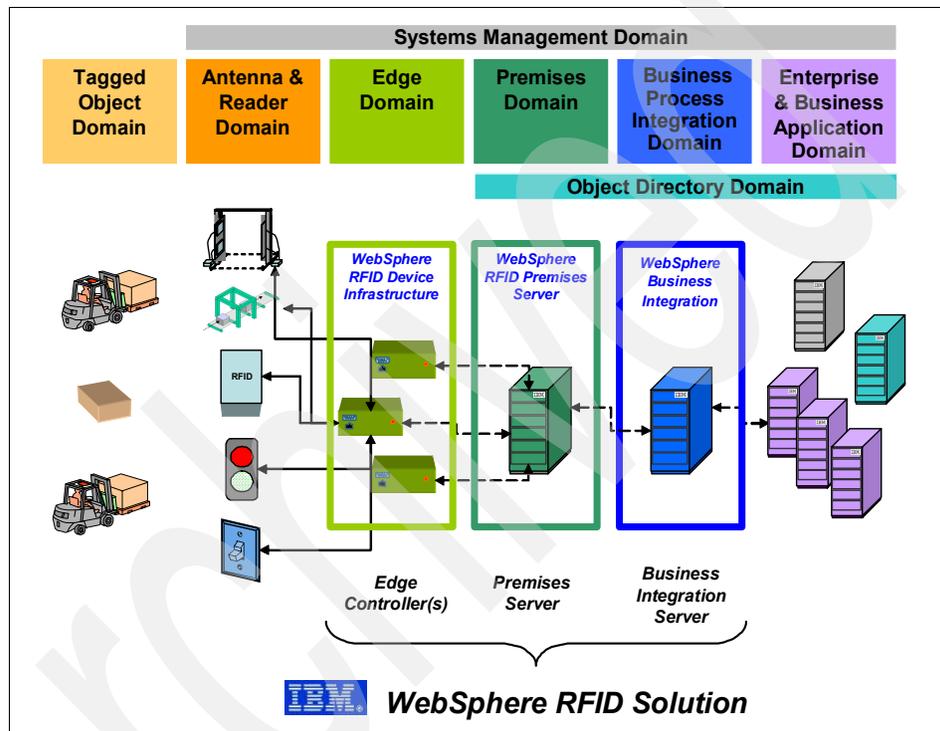


Figure 2-5 WebSphere RFID solution implementation

A WebSphere RFID solution consists of the following:

- ▶ WebSphere RFID Premises Server
- ▶ Edge Controller(s) embedded with WebSphere RFID Device Infrastructure
- ▶ Business Integration Server

Figure 2-6 shows the components and functions of the WebSphere RFID solution.

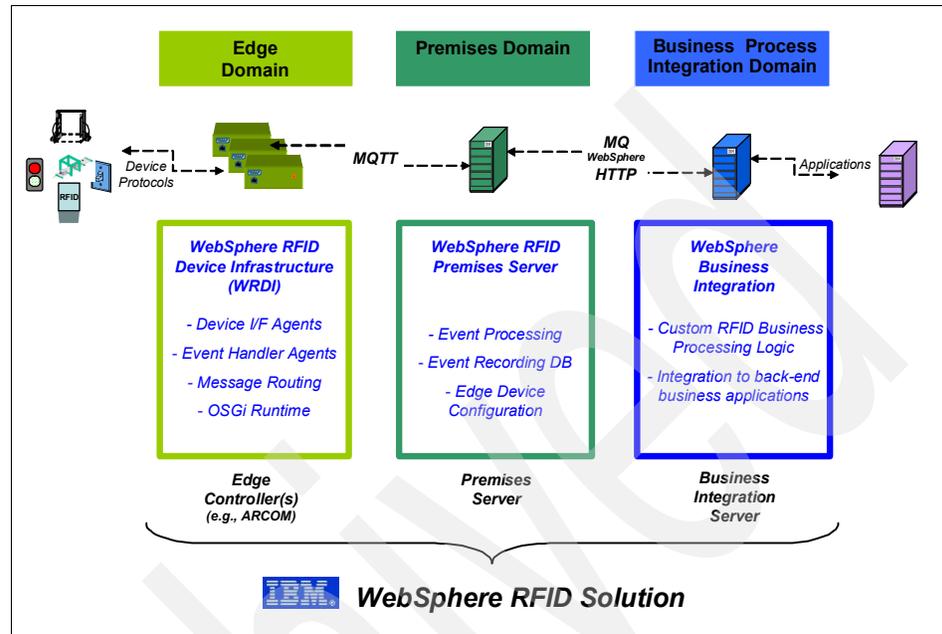


Figure 2-6 WebSphere RFID solution: components and functions

Additionally, WebSphere RFID solution implementers must develop custom application software that embodies the operational and business integration for specific customer requirements. This custom software development includes code to:

- ▶ Process RFID events from the Edge Controller, optionally writing code for the Edge Controller to modify its event processing and device controller logic.
- ▶ Implement premises-specific business logic to interpret RFID event information in the context of the business requirements and to respond appropriately.
- ▶ Collect and correlate RFID data for reports and to integrate the information into back-end business processes and systems.

This custom software should be developed by software professionals who are skilled in IBM RFID solution middleware development, such as the IBM Global Services RFID consulting and services team or an authorized IBM RFID Business Partner.

The WebSphere RFID solution includes a pre-built Dock Door Receiving sample application that demonstrates how you can use the WebSphere RFID

Infrastructure. You can also use the application as a foundation from which the implementers can customize an application quickly to meet production needs.

2.3.1 WebSphere RFID Premises Server

The WebSphere RFID Premises Server is a J2EE-based platform for processing RFID information and events from RFID readers, controllers, and automation equipment of the Edge Domain of the IBM RFID solutions architecture. The Premises Server also provides the platform for integrating these RFID processes into back-end systems, interfacing into the Business Processing Integration Domain and business applications such as ERP, Warehouse Management, and Supply Chain Management.

The Premises Server implements the infrastructure for the Premises Domain of the IBM RFID Solution Architecture and provides the following functions:

- ▶ Processes RFID events according to use case and custom business requirements.
- ▶ Creates and maintains a database of RFID events that occur on a given premises.
- ▶ Administers configuration information, including information for Edge controllers.
- ▶ Supports communication of RFID event and with external applications through standard protocols and interfaces.

The Premises Server is capable of interpreting and correlating high volumes of data from RFID devices that are connected to the server to gain instant visibility of RFID tagged pallets and products. Information from that location can be integrated with the enterprise and shared with the worldwide supply chain to deliver business insight within the enterprise and to partners and customers.

You can purchase the WebSphere RFID Premises Server as a separate bundled product that includes specially priced and licensed WebSphere, DB2, MQ, and Tivoli software, or you can purchase it as an add-on feature to the WebSphere Remote Server for Retail.

For more information about the WebSphere RFID Premises Server, see 3.1, “WebSphere RFID Premises Server overview” on page 54.

2.3.2 WebSphere RFID Device Infrastructure

WebSphere RFID Device Infrastructure is a set of IBM technologies that support the basic functions for RFID event collection and reporting to IBM WebSphere RFID Premises Servers. WebSphere RFID Device Infrastructure implements the function for the Edge Domain of the IBM RFID Solution Architecture and provides the following functions:

- ▶ Interfaces to variety of RFID devices using abstracted device software agents and isolates upstream domains from unique device interface characteristics.
- ▶ Filters and aggregates RFID event data, eliminating duplicate data and identifying RFID events, thus reducing network traffic.
- ▶ Delivers RFID event messages to the IBM WebSphere RFID Premises Server using a messaging and event buffer.
- ▶ Provides software distribution management.

These capabilities are licensed to and delivered by IBM RFID Business Partners in RFID device products generically called *Edge Controllers*. An Edge Controller is a network node that controls a set of I/O devices. Edge Controllers manage I/O devices and their event processing, filter tag information, and send tag information to the Premises Server. Today, IBM WebSphere RFID Device Infrastructure is supported by the Arcom Viper, as shown in Figure 2-7.



Figure 2-7 Arcom Viper Edge Controller
Photograph courtesy of Arcom

For a complete list of supported devices, see A.1, “IBM WebSphere RFID V1.0.2 matrix” on page 246. For more information about the WebSphere RFID Device Infrastructure, see Chapter 4, “WebSphere RFID Device Infrastructure” on page 73.

2.3.3 IBM Business Integration Server

The function of the IBM Business Integration Server is generally undefined in the context of the generic WebSphere RFID solution but presumed to be custom-developed business logic. This custom business logic interfaces with the WebSphere RFID Premises Server and implements custom operational logic, such as RFID event handling, as well as other Premises Domain functions that are specific to the solution and any integration with back-end business systems. Further, the IBM Business Integration Server is presumed to use the IBM WebSphere family of application and integration middleware products and tooling.

2.3.4 WebSphere RFID Dock Door Receiving Starter Kit (Kimono)

IBM provides a pre-built RFID Dock Door Receiving Starter Kit application that embodies industry best practices to speed customer RFID implementations. This sample application is also referred to as *Kimono*. You can install this use case and add your own custom logic to integrate the Dock Door Receiving logic into your business systems.

Also, a Print, Verify, and Ship (PVS) Starter Kit sample use case is now available as an unsupported Technology Preview. This kit helps suppliers with the RFID necessary to print labels for goods, to verify that the correct goods are part of the correct shipments, and then to track their shipment. For more information about PVS, contact your IBM representative.

2.4 A peek inside the WebSphere RFID solution

Figure 2-8 shows the WebSphere RFID solution in detail.

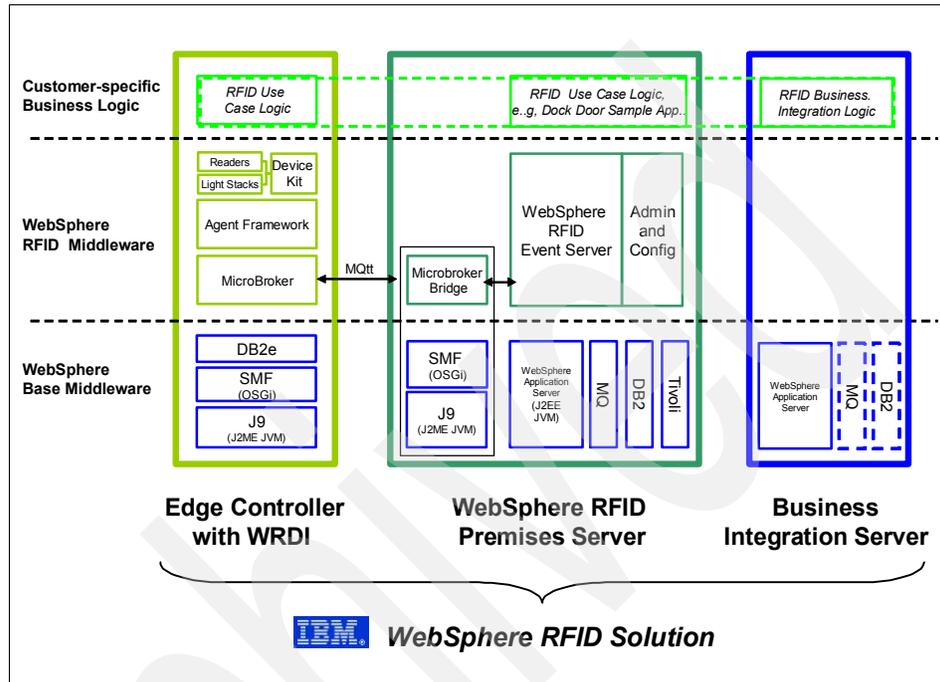


Figure 2-8 A peek inside the WebSphere RFID Solution

The operation of the WebSphere RFID Premises Server is discussed in Chapter 3, “IBM WebSphere RFID Premises Server” on page 53. These internal components and the operation of the WebSphere RFID Device Infrastructure-based Edge Controllers is discussed in Chapter 4, “WebSphere RFID Device Infrastructure” on page 73. We do not discuss the internal components of the IBM Business Integration Server, because they are undefined in a generic solution. Note, however, that the IBM Business Integration Server is presumed to take advantage of the IBM WebSphere family of application and integration middleware (and tooling).

Figure 2-9 shows a sample, high-level flow of event information through the various software components of a WebSphere RFID solution that implements the Dock Door Receiving Starter Kit sample application, Kimono.

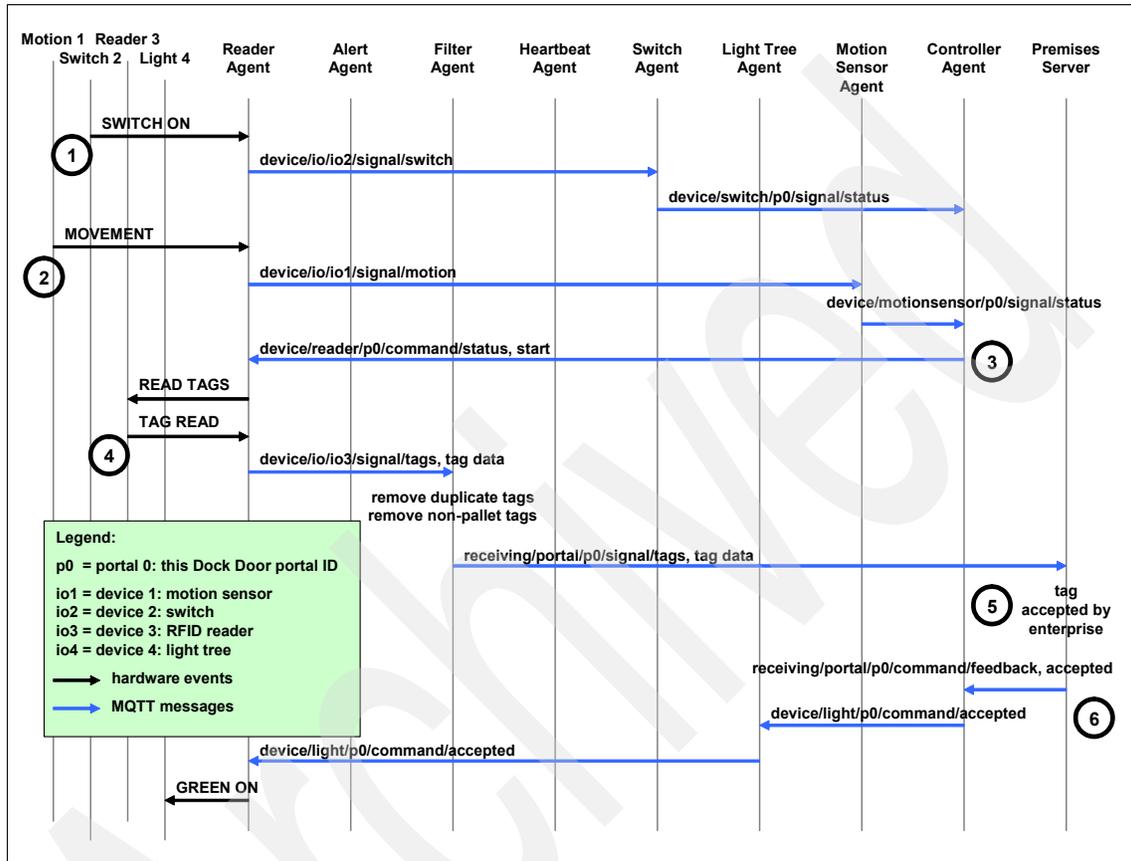


Figure 2-9 WebSphere RFID solution: dock door example flow

Archived



IBM WebSphere RFID Premises Server

This chapter introduces the product concepts and architecture of the IBM WebSphere RFID Premises Server. The WebSphere RFID Premises Server is the centerpiece of the IBM RFID solution, serving as an intelligent intermediary between the world of physical objects and devices of the Edge Domain and the world of abstracted RFID business events of the Business Process Integration Domain.

3.1 WebSphere RFID Premises Server overview

The IBM WebSphere RFID Premises Server is an application based in WebSphere that performs the functions of the Premises Domain within the IBM RFID Solution Architecture (Figure 3-1). The Premises Server processes RFID information and events from the RFID readers, controllers, and automation equipment of the Edge Domain and provides access to RFID information to the Business Processing Integration Domain.

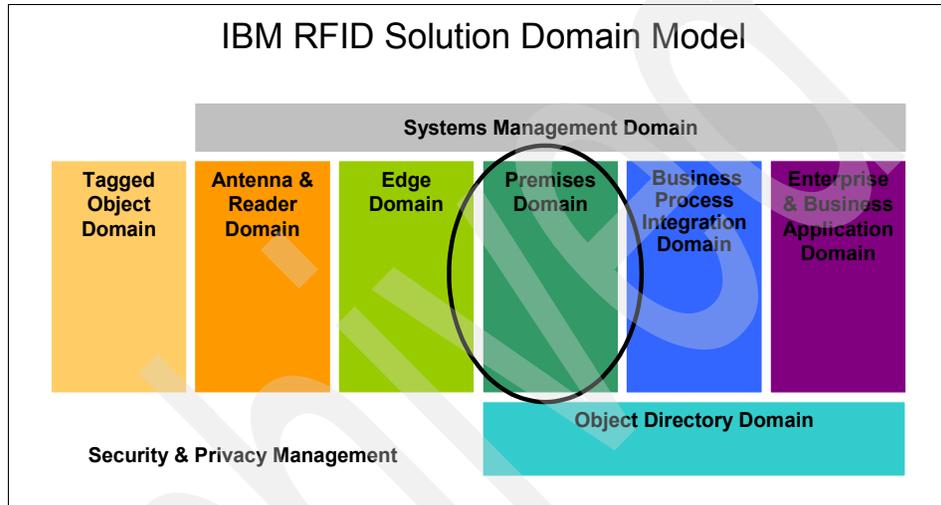


Figure 3-1 The Premises Server in context of IBM RFID Solution Domain Model

The Premises Server is capable of interpreting and correlating high volumes of data from RFID devices that are connected through Edge Controllers to gain real-time visibility to RFID tagged pallets and products. Information from that location can be integrated with the enterprise and shared with the worldwide supply chain to deliver business insight within the enterprise and to partners and customers.

For example, viewing sales data in real-time allows retail managers to evaluate instantly how products are selling. The retailer can then respond immediately by pushing an on-the-fly advertising message that highlights the product to digital media displays to drive new purchases, and then updating the price files in the Point-of-Sales (POS) systems. A real-time view of sales data can also keep shelves stocked, resulting in higher customer satisfaction and increased sales.

The WebSphere RFID Premises Server serves as the intelligent intermediary and interpreter between the physical world of RFID devices using WebSphere RFID Device Infrastructure based Edge Controllers and the world of IT applications using a Business Integration Server (Figure 3-2).

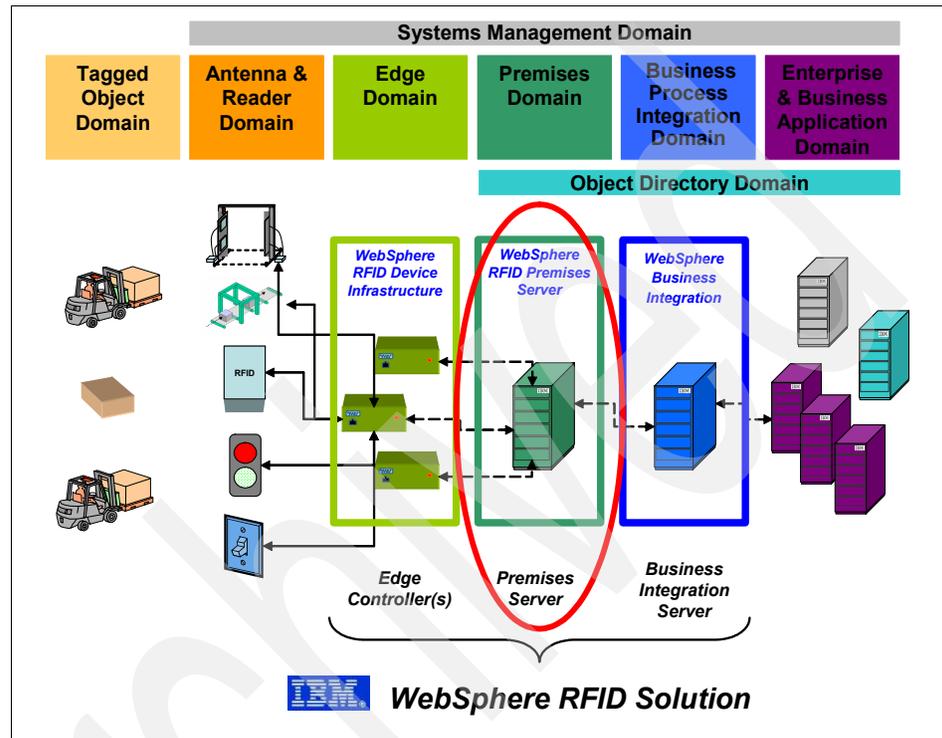


Figure 3-2 Premises Server in the context of RFID solution architecture

As a WebSphere family product, the WebSphere RFID Premises Server is J2EE-based open-standards middleware. It enables RFID solutions to be highly customized and optimized to specific business requirements. It connects RFID event information and processing logic to other enterprise applications in back-end systems using open-standard techniques and protocols, such as Web Services, RMI/IIOP, JMS, and JDBC™.

3.1.1 Key features

The key features of the WebSphere RFID Premises Server are the following:

- ▶ Event processing

The WebSphere RFID Premises Server collects, records, and manages the flow of RFID event information coming from RFID Edge Controllers. The RFID event data is handled in ways that are specific to a particular RFID use case scenario and in accordance with an enterprise's custom business application needs.

So, not only does the Premises Server provide standard processing flow for RFID event information, it also provides an execution environment where premises-specific business logic can be applied to suit operational and business integration requirements. In fact, you can develop multiple RFID-enabled applications with unique business logic for a specific location or type of business. For example, a distribution center for food might have different business logic from a distribution center for hardware, although they belong to the same retail chain.

- ▶ Event record database

The WebSphere RFID Premises Server provides a robust and reliable system for the delivery and storage of RFID event information, creating a persistent RFID record database for all RFID events that are handled by the server. All RFID events that have been observed on the premises can be stored in a DB2 relational database, which can then be accessed from (or delivered to) other business logic for further processing and integration into back-end applications.

- ▶ Edge Controller configuration

The WebSphere RFID Premises Server provides an administration and configuration console Web application where administrators configure the Edge network device topology. The Edge Controller devices retrieve configuration information at startup from XML files that are stored and administered at the Premises Server.

3.1.2 Premises Server software bundle

You can purchase the WebSphere RFID Premises Server as a separate bundled product that includes specially priced and licensed WebSphere, DB2, MQ, and Tivoli software, or it can be purchased as an add-on feature to the WebSphere Remote Server for Retail. In the latter case, the WebSphere RFID V1.0.2 is packaged separately and requires WebSphere Remote Server for Retail.

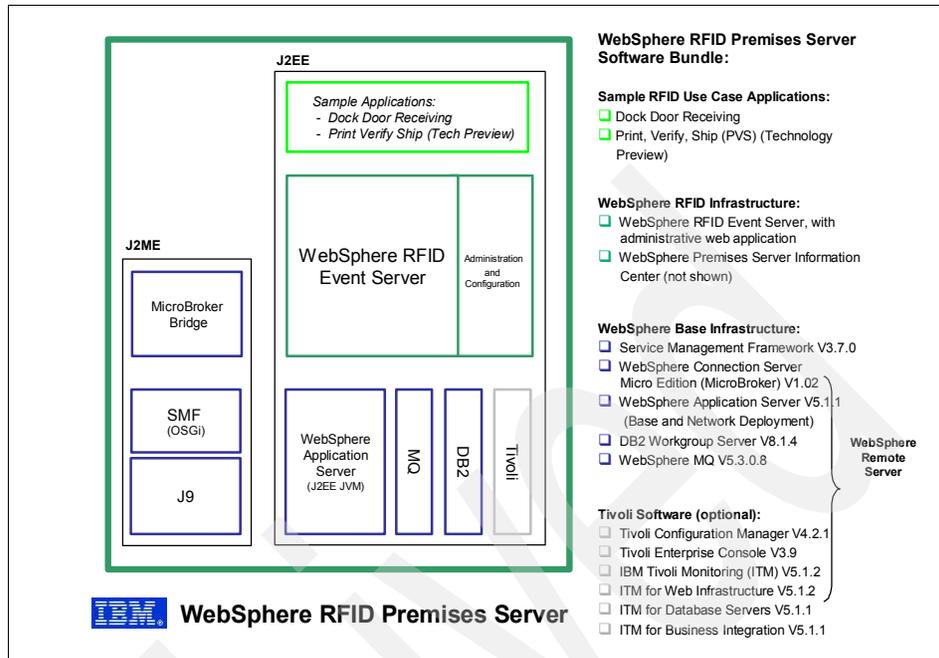


Figure 3-3 WebSphere RFID Premises Server software bundle

The RFID Premises Server product bundle includes:

- ▶ WebSphere RFID Premises Server Software V1.0.2
 - WebSphere RFID Event Server
 - WebSphere RFID Solution InfoCenter
 - WebSphere RFID Dock Door Receiving Starter Kit
- ▶ Prerequisite base WebSphere Remote Server for Retail software components:
 - IBM WebSphere Application Server V5.1.1 and WebSphere Application Server Network Deployment V5.1.1
 - IBM DB2™ Workgroup Server V8.1.4
 - IBM WebSphere MQ V5.3.0.8
 - IBM WebSphere Connection Server Micro Edition (MicroBroker) V1.02
 - Service Management Framework V3.7.0
 - Tivoli Management Software (all optional):
 - IBM Tivoli Configuration Manager V4.2.1
 - IBM Tivoli Enterprise™ Console V3.9
 - IBM Tivoli Monitoring V5.1.2
 - IBM Tivoli Monitoring for Web Infrastructure V5.1.2
 - IBM Tivoli Monitoring for Database Servers V5.1.1
 - IBM Tivoli Monitoring for Business Integration V5.1.1

Note: The software that is bundled in the WebSphere RFID Premises Server is specially priced and licensed for use in IBM RFID solutions. You must acquire a separate license for use of any of these software products outside of the RFID solution environment.

For more information about purchasing WebSphere RFID Premises Server software, visit:

http://www.ibm.com/software/pervasive/ws_rfid_premises_server

RFID Event Server

The RFID Event Server is the primary component of the WebSphere RFID Premises Server. It provides all of the functions that are associated generally with the Premises Domain of the IBM RFID Solution Architecture.

The RFID Event Server processes RFID event information using a system of WebSphere MQ message queues, message queue handlers, and event processing logic. You can customize the event processing logic, called *event tasks*, to provide the desired RFID solution flow according to operational and integration requirements. The RFID Event Server also records the RFID event information that is associated with the Premises Server into an IBM DB2 V8.1 or Oracle 9i relational database.

RFID Solution Information Center

The WebSphere RFID Premises Server CD set includes the WebSphere RFID Solution Information Center V1.02 online documentation about the setup and use of the WebSphere RFID Premises Server V1.02 and related components.

RFID Dock Door Receiving Starter Kit

IBM provides a pre-built RFID Dock Door Receiving Starter Kit sample application that embodies industry best practices to speed customer RFID implementations. You can install this use case and add your own custom logic to integrate the Dock Door Receiving logic into your business systems.

Also, a Print, Verify, Ship (PVS) Starter Kit sample application is available as an unsupported Technology Preview. This starter kit helps suppliers with the RFID necessary to print labels for goods, verify that the correct goods are part of the correct shipments, and then track their shipment. For more information about this Starter Kit, contact your IBM representative.

WebSphere Application Server V5.1.1

The WebSphere Application Server family of interoperable products provides a next-generation application server on an industry-standard foundation. The IBM

WebSphere Application Server family includes several packages with varying degrees of function. The WebSphere Application Server, Base Server edition, is included with WebSphere RFID Premises Server. This edition is designed for standard programming and run-time needs of single-server production environments. Although the administration presumes a single-server environment with no clustering for failover or workload balancing, these base stand-alone nodes can be added to a centrally administered network at a later time using WebSphere Application Server Network Deployment.

WebSphere Application Server Network Deployment V5.1.1

WebSphere Application Server Network Deployment is also included with the WebSphere Premises Server and addresses multiple-server production environments. It provides centralized administration, as well as basic clustering and caching support for load balancing and failover. In addition to all of the features and functions within the base WebSphere Application Server, this edition delivers advanced deployment services such as:

- ▶ Universal Description, Discovery, and Integration (UDDI) V3, which enable you to describe and to discover Web services in a more secure manner as well as to deliver advanced Web services Security to enhance the security of Web services interaction.
- ▶ Web Services Gateway, which enables Web services invocation by users from outside the firewall with the benefit of robust security protection.
- ▶ Advanced failover and clustering capabilities, including failure bypass, load balancing, caching, and centralized security.
- ▶ Simplified, browser-based administration for remote management.
- ▶ Intelligent workload distribution across a cluster.

DB2 WorkGroup Server Edition V8.1.4

IBM DB2 Universal Database™ WorkGroup Server Edition is a multiuser version of DB2 Universal Database that allows you to create and manage single partition or multiple partition database environments. Partitioned database systems can manage high volumes of data and provide benefits such as high availability and increased performance. For more information, see:

<http://www.ibm.com/software/data/db2/udb>

WebSphere MQ V5.3.0.8

IBM WebSphere MQ provides assured once-only delivery of messages across more than 35 industry platforms using a variety of communications protocols. WebSphere MQ supports a variety of application programming interfaces (including MQI, AMI, and JMS), which provide support for several programming languages as well as point-to-point and publish/subscribe communication

models. In addition to support for application programming, WebSphere MQ provides several connectors and gateways to a variety of other products, such as Microsoft® Exchange, Lotus® Domino®, SAP/R3, CICS®, and IMS™.

WebSphere MQ provides support for delivering XML documents and Simple Object Access Protocol (SOAP) messages. It connects applications using Web Services and provides support for the Java Message Service (JMS) interface standard. It offers security using the Internet standard Secure Sockets Layer (SSL).

For more information, see the IBM WebSphere MQ Web site at:

<http://www.ibm.com/software/integration/wmq>

WebSphere Connection Server Micro Edition (MicroBroker) V1.0

The WebSphere Connection Server Micro Edition, also called MicroBroker, is a very small footprint, 100% Java message broker that is capable of running in resource-constrained environments. It is suitable for embedding in applications and solutions that have a need for messaging, notification and event services. MicroBroker supports the publish and subscribe messaging paradigm. It provides a messaging infrastructure, which enables lightweight messaging clients to communicate with each other, on one host or across a network, as well as with enterprise brokers through its bridging capabilities. MicroBroker uses the MQ Telemetry Transport (MQTT) protocol over TCP/IP.

The WebSphere RFID Device Infrastructure agents for the Edge Controller are written using the MicroBroker Application Framework. Additionally, the MicroBroker Bridge component of MicroBroker is used in the WebSphere RFID Premises Server to connect to the Edge device and MQTT protocol.

Service Management Framework V3.7

The Service Management Framework Runtime is the IBM implementation of the OSGi Service Platform. The Framework acts as a layer that enables operators to deploy multiple applications on a single Java Virtual Machine (JVM™). In addition, it provides a framework for application life cycle including delivery to the device as well as dynamic starting and stopping of the applications.

Application developers partition applications into services and other resources. These services and resources are then packaged into OSGi bundles, files that serve as the delivery unit for applications. OSGi bundles have manifests with special headers that enable the sharing of classes and services at the package level. Bundles can be started and stopped dynamically, allowing systems to be updated without extended service or downtime.

IBM Services Management Framework (SMF) itself is a Java application, running in a J2ME JVM. In essence, it is the host environment in which the bundles execute. These bundles can represent several separate applications or portions of applications (all running in the same JVM).

Tivoli Configuration Manager V4.2.1

IBM Tivoli Configuration Manager provides the ability to capture your best practices for software distribution, automate those best practices, and enforce corporate standards. It helps you gain total control over your heterogeneous enterprise software and hardware.

The software distribution module enables you to deploy complex mission-critical applications rapidly and efficiently to multiple locations from a central point.

The inventory module lets you scan for and collect hardware and software configuration information automatically from computer systems across your enterprise. This inventory configuration data can be used in the reference models to remediate systems automatically that are not compliant (for example, making sure patches are installed to reduce vulnerabilities).

You can find more information about the Tivoli Configuration Manager at:

<http://www.ibm.com/software/tivoli/products/config-mgr>

Tivoli Enterprise Console V3.9

IBM Tivoli Enterprise Console® provides sophisticated, automated problem diagnosis and resolution to improve system performance and to reduce support costs. The new enhancements focus on time-to-value and ease-of-use with best practices available without modification to simplify and to accelerate deployment. The auto-discovery feature allows system administrators to understand the environment and process events appropriately. The Web console enhances visualization while providing remote access to events and console operations.

IBM Tivoli Enterprise Console highlights include the following:

- ▶ The real value in event management goes beyond simple filtering and provides root cause analysis and resolution.
- ▶ The new Web console provides improved visualization as well as access from anywhere.
- ▶ Pre-configured rules provide best-practices event management.
- ▶ Auto-discovery and problem diagnosis increase operator responsiveness and efficiency.

- ▶ Integrated network management extends Tivoli Enterprise Console reach and diagnosis for end-to-end management of your IT environment.
- ▶ Tivoli Enterprise Console enables comprehensive management that even accepts events from non-Tivoli products or systems.

IBM Tivoli Enterprise Console also includes IBM Tivoli Risk Manager (limited license), which provides monitoring and management of firewalls and intrusion detection systems. It also include IBM Tivoli Comprehensive Network Address Translator, enabling integrated management of overlapping IP domains.

You can find more information about the IBM Tivoli Enterprise Console at:

<http://www.ibm.com/software/tivoli/products/enterprise-console>

Tivoli Monitoring V5.1.2

IBM Tivoli Monitoring provides monitoring for essential system resources, to detect bottlenecks and potential problems and to recover automatically from critical situations. Tivoli Monitoring saves system administrators from scanning through extensive performance data manually before problems can be solved. Using industry best-practices, Tivoli Monitoring can provide immediate value to the enterprise.

Based on the new IBM Tivoli Monitoring technology, IBM Tivoli Monitoring Active Directory Option provides an invaluable set of pre-configured, automated best practices that are available without modification and that manage the directory proactively by monitoring essential resources and detecting potential problems. IBM Tivoli Monitoring Active Directory Option also provides seamless integration with other Tivoli solutions, including the Tivoli Business System Manager and the Tivoli Enterprise Console, providing a true end-to-end solution.

You can find more information about the IBM Tivoli Monitoring Web site at:

<http://www.ibm.com/software/tivoli/products/monitor>

IBM Tivoli Monitoring for Web Infrastructure V5.1.2

IBM Tivoli Monitoring for Web Infrastructure provides a single point of control to enable IT organizations to understand the health of the key elements of a Web-based environment. It allows administrators to identify problems quickly, to alert appropriate personnel as required, and to offer a means for automated problem correction that takes advantage of IBM best practices.

In addition, IBM Tivoli Monitoring for Web Infrastructure provides a real-time view of performance health and feeds a common data warehouse for historical reporting and analysis. Ultimately, this tool increases the effectiveness of an IT organization and ensures optimal performance and availability of the critical Web infrastructure.

IBM Tivoli Monitoring for Database Servers V5.1.0

IBM Tivoli Monitoring for Databases helps ensure the availability and optimal performance of DB2, Oracle, Informix®, Microsoft SQL Server, and Sybase database servers. Through the use of best practices incorporated within IBM Tivoli Monitoring for Database Servers, the typical database administrator dilemma of determining what to monitor, when to monitor, and how to interpret and act upon the monitoring results is eliminated. This leaves more time for the administrator to focus on more complex business-critical tasks. IBM Tivoli Monitoring for Database Servers provides routine, consistent monitoring that anticipates and corrects problems before database performance and customer confidence is degraded.

IBM Tivoli Monitoring for Database Servers provides a set of monitors for quick deployment and activation leveraging IBM best practices. Custom monitors, thresholds and tasks can also be defined by the database administrator. Through the implementation of IBM Tivoli Monitoring for Databases, database administrators are alerted when key performance and resource allocation problems are detected.

IBM Tivoli Monitoring for Business Integration V5.1.1

IBM Tivoli Monitoring for Business Integration provides complete management for IBM WebSphere MQ, IBM WebSphere MQ Integrator, IBM WebSphere MQ Workflow, and IBM WebSphere Interchange Server. From a central console, systems administrators can configure, monitor, and manage business integration software across host and distributed platforms. IBM Tivoli Monitoring for Business Integration offers stable, secure, and proactive monitoring and management through a powerful set of tools that takes advantage of common Tivoli technology to provide rapid time to value and greater ease of use.

IBM Tivoli Monitoring for Business Integration monitors the status of key WebSphere Business Integration components, such as queue managers, queues, channels, and message flows, identifies problems in real time, and notifies administrators by providing local correlation, root cause analysis and corrective action capabilities for quick problem resolution. IBM best practices are incorporated to provide expertise in managing these products. IBM Tivoli Monitoring for Business Integration provides the information needed to understand the business impact of the business integrated environment as the data collected allows you to analyze performance, trends, and address issues before they affect users.

3.2 WebSphere RFID Premises Server Architecture

The WebSphere RFID Premises Server is comprised of a J2EE runtime environment and a J2ME OSGi runtime environment (Figure 3-4). The J2EE environment contains the RFID Event Server and associated application and custom business logic. The RFID Event Server is supported by the IBM infrastructure components of WebSphere Application Server, DB2, and WebSphere MQ.

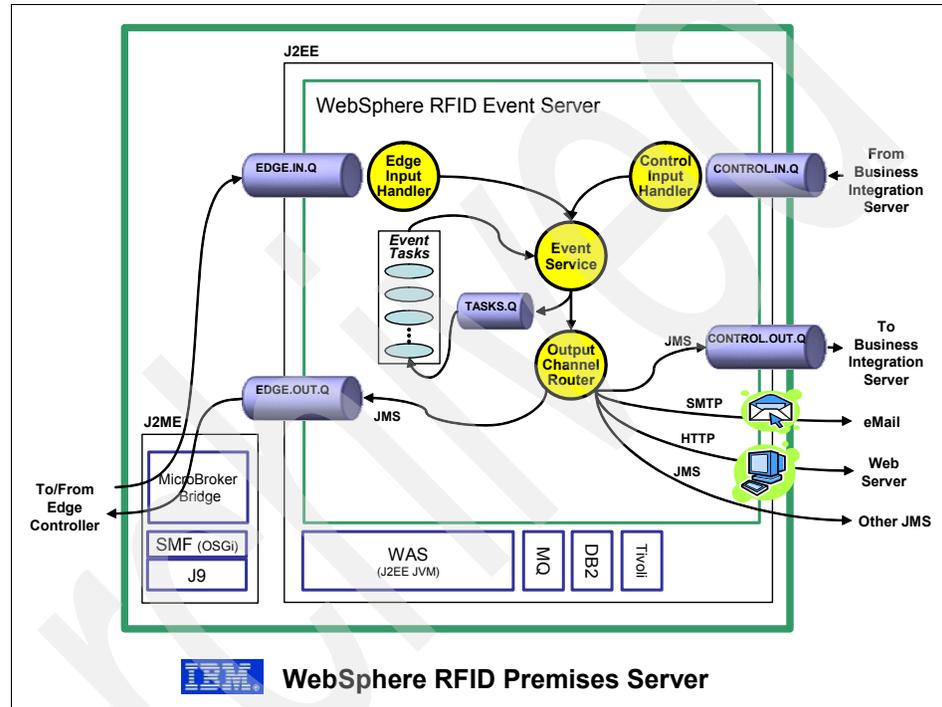


Figure 3-4 RFID Event Server component overview

The J2ME OSGi environment contains the MicroBroker Bridge component. The MicroBroker Bridge is simply responsible for transforming the event data from the Edge Controller to a format that can be understood and processed by the RFID Event Server. The MicroBroker Bridge is part of the IBM WebSphere Connection Server Micro Edition (MicroBroker), and runs under the SMF, which is the IBM implementation of OSGi, on an IBM Java Micro Edition JVM (J9) engine (the IBM implementation of J2ME).

The RFID event processing within the WebSphere RFID Premises Server as well as the interfacing of the WebSphere RFID Premises Server to external

components (for example, those of the Edge and Business Integration domains) are generally implemented using WebSphere MQ messaging.

3.2.1 MicroBroker Bridge

The MicroBroker Bridge is a component of MicroBroker software that connects the MicroBroker environment to other messaging environments, such as the WebSphere MQ environment of the WebSphere RFID Premises Server. The MicroBroker Bridge component of the WebSphere RFID Premises Server acts as an XML message translator between the MQTT publish-subscribe protocol, which is used by the WRDI-based Edge controllers, and the WebSphere MQ protocol, which is used by the RFID Event Server.

The MicroBroker Bridge uses Java Message Service (JMS) to communicate with the WebSphere MQ component in the J2EE environment, mapping and transforming the publish-subscribe topic space of the Edge's MQTT protocol to the WebSphere MQ protocol and queue structure that is used by the RFID Event Server.

In WebSphere RFID Premises Server, the MicroBroker Bridge is implemented as an OSGi bundle running in an IBM Services Management Framework (SMF) environment. The IBM SMF process within the WebSphere RFID Premises Server is generally installed and run as a Microsoft Windows® service.

3.2.2 RFID Event Server

The RFID Event Server is implemented as a J2EE application under WebSphere Application Server, which provides the basis for developing and deploying custom business logic as well as for the use of open standard interfaces (for example, Web Services, HTTP, RMI/IIOP, JMS, and JDBC) to connect and to integrate with other components.

3.3 Event processing

RFID event processing is the most fundamental function of the IBM WebSphere RFID Premises Server and the sole function of the RFID Event Server component. The processing of event information for a particular RFID solution must satisfy both the operational requirements of the particular RFID use case scenario (for example, pallet verification) and the business application and integration requirements.

3.3.1 Concepts

Understanding how events are processed in the WebSphere RFID Premises Server requires knowing some basics about the concepts of events, queues, tasks, and channels.

Events

An *event* is some significant occurrence in the RFID solution that is meaningful to the software components for RFID tracking or processing. Events are represented in the system by XML messages that contain information about the event. Event messages can represent abstracted events that are generated from RFID hardware devices (for example, tag reads from RFID tag readers), or they can represent internally generated system events that are meaningful only to the business logic.

The Premises Server includes a set of predefined event templates (for example, a Tag Read event). You can also extend the defined event types as required by your custom programming requirements. All event messages conform to the *IBMPremisesUnifiedMessageFormat* schema declaration.

Example 3-1 shows an XML message that represents the reading of an RFID tag.

Example 3-1 Sample XML message that represents the reading of an RFID tag

```
<?xml version='1.0' encoding='UTF-8'?><ibmprem:ibm-premises-unified-format
dts='2005-09-22T13:45:11' xmlns:ibmprem='http://www.ibm.com'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.ibm.com
IBMPremisesUnifiedMessageFormat.xsd'><event location='P1'
type='tag_read'><rfid-tag-data antenna='0' count='1' discovered='1127396259506'
reader='R1'
tagid='a5a5510114341813' /></event></ibmprem:ibm-premises-unified-format>
```

Queues

Event messages flow through the RFID infrastructure using a system of MQ message queues. The event originator places an event onto a particular message queue, not necessarily knowing what component or components will receive the message or how it is processed. More than one component can be waiting to process messages that arrive in a particular queue. Further, a component can be selective about what kind of message it processes from a particular queue.

Table 3-1 lists the five MQ queues that the RFID Event Server defines for its operation.

Table 3-1 RFID Event Server queues

Queue	Purpose
EDGE.Q.IN	For messages from Edge Controller(s)
EDGE.Q.OUT	For messages to the Edge Controller(s)
CONTROL.Q.IN	For messages from business logic
CONTROL.Q.OUT	For messages to business logic
TASKS.Q	For processing of messages

Tasks

When an event message is placed into a message queue, the WebSphere MQ then notifies one or more software components, called *Tasks*, which have expressed an interest in receiving (that is, are subscribed to) the messages placed there. Tasks then receive the event message and process it accordingly. Tasks are implemented in WebSphere Application Server as Message Driven Beans (MDBs). An MDB is a Enterprise Java Bean (EJB™) that allows J2EE applications to selectively process messages asynchronously. For example, in the Dock Door Receiving Starter Kit sample application, there is an event task called the TagReadHandler than gets invoked to process TagRead event messages. Also, note that the WebSphere RFID Premises Server allows you to define new tasks and to associate them to process specific event types.

Channels

A *channel* is the set of interfaces through which events messages are exchanged with the Premises Server. Essentially, channels are MQ queues that are used to communicate with external programs.

Input channels are predefined by the system and implemented as WebSphere MQ Java Message Service (JMS) queues. There are two input channels, one for each of the two adjacent domains of the Premises Server, namely the Edge Controller channel (EDGE.IN.Q) and the Control channel for Business Integration (CONTROL.IN.Q).

Output channels represent interfaces by which messages are sent from the Premises domain to the rest of the adjacent domains. Like the input channels, there are two output channels defined by the system. One output channel is for the Edge Controller (EDGE.OUT.Q) and the other is for the Control channel for Business Integration (CONTROL.OUT.Q).

Additional output channels can also be defined as required. Currently, the following output channels types are predefined for use:

- ▶ E-mail
- ▶ HTTP
- ▶ WebSphere MQ
- ▶ JMS message
- ▶ JMS topic

Further, these output channel types can be extended using custom programming.

3.3.2 Message flow

Figure 3-5 on page 69 shows the internal structure of event message queues, channels, and tasks that comprise the RFID Event server. The messages are queued up for processing in their respective queues, namely the EDGE.IN.Q and the CONTROL.IN.Q, which is part of the Event Server infrastructure.

In general, messages from the Edge Controller or business integration software are queued up for processing in their respective queues, EDGE.IN.Q and CONTROL.IN.Q. Messages are processed by respective input handler components, implemented as Message Driven Beans (MDBs). The input handlers then make use of the *Event Service interface* to publish messages to the internal *Task queue* or the respective output channels. Event message types can have output channels that are associated with them, in which case they are routed automatically to the pre-configured output channels.

Events are generally processed by handlers called event *Tasks*, which are implemented as Message Driven Beans (MDBs). Tasks listen for their respective messages on TASK.Q. The event is then consumed and processed, and the task in turn could make use of the *Event Service interface* to publish a new message to the output channels or back to the internal TASK.Q resource.

Note: Event Tasks can be created and configured through the administrative console application of the WebSphere RFID Premises Server. Tasks constitute the starting point for event processing, but need not, and should not, consume large amounts of processing to handle an event. Tasks should execute quickly and efficiently and make use of other components to process fully a given event. These other components could be deployed on the Premises Server or they could exist on the other computing systems in the enterprise.

3.3.3 Dock Door Starter Kit tag_read event example

The Dock Door Starter Kit (Kimono) application is described in 2.3.4, “WebSphere RFID Dock Door Receiving Starter Kit (Kimono)” on page 49.

Figure 3-5 on page 69 illustrates the path and sequence of processing steps that occur in the Event Server for a Tag Read event in the case of Kimono. Note how the tag_read event message gets mapped and transformed by the event tasks into new message types that are re-queued for possible processing by other interested event tasks. In this example, the tag_read event that are abstracted from the physical read on the Edge Controller propagates into a new_tag event, which is then propagated into a dock_door_receiving event.

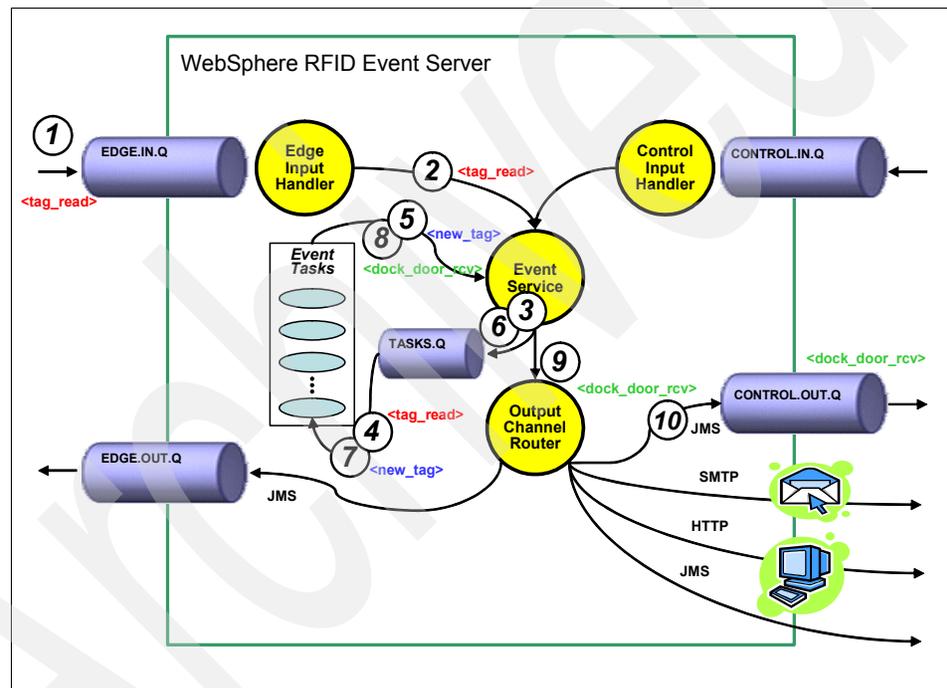


Figure 3-5 RFID Event Server message flow

The following steps describe the path and sequence of processing steps that occur in the Event Server for a Tag Read event:

1. The tag_read message flows from the Edge Controller and through the MicroBroker Bridge and into the EDGE.IN.Q message queue. The MicroBroker Bridge calls the *transform* specified in the following file:
C:\IBM\RFID\edgecontroller\smf\MicroBroker\PremisesBridge\bridge.properties

The transform must be registered, loaded into, and running in the Premises SMF. The called transform morphs the message from the published topic format of the MQTT protocol to the XML template format of the corresponding message type, which in this case is type tag_read.

Example 3-2 shows the MQTT message from the Edge Controller that indicates a tag a5a5510114341813.

Example 3-2 MQTT message from the Edge Controller

```
receiving/portal/P1/signal/tags = {a5a5510114341813=[{R1:1127397396761:0:1}]}
```

The MicroBroker Bridge calls the registered transform (which is loaded and running in the Premises SMF) to morph the message into what is shown in Example 3-3 to conform to the WebSphere RFID Premises Server XML schema and WebSphere MQ message format.

Example 3-3 MicroBroker Bridge transformed message

```
<?xml version='1.0' encoding='UTF-8'?><ibmprem:ibm-premises-unified-format
dts='2005-09-22T13:45:11' xmlns:ibmprem='http://www.ibm.com'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.ibm.com
IBMPremisesUnifiedMessageFormat.xsd'><event location='P1'
type='tag_read'><rfid-tag-data antenna='0' count='1' discovered='1127396259506'
reader='R1'
tagid='a5a5510114341813' /></event></ibmprem:ibm-premises-unified-format>
```

2. The Premises Server Edge Input Handler MDB is listening for messages placed on EDGE.IN.Q. The EJB container sends the tag_read message to the Edge Input Handler MDB. The Edge Input Handler MDB calls the Event Service to process the event message. In our example, for a tag_read event, there is one task defined, Tag Read Event Handler, and no Output Channels. The event service constructs the MDB message selector by prepending the Task ID to the event name. For the Tag Read Event Handler MDB, the message selector is:

```
IBMRFIDTASK='Tag Read Event Handler_tag_read'
```

3. The Event Service puts the tag_read message on the TASK.Q message queue with the Tag Read Event Handler MDB message selector to send the message to the Tasks interested in the message (or, in other examples, perhaps the Output Channels interested in the message, or both the Tasks and the Output Channels).

4. The EJB container sends the tag_read message to the Tag Read Event Handler MDB. The Tag Read Event Handler MDB fetches a list of previously read tags. If this is the first time this particular tag has been seen, the handler constructs a new_tag event. If the tag has been seen before, it constructs a repeat_tag event.
5. The Tag Read Event Handler MDB then calls the Event Service to send the message to the Tasks interested in the message.
6. For the new_tag and repeat_tag events, there are no Output Channels defined. There is one task, Dock Door Receiving Event Handler, defined. The event service constructs the MDB message selector by prepending the Task ID to the event name. For the Dock Door Receiving Event Handler MDB, the message selector is something like that shown in Example 3-4.

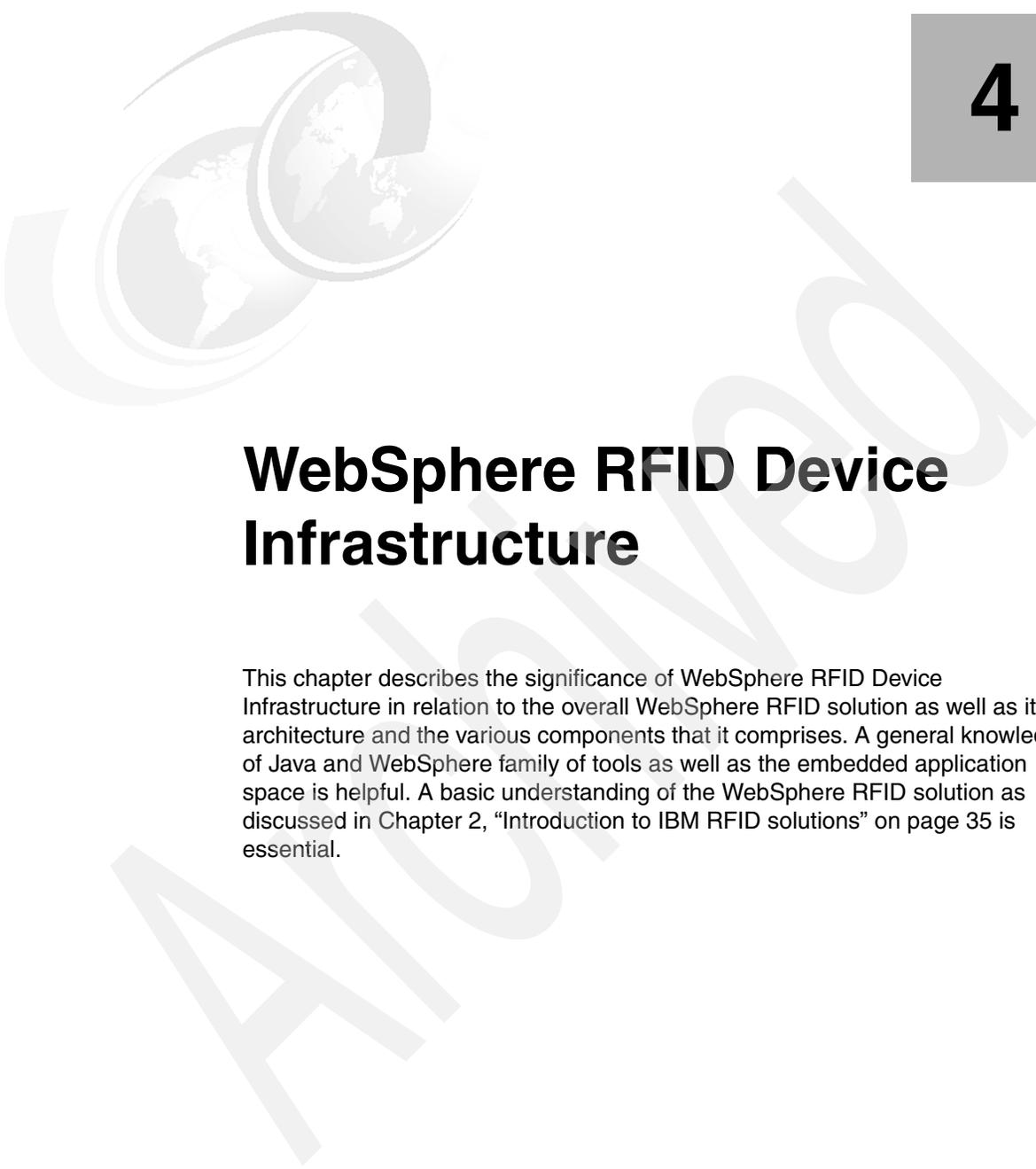
Example 3-4 Message selector for the Dock Door Receiving Event Handler MDB

```
IBMRFIDTASK='Dock Door Receiving Event Handler_new_tag' OR IBMRFIDTASK='Dock  
Door Receiving Event Handler_repeat_tag'
```

The Event Service puts the new_tag (or repeat_tag message) on the TASK.Q message queue with the Dock Door Receiving Event Handler MDB message selector.

7. The EJB container sends the new_tag or repeat_tag event message to the Dock Door Receiving Event Handler MDB.
8. The Dock Door Receiving Event Handler MDB then converts the new_tag or repeat_tag message into a dock_door_receiving event, and then calls the Event Service.
9. The Event Service determines that for the dock_door_receiving event type, there is one Output Channel defined, a JMS channel pointing to the CONTROL.OUT.Q message queue. There are no tasks defined. The Event Service calls the Output Channel Router.
10. The Output Channel Router puts the dock_door_receiving message on the CONTROL.OUT.Q message queue, bound for the custom business logic to process.

Archived



WebSphere RFID Device Infrastructure

This chapter describes the significance of WebSphere RFID Device Infrastructure in relation to the overall WebSphere RFID solution as well as its architecture and the various components that it comprises. A general knowledge of Java and WebSphere family of tools as well as the embedded application space is helpful. A basic understanding of the WebSphere RFID solution as discussed in Chapter 2, “Introduction to IBM RFID solutions” on page 35 is essential.

4.1 Overview of the WebSphere RFID Device Infrastructure

Figure 4-1 sets the context for this chapter, which focuses on the Edge Domain.

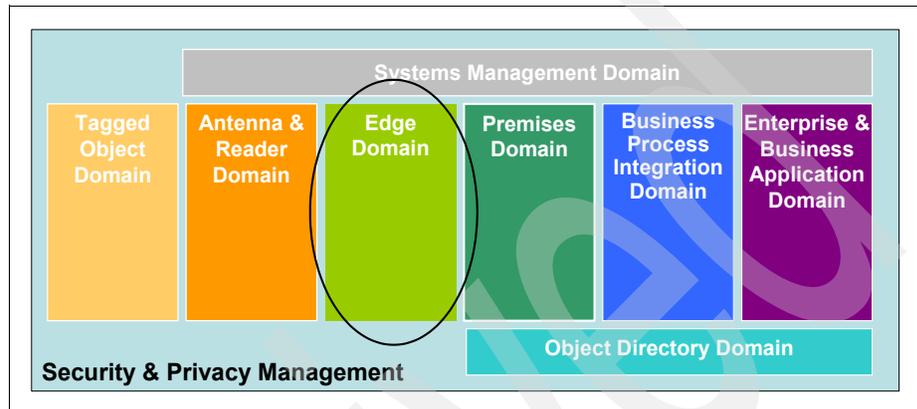


Figure 4-1 RFID Domain Model - Edge Domain

In the overall multi-tiered design of the IBM RFID solution, the Edge Domain bridges the gap between the Antenna and Reader Domain and the Premises Domain. Detailed discussions on the Antenna and Reader Domain are beyond the scope of this book. However, it is important to discuss it briefly in this chapter.

The Antenna and Reader Domain consists primarily of RFID readers and printers among others. While there are several readers and printers in the marketplace, the process of selecting, testing, deploying, and integrating them with the edge domain is quite complex. There is no real standardization in the technology thus far, and the differences in their operating systems, device administration, interfaces, communications protocols, and so forth, make the RFID implementation efforts a daunting task. Adding to the complexity, the devices frequently undergo firmware upgrades to keep up with the latest in standards and industry practices.

WebSphere RFID Device Infrastructure is a software package that is designed for the Edge Domain to help integrate these critical components seamlessly, irrespective of the choice of the device. Unlike the WebSphere RFID Premises Server, WebSphere RFID Device Infrastructure functionality is at the heart of pervasive computing. Disparate hardware and software is networked seamlessly to provide the *any device, any data, any place* capability. In such a dynamic environment, it is critical that a flexible infrastructure software be available to remove these complexities and to provide a manageable application to the user.

Note: WebSphere RFID Device Infrastructure is not a product, in the sense that it is not something you can install from a CD and configure right away. It is an infrastructure that provides all the necessary components for building an RFID solution. An IBM services partner can help integrate these components and configure them per your business requirements.

WebSphere RFID Device Infrastructure is a small footprint platform that can be extended to any memory constrained device (Figure 4-2). These devices are a small sample of such memory constrained devices that are capable of hosting WebSphere RFID Device Infrastructure.

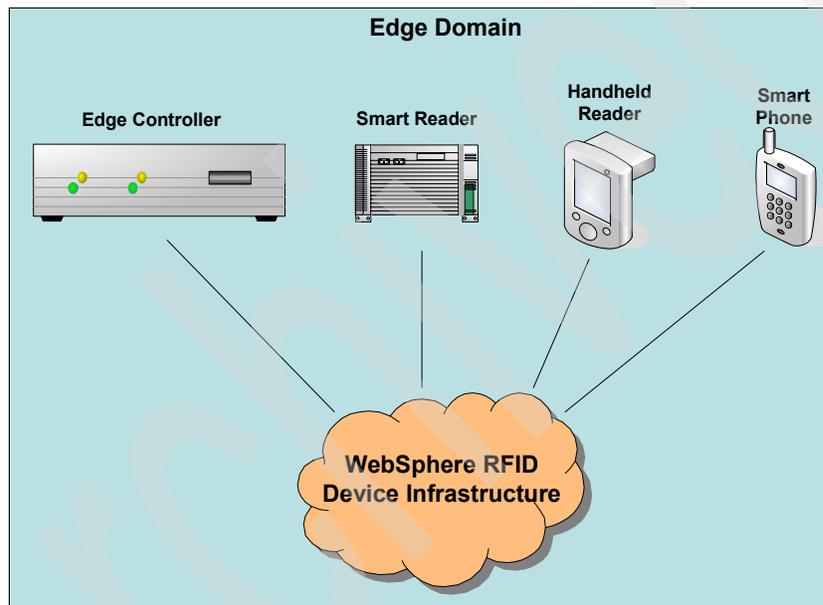


Figure 4-2 WebSphere RFID Device Infrastructure pervasive overview

Each of these devices are different in their constraints, as follows:

- ▶ Edge Controller

An embedded device with adequate processing power and memory. It is capable of hosting a variety of embedded applications and is capable of interacting and controlling RFID readers, printers, and any other infrastructure devices. Typically, this device is an industrial grade embedded system that is suitable for the environment in which it is deployed.

- ▶ **Smart Reader**
A new breed of intelligent RFID reader with additional capabilities to host embedded applications. The memory and processing constraints can be similar to a dedicated Edge Controller.
- ▶ **Handheld Reader**
A Handheld Reader usually is an RFID reader on handheld mobile device. The processing power and memory constraints can be similar to a typical handheld PDA.
- ▶ **Smart Phone**
An intelligent phone with an embedded operating system that is capable of hosting simple applications. Processing power and memory can be less than a PDA.

Attention: For the purposes of this discussion, the Edge Controller represents the hosting platform for WebSphere RFID Device Infrastructure. For a list of current OEM Edge Controllers that are supported by WebSphere RFID Device Infrastructure, refer to A.1, “IBM WebSphere RFID V1.0.2 matrix” on page 246.

WebSphere RFID Device Infrastructure comes pre-installed on your OEM Edge Controller through an IBM authorized OEM Business Partner only. You cannot purchase it directly from IBM. Contact an IBM account representative to obtain availability information.

The functionality of the WebSphere RFID Device Infrastructure in the Edge Domain is to provide a platform for device integration as well as early RFID data and business processing capability at the network’s edge. It facilitates the delivery of tag data to WebSphere RFID Premises Server. Besides the necessary hooks to build appropriate business process logic, it provides the base device adapters to integrate and control supported RFID hardware devices such as readers, scanners, printers, and so forth with the Edge Controller and ensure scalability and flexibility of the RFID solution deployment.

Important: The terms Edge Controller and WebSphere RFID Device Infrastructure are sometimes erroneously referred to as the Edge Server. It is important to note that WebSphere Edge Server is a separate product and has no significant bearing to the WebSphere RFID solution.

4.2 Key features

Figure 4-3 shows how the Edge Controller serves as the intermediary between the Premises Server and the devices on the edge (readers and printers).

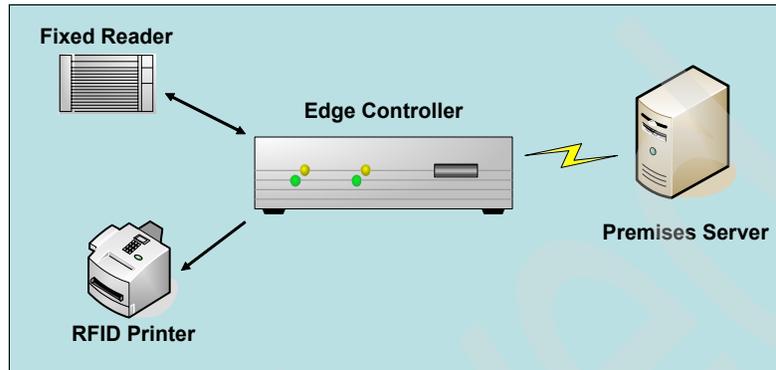


Figure 4-3 Edge Controller overview

Some of the key features of WebSphere RFID Device Infrastructure are:

► Device agnostic

WebSphere RFID Device Infrastructure acts as an integration point for RFID readers, printers, and other infrastructure devices, such as any sensors and actuators. RFID reader and printer manufacturers typically provide complex configuration options, proprietary interface languages, and communication protocols. WebSphere RFID Device Infrastructure eliminates the need for RFID integrators to be aware of these complexities and provides a device agnostic interface with which supported devices can be configured hiding the complexity. This allows for a single instance of WebSphere RFID Device Infrastructure to be capable of handling multiple readers and printers.

► Monitor sensors

Sensors are devices that detect changes in specific operating conditions and report vital data. In an RFID deployment, sensors are typically used to control the operation of RFID readers. Although RFID readers are capable of operating continuously, they are operated on a *need-to* basis for a variety of reasons:

- Some countries have *duty-cycle* restrictions on the power output from these readers as well as the amount of time that they can be ON in any given time period.
- Having the readers ON when not needed can cause unnecessary tag reads from stray tags in the read range, which can generate network traffic as well as kick off unintended business processes.

- Avoiding interference with other readers in the area.
- Even though no noticeable effects of RFID transmissions on humans has been documented, many prefer to avoid the possibility.

By using a sensor that is capable of providing a digital input, WebSphere RFID Device Infrastructure can control readers as necessary. Some of the example cases where a sensor might be helpful are:

- Fork lift truck approaching or leaving a dock door.
- Conveyor belt moving at a certain speed.
- Dock door opening or closing.
- Shrink wrap station spinning at a certain speed.

Sensors that are calibrated to pick up these individual events can transmit a digital signal to the RFID readers that can trigger the appropriate event.

► Operate control devices

Control any actuators or annunciators that need to be operated based on specific RFID events.

– Actuators

Actuators are mechanical devices that are capable of controlling or moving objects. Examples include:

- Robotic arm
- Automatic gate opener

– Annunciator

Annunciators are electrically controlled devices. Examples include:

- A visual light stack
- A LED display

► Provide filtering

Tag data is critical in RFID implementations. However, when tag data is not managed, tag reads can be propagated upstream to WebSphere RFID Premises Server causing network overload. WebSphere RFID Device Infrastructure provides configurable filters to ensure data that is required by the upstream processes is propagated and the redundant data filtered out. This is a very helpful feature in keeping network traffic down and eliminating the need to boost network infrastructure costs.

When RFID readers are in Read mode, they interrogate all tags in the read zone. The readers report on tags as they find them, this causes several duplicate reads to be passed to WebSphere RFID Device Infrastructure.

WebSphere RFID Device Infrastructure allows for the configuration of tag reads such that only unique tag reads are passed upstream to Premises Server. Some of the examples by which filtering can be performed are:

- Bad tag filtering

Any tag data that fails the data integrity checks will be filtered as a bad read, preventing such data from being sent to the Premises Server and additional processing in that domain.

- Duplicate tag filtering

Duplicate tag reads that are received by the Edge Controller within a short time interval will be considered to be part of the same read and thus, filtered out. The time interval itself is a configurable property.

- Spatial filtering

When multiple readers operate within close proximity from each other, a tag can potentially be detected by a wrong reader along with the correct one. In such a situation the spatial filter can filter the data and assign the correct reader for that tag read based on the signal strength received from the tag.

- ▶ Aggregate tag data

Tag reads can be propagated to the WebSphere RFID Premises Server individually, however, it is highly desirable to aggregate multiple reads into single transaction before sending the data upstream. Each transaction might have a processing overhead that can be reduced, by aggregating appropriately. WebSphere RFID Device Infrastructure allows for tag data to be aggregated on the Edge Controller, reducing the number of individual transactions that need to be sent between the Edge Controller and the Premises Server.

- ▶ Provide software distribution management

Reduces Total Cost of Ownership by allowing remote management of devices. Existing readers frequently undergo firmware updates to ensure compliance with global standards. Remote management speeds deployment.

- ▶ Perform reliably

It is critical that tags that are read by a reader and intended for WebSphere RFID Premises Server actually make their way there. WebSphere RFID Device Infrastructure provides guaranteed message delivery, in which a tag that is read by an RFID reader is actually delivered to the Premises Server, provided it is not filtered by any applicable filtering or business logic at the Edge Controller. By the same token, any other special RFID events also make their way to the Premises Server.

WebSphere RFID Device Infrastructure operates in an asynchronous mode with the Premises Server, making it a stand-alone unit capable of operating without back-end system support. This ensures that it is available to operate readers and devices without interruptions.

- ▶ Develop applications easily

WebSphere RFID Device Infrastructure is based on technologies from IBM that can run efficiently on many memory constrained devices. It is important for application developers to be able to create and to deploy applications easily. To this end, processes and applications can be built on an Eclipse platform and executed at the edge.

4.3 Core technologies

WebSphere RFID Device Infrastructure is an open standards based middleware platform that is designed for memory constrained devices, such as the Edge Controller. It is built based on three core technologies from IBM (Figure 4-4):

- ▶ Workplace™ Client Technology, Micro Edition (WCTME)
- ▶ WebSphere Sensors and Actuators Base Infrastructure
- ▶ WebSphere Identification Tracking Kit

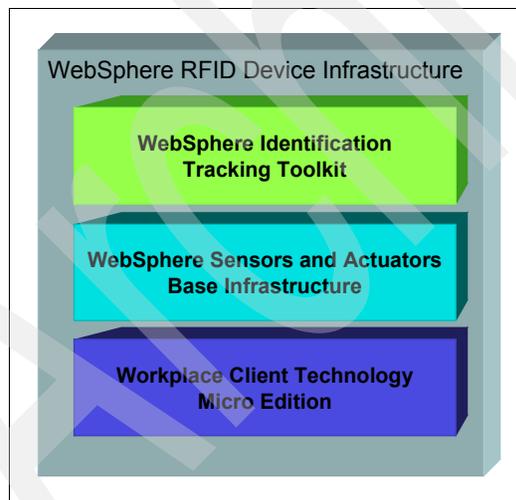


Figure 4-4 WebSphere RFID Device Infrastructure overview

The sections that follow discuss these core technologies in detail.

Note: The core technologies that we discuss here might be packaged with additional components that we do not discuss. Our discussion focuses on the key components from these technologies that play an important role in the WebSphere RFID Device Infrastructure architecture.

4.3.1 Workplace Client Technology, Micro Edition

IBM Workplace Client Technology™, Micro Edition is an e-business application platform that assists in the deployment of applications to server-managed clients. It delivers a Java powered platform where devices have access to pre-tested content, can be maintained over the air (OTA), and are pre-enabled for access to many enterprise applications, data, and transactions.

It allows you to build server managed clients that support multiple user types, user experiences, access points, and forms of connectivity enabling flexible and cost effective access to business processes, applications, and content. Based on the Eclipse platform from IBM, it is a commitment from IBM to open standards combined with the portability of Java for pervasive devices.

The key aspects of Workplace Client Technology, Micro Edition are that it is:

- ▶ Componentized
- ▶ Offline capable
- ▶ Secure
- ▶ Server managed

Important: Workplace Client Technology, Micro Edition is:

- ▶ Configurable to address memory and CPU constraints
- ▶ Available on multiple platforms and in multiple configurations, supporting application deployment across millions of devices

Refer to *IBM Workplace Client Technology Micro Edition Version 5.7.1: Application Development and Case Study*, SG24-6496 for further information.

Workplace Client Technology, Micro Edition is an enabling technology formed by the components shown in Figure 4-5.

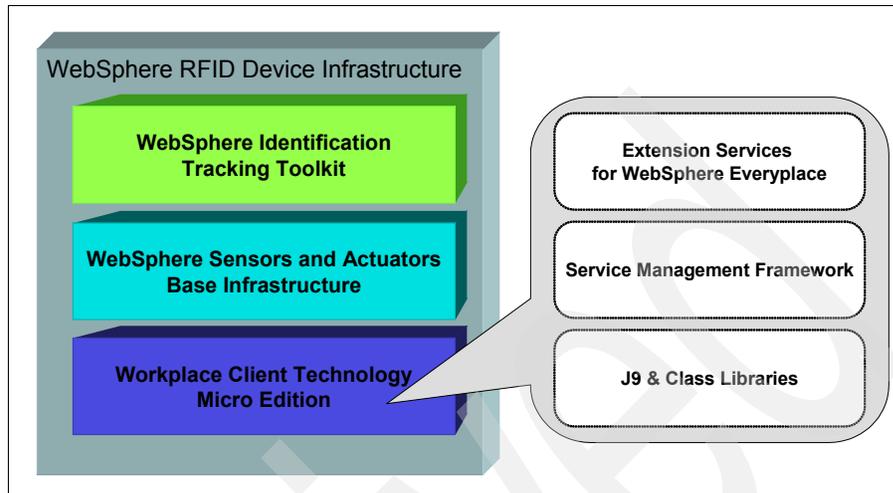


Figure 4-5 Workplace Client Technology Micro Edition overview

J9 and Class Libraries

The IBM J9 Virtual Machine and associated class libraries provide a runtime environment where Java code for the memory constrained devices can be executed. J9 is the IBM implementation of Java Virtual Machine (JVM) for Java 2 Micro Edition (J2ME). It implements a configurable, compact, fast, and predictable architectural layer that provides a common interface for application programs regardless of the underlying device, hardware or operating system. Because it is created with portability in mind, porting applications between platforms and devices is rather easy and seamless.

Service Management Framework

IBM Service Management Framework is a production-ready software management platform for network-delivered applications. It is ideally suitable for resource constrained devices such as set-top boxes, but is equally capable on full-scale servers.

Note: The IBM Service Management Framework is an implementation of Open Services Gateway Initiative (OSGi) V3.0 specification. The OSGi Alliance is an industry group that defines and promotes an open standard for the network delivery of managed services to local networks and devices. For more information about OSGi, visit their Web site at:

<http://www.osgi.org>

Some of the key features of Service Management Framework include:

- ▶ A managed service platform that can be remotely configured, without the need to shutdown the underlying device or JVM.
- ▶ 100% portable with no native code.
- ▶ Deployed components are independent of implementation of other components.
- ▶ Highly reliable and scalable infrastructure to handle large-scale deployments.

Software components that are created to run in Service Management Framework runtime environment are called *bundles*. Figure 4-6 illustrates how you can deploy bundles in a Service Management Framework runtime environment. You can deploy them as independent units or take advantage of one or more bundles.

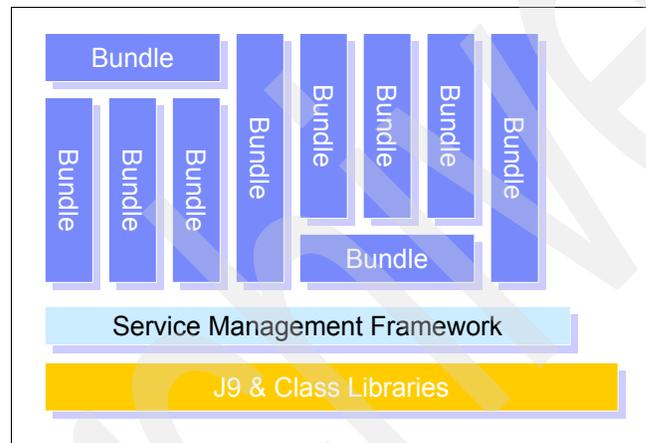


Figure 4-6 Service Management Framework architecture

Bundles are delivered over the network to the device and can publish services that are subscribed to by other bundles. In this way, they are loosely coupled components that collaborate through services in a well-defined manner. They can be installed and uninstalled at any time, without effecting other bundles.

Extension Services for WebSphere Everyplace

Extension Services for WebSphere Everyplace® is an integrated middleware and tooling platform that enables connection independent delivery and management of applications and services to pervasive devices. It is a component based runtime environment based on Service Management Framework that facilitates applications to be run on network enabled devices in a connected, disconnected or intermittently connected mode. Extension Services applications are designed in much the same way as standard enterprise applications. However, the environment in which these applications executes

has different capabilities than one might expect of applications running on a single node in an enterprise system.

Extension Services enables the development of applications for a wide range of platforms and devices extending the capabilities provided by Service Management Framework to provide additional enterprise capabilities. Extension Services provides enterprise components that can be used in solutions for the following situations:

- ▶ Platform

Extension Services based applications can run on a minimum set of J9 class libraries, providing the flexibility to deploy applications on a wide range of resource constrained devices and platforms.

- ▶ Platform Management

Provides a SyncML/DM based OSGi Agent that is capable of interacting with a Device Management Server to check on jobs to execute and to manage software distribution and device configuration.

Note: SyncML/DM is an Open Management Alliance (OMA) driven initiative to provide standards based remote management of devices. OMA is the leading forum for developing market driven, interoperable mobile service enablers. For further information regarding OMA, see:

<http://www.openmobilealliance.org>

- ▶ Data

Provides access to relational databases using Java Database Connectivity (JDBC) interfaces.

Note: JDBC is a set of APIs from Sun™ Microsystems™ that simplifies external access to databases by providing necessary library routines.

- ▶ Messaging

Supports Java Message Service (JMS) based messaging along with IBM MQ Everyplace as a messaging provider. Also supports the IBM lightweight publish/subscribe TCP/IP protocol.

Note: JMS is Sun's standard API for message queuing systems.

► Interaction Services

Ability to create Web applications with complete Graphical User Interface capabilities.

Note: This is not an exhaustive list, nor is it comprehensive of all possible decision points when deploying applications using Extension Services for WebSphere Everyplace. It provides the topics that you might need to consider when deploying client applications.

4.3.2 WebSphere Sensors and Actuators Base Infrastructure

IBM Sensors and Actuators is a business unit within IBM that is responsible for building RFID as well as other sensor-based technology solutions. WebSphere Sensors and Actuators Base Infrastructure is an application platform, which provides a base level of software that is suitable for a wide range of industry domains, such as Identification Tracking, Industrial Automation, Asset Monitoring and In-Vehicle Tracking. Built on top of the IBM Workplace Client Technology Micro Edition, it comprises of the lowest common denominator of all sensors and actuators core components and is packaged together to provide an optimal base runtime environment.

Note: WebSphere Sensors and Actuators Base Infrastructure brings together common infrastructure components that can be extended by both RFID as well other non-RFID industry applications. For further information regarding the Sensors and Actuators initiative, see:

<http://www.ibm.com/solutions/businesssolutions/sensors/index.jsp>

WebSphere Sensors and Actuators Base Infrastructure is made up of other key technology components, as shown in Figure 4-7.

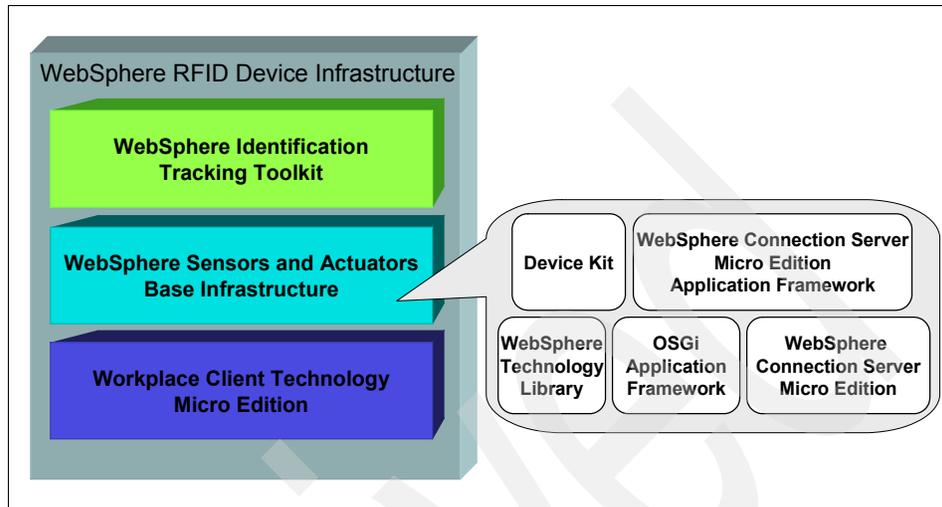


Figure 4-7 WebSphere Sensors and Actuators Base Infrastructure

WebSphere Technology Library

This is a set of API that can be used with the J9 and class libraries to supplement additional functionality on embedded devices such as serial communications. This way, you can build applications that depend on the base J9 and class libraries as well as the selected API from the WebSphere Technology Library.

Device Kit

Device Kit is a tooling kit that allows accelerated and simplified development of device adaptors that interact with the physical device such as RFID readers, printers and any infrastructure devices such as Programmable Logic Controllers (PLC) or other sensors and actuators. Adaptors that are created by the Device Kit are 100% Java and are portable across platforms and hardware. Adaptors can coordinate signals and messages between hardware and the application.

OSGi Application Framework

OSGi Application Framework is an IBM custom framework that is built on top of the IBM implementation of OSGi Service Gateway (Service Management Framework). It provides a well-defined application level view of Service Management Framework and takes out the complexity that is involved in the inner workings of OSGi by allowing the developer to focus on the application functionality.

The main benefits of using OSGi Application Framework are:

- ▶ OSGi Application Framework is 100% pure Java and installs itself as a bundle (Figure 4-8).
- ▶ Reduced bundle footprint by using common abstractions.
- ▶ Increased bundle reliability and predictability
- ▶ Reduced bundle development time.
- ▶ Reduced training costs, eliminating the need to train developers at the Service Management Framework level.

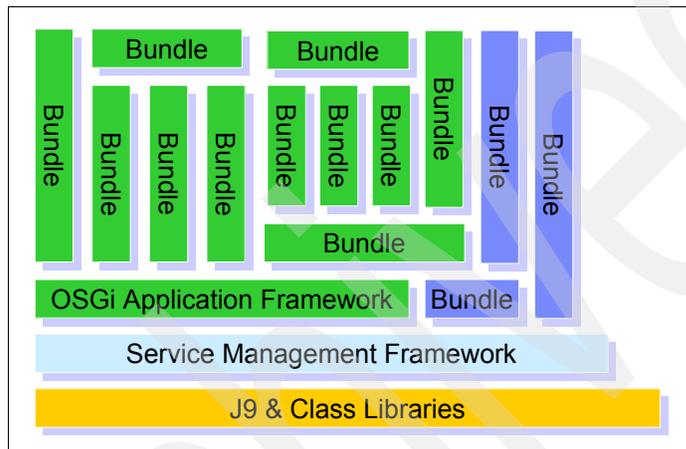


Figure 4-8 OSGi Application Framework Architecture

WebSphere Connection Server Micro Edition

WebSphere Connection Server Micro Edition is one of the most important technologies of WebSphere RFID Device Infrastructure. It is a very small footprint, 100% Java technology capable of running on resource constrained environments such as cars, Programmable Logic Controllers (PLC), set top boxes, edge of network, and leaf node devices such as the Edge Controllers. It is also suitable for embedding in applications and solutions that have a need for messaging, notification and event services, where messaging is enabled by WebSphere MQ Telemetry Transport, a lightweight publish-subscribe protocol over TCP/IP.

Note: Under the publish-subscribe mechanism, WebSphere Connection Server Micro Edition bundles communicate with each other using a *topic*. A topic is a key that describes the published data. A topic published by a publishing bundle is sent automatically to another bundle that subscribes for that topic.

WebSphere Connection Server Micro Edition runs stand-alone as OSGi bundles (Figure 4-9). The two most important components of WebSphere Connection Server Micro Edition that provide this messaging infrastructure are:

► MicroBroker Bus

This is the message broker part of WebSphere Connection Server Micro Edition, a very small footprint implementation for the WebSphere MQ Telemetry Transport communications.

Conceptually, WebSphere Connection Server Micro Edition is smaller than the IBM enterprise message broker by an order of significant magnitude. It provides a messaging infrastructure that enables tiny messaging clients to communicate on one box or across boxes.

In the case of WebSphere RFID Device Infrastructure, the MicroBroker Bus provides the messaging platform by which one WebSphere Connection Server Micro Edition bundle can communicate with another.

Note: MicroBroker Bus is also commonly referred to as *MicroBroker*.

► MicroBroker Bridge

A MicroBroker bridge connects one MicroBroker and other WebSphere MQ Telemetry Transport capable message brokers (including other MicroBrokers) as well as WebSphere MQ.

In the case of the WebSphere RFID solution, the MicroBroker bridge allows an Edge Controller to connect to the Premises Server and exchange critical messages.

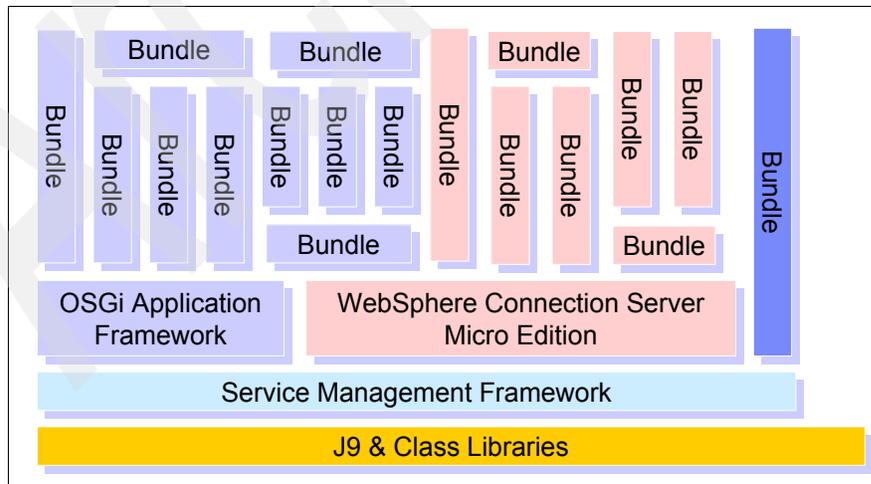


Figure 4-9 WebSphere Connection Server Micro Edition Architecture

WebSphere Connection Server Micro Edition Application Framework

WebSphere Connection Server Micro Edition Application Framework is an IBM custom framework that is built on top of WebSphere Connection Server Micro Edition as well as the OSGi Application Framework (Figure 4-10 on page 90). It simplifies the process of building applications that use WebSphere Connection Server Micro Edition, as it provides a well-defined application level view of OSGi Application Framework and WebSphere Connection Server Micro Edition combined. WebSphere Connection Server Micro Edition Application Framework is a 100% pure Java bundle and by taking care of some of the underlying issues it allows application developers to focus on domain logic instead of the inner workings of WebSphere Connection Server Micro Edition.

The benefits of using WebSphere Connection Server Micro Edition Application Framework include:

- ▶ Standardizing the use of WebSphere Connection Server Micro Edition across the application.
- ▶ Assisting in connecting to the MicroBroker Bus and managing the connection.
- ▶ Simplifying the creation and maintenance of applications.
- ▶ Reducing the size and complexity of an application by refactoring common behavior into a framework.
- ▶ Increasing application quality by applying best practices.

Note: WebSphere Connection Server Micro Edition Application Framework based bundles are called *MicroBroker agents*, which communicate over WebSphere MQTT protocol.

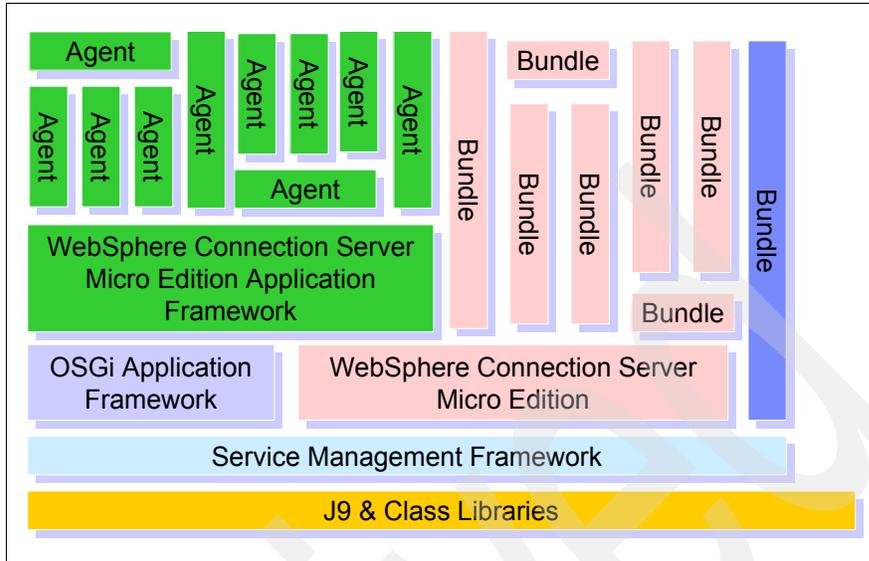


Figure 4-10 WebSphere Connection Server Micro Edition Application Framework

4.3.3 WebSphere Identification Tracking Kit

WebSphere Identification Tracking Kit is the RFID layer of WebSphere RFID Device Infrastructure. It consists of the RFID Device Kit, as shown in Figure 4-11.

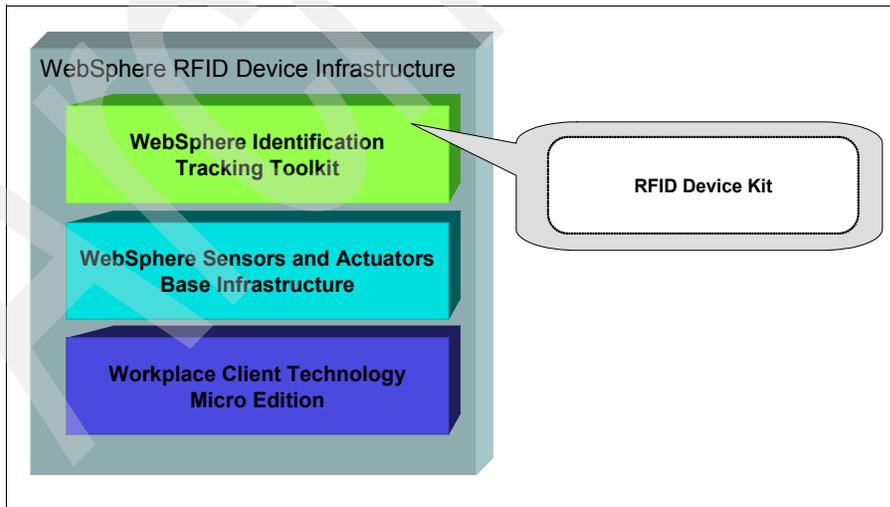


Figure 4-11 WebSphere Identification Tracking Kit

RFID Device Kit

The RFID Device Kit is similar to the master Device Kit (discussed as part of the WebSphere Sensors and Actuators Base Infrastructure) with the exception that it wraps the adaptors that are created by the base device kit into MicroBroker Agents.

4.4 Internal architecture and operation

WebSphere RFID Device Infrastructure is built on an agent based architecture, where all tasks and operations are packaged as loosely coupled MicroBroker agents that communicate with each other using the MicroBroker.

4.4.1 Agent architecture

An *agent* is a software component running on the Edge Controller, which uses the publish/subscribe mechanism of MicroBroker bus to communicate with the other software components of the WebSphere RFID Device Infrastructure (Figure 4-12).

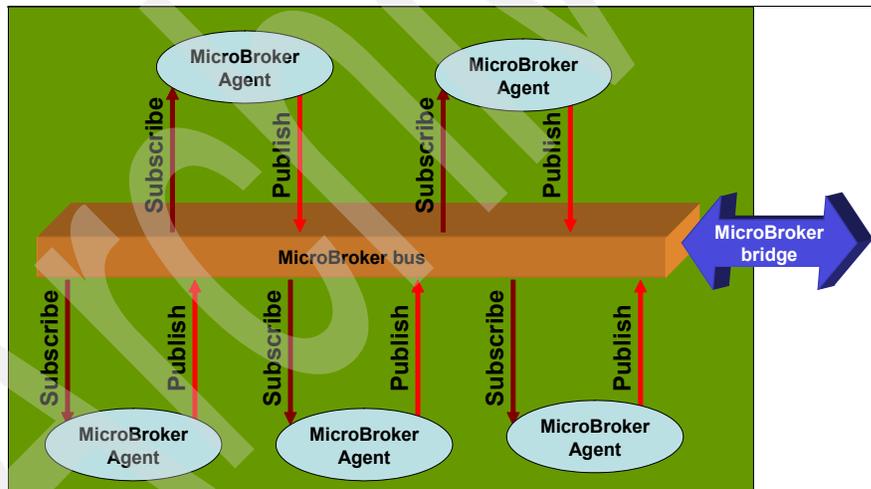


Figure 4-12 Agent Architecture overview

Likewise, the MicroBroker agents on the Edge Controller communicate with the Premises Server through the MicroBroker bridge over the WebSphere MQTT protocol using a publish/subscribe mechanism.

Note: A MicroBroker agent can be thought of as a black box that receives input, performs some processing, and transmits an output. Its role is to receive input — should it be hardware or a MicroBroker agent subscribed message — process the input, and produce output — should it be hardware or a MicroBroker publish message.

Each agent is specialized for a particular task, to ensure loose coupling between agents, thus loading only those agents which are necessary to this Edge Controller assigned role.

Because the MicroBroker agents are loosely coupled, they are optional when no other agent relies on their output. For example, all the devices and RFID reader related agents might not be defined when the corresponding hardware that they are driving is not present in the solution platform.

4.4.2 Operations overview

In this section, we discuss all of the major integration points and how they are connected to each other.

Edge configuration

Figure 4-13 shows an overview of the Edge configuration.

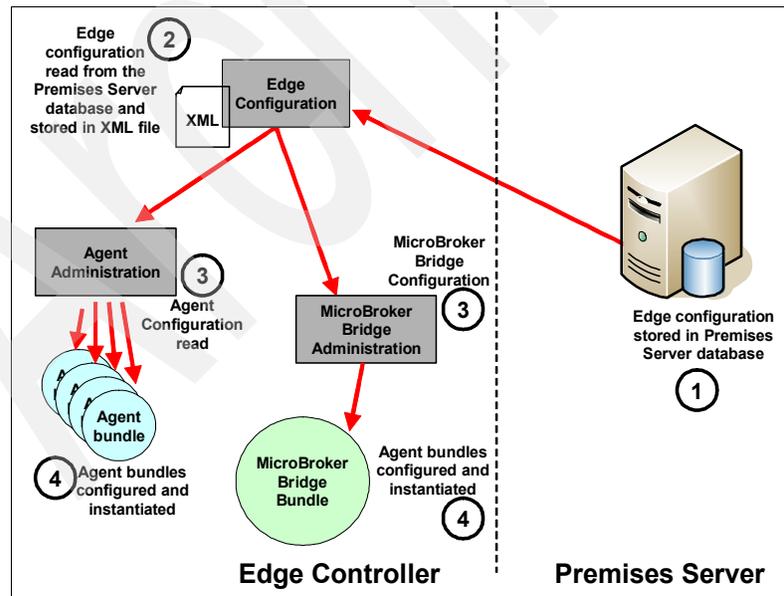


Figure 4-13 Edge configuration overview

This is how the Edge configuration is fetched:

- ▶ The Edge Controller configuration is fetched from the Premises Server database when the Edge starts.
- ▶ It is then stored in an internal XML file which is accessed by:
 - An Agent Manager component to generate all the necessary agents needed to run the Edge Controller.
 - A MicroBroker Administration component to initiate the MicroBroker Bridge and set all the properties.
- ▶ When all the necessary agents are generated and instantiated, the Edge Controller is ready to operate.

Note: Defining Edge Controller configuration on the Premises Server is discussed in 7.3.4, “Controllers” on page 162.

MicroBroker Bus and MicroBroker Bridge

The MicroBroker Bus and MicroBroker bridge are the communications backbone of WebSphere RFID Device Infrastructure that are installed on a Edge Controller. As shown in Figure 4-14, messages can flow between the MicroBroker agents as well as between the Edge Controller and the Premises Server using these two core components as follows:

- ▶ Messages originating from the MicroBroker agents are published to the MicroBroker bus, which are then propagated to the MicroBroker Bridge and on to the Premises Server.
- ▶ Messages originating from the Premises Server are published to the MicroBroker bridge and are propagated to the subscribing agents using the MicroBroker bus.

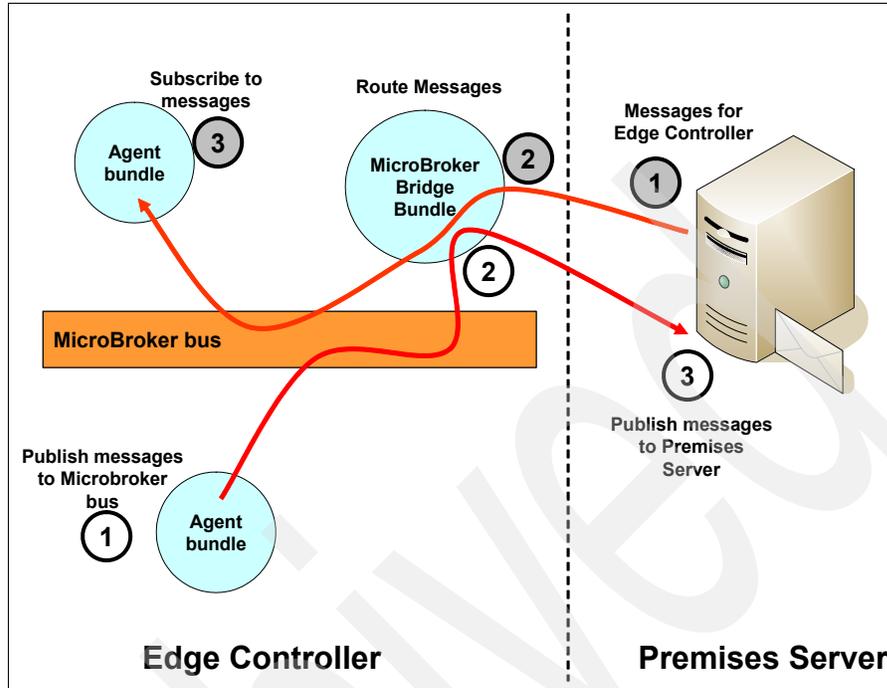


Figure 4-14 MicroBroker Bus and MicroBroker Bridge Configuration operation

Device Kit extension

The Device Kits that are provided as part of the WebSphere Sensor and Actuators Base infrastructure and the WebSphere Identification Tracking Kit allow the construction of the necessary software components to interface with the physical hardware devices as well as wrap them up as MicroBroker agents, so they can be easily integrated into the WebSphere RFID Device Infrastructure. Devices typically respond to the configuration and control commands that are sent by the MicroBroker agents, while the agents receive the data and status messages that are sent by the devices and propagate them to subscribers of this information using the MicroBroker bus (Figure 4-15).

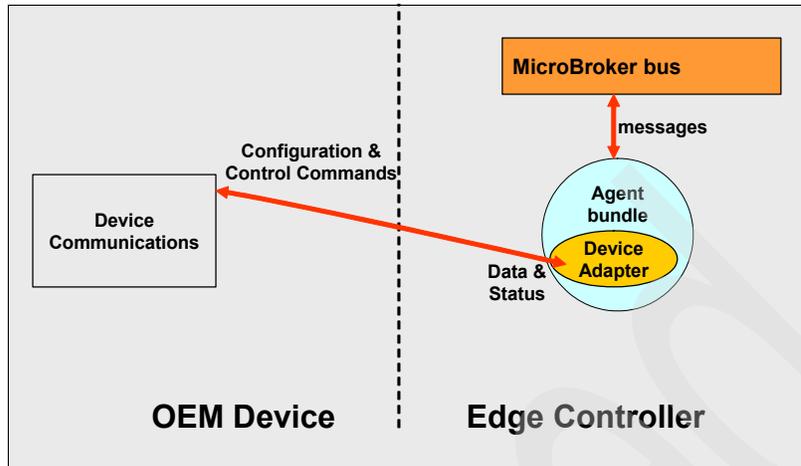


Figure 4-15 Device communications

Additional operations

In addition to the integration options that are readily available through the agent architecture of WebSphere RFID Device Infrastructure, developers can implement additional integration methods by extending the core technologies of WebSphere RFID Device Infrastructure, such as the Extension Services for WebSphere Everyplace (Figure 4-16). Some such integration options might include (but are not limited to) the following:

- ▶ Web Services
- ▶ Java Message Service (JMS) interfaces
- ▶ Graphical User Interfaces
- ▶ Database persistence

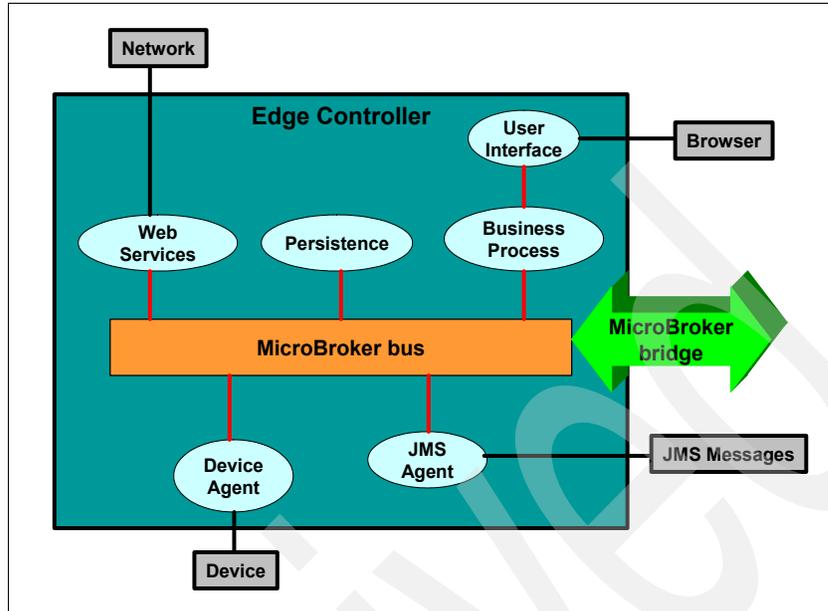


Figure 4-16 Additional extension options

In Figure 4-16, note how the agent architecture and the core technologies on which WebSphere RFID Device Infrastructure are built can be integrated seamlessly with external systems.

Important: Figure 4-16 is only a sample to illustrate the additional choices that are available to integrate WebSphere RFID Device Infrastructure with an external system of your choice. We present it here for clarity of the discussion topic within the scope of this section. It has no significance with the remaining chapters or the current implementation of the WebSphere RFID solution.

An IBM service partner can provide additional information that is applicable to a specific requirement that you might have. However, you might need to negotiate a separate service agreement with IBM for any deviations from the recommended implementation that is covered by the WebSphere RFID product documentation.

4.5 Dock Door Receiving Starter Kit

WebSphere RFID Device Infrastructure provides a set of sample code in support of Dock Door Receiving reference implementation. The code is available in the form of MicroBroker agents.

4.5.1 Example agents

The following is a list of agents that are readily available with the Dock Door Receiving Starter Kit:

- ▶ **Alert Agent**
Logs informational, debug, and error data up to the Premises Server.
- ▶ **Controller Agent**
Receives information about the status of I/O devices and issues commands to other I/O devices based on that status. Essentially, this is a central piece of the agent architecture whose sole responsibility is to coordinate events within the WebSphere Connection Server Micro Edition.
- ▶ **Duty Cycle Agent**
- ▶ **Filter Agent**
Filters RFID tag data to filter duplicates or case tags.
- ▶ **Heartbeat Agent**
Monitors the state of each RFID reader and forwards it to the Premises Server.
- ▶ **LightTree Agent**
Responsible for controlling visual indicators on a light tree that accepts digital input.
- ▶ **MotionSensor Agent**
Reports data received from a motion sensor capable of providing digital output.
- ▶ **Restart Agent**
- ▶ **Switch Agent**
Controls the state of a switch.

Note: In addition to these agents, the Dock Door Receiving Starter Kit also ships with one agent for each supported RFID reader and Edge Controller. For a list of currently supported hardware, refer to Appendix A, “Supported device matrix” on page 245.

4.5.2 Reference implementation

The Dock Door Receiving reference implementation is a sample implementation to illustrate a use case of an end-to-end implementation of the WebSphere RFID solution (Figure 4-17). Tags that are read by a reader at the reference dock door are propagated to the Premises Server for validation and the response is returned back to the Edge Controller. The scenario also demonstrates the use of a motion sensor, switch, and light tree to illustrate the flow of messages among various components within the agent architecture as well as within the overall WebSphere RFID solution.

Note: The Dock Door Receiving reference implementation on the Edge Controller is also commonly known by the name of *Kimono*. Thus, all of the readily available agents are also called Kimono agents.

Each of the Kimono agents shown in Figure 4-17 publishes and subscribes to a topic or a set of topics (also shown).

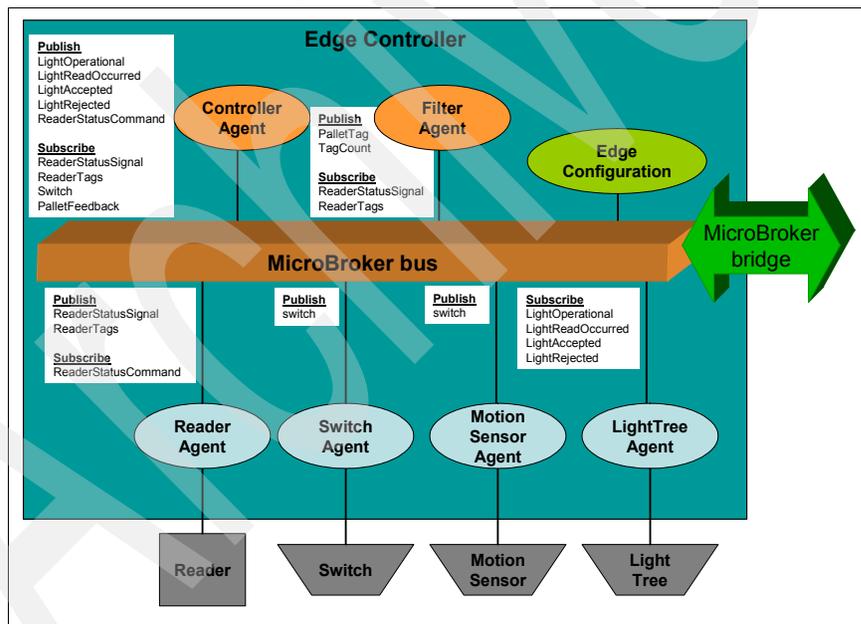


Figure 4-17 Dock Door Receiving reference implementation

Chapter 8, “Running the Dock Door Receiving scenario” on page 181 provides step-by-step instructions on how to run the Dock Door Receiving reference implementation.



Part 2

Setting up the WebSphere RFID solution

In this part, we give step-by-step instructions for installing the WebSphere RFID Premises Server and related software. We explain the Administrative Console and the tasks that are required to get your system started. You will learn how to use the Administrative Console to configure the Premises Server, Edge Controllers, RFID readers, and the components that enable them to communicate.

Archived



Planning your WebSphere RFID solution

This chapter tells you the information that you need to know and collect prior to installing an RFID solution, such as the supported platforms, the solution topology to be adopted, and the software pre-requisites.

5.1 Platform support

IBM WebSphere RFID solution includes software both for the Edge Controller and the Premises Server. This section details their hardware and software requirement as well as the RFID devices that they support.

Refer to Chapter 3, “IBM WebSphere RFID Premises Server” on page 53 and Chapter 4, “WebSphere RFID Device Infrastructure” on page 73 for more information about the concepts of Premises Server and Edge Controller.

5.1.1 Premises Server hardware requirements

The following minimum hardware configuration is recommended:

- ▶ RAM: 2 GB
- ▶ Processor: 3 GHz Pentium® 4
- ▶ Free disk space: 8 GB
- ▶ Temporary disk space during installation: 500 MB

5.1.2 Premises Server software requirements

The Premises Server must be installed either on Windows 2000 Server Service Pack 4 or on Windows Server 2003 Service Pack 1.

Prior to installing the Premises Server software, the following software must be installed:

- ▶ Premises Server machine:
 - WebSphere Application Server V5.1 Fix Pack 1
 - DB2 Workgroup Server V8.1.4
 - WebSphere MQSeries® V5.3.0.8
- ▶ Management machine:
WebSphere Everyplace Device Management V5.0 Fix Pack 1 with OSGi component

Note: These software packages are included with IBM WebSphere RFID CD package. Chapter 6, “Installing the WebSphere RFID solution” on page 111 describes the installation of these prerequisites. Should you have these prerequisites already installed on your platform, verify that they fit the specific steps that are described in this chapter.

You might optionally need the WebSphere Application Server Network Deployment software packages when distributed environment is required. You might optionally need the Tivoli software packages when network monitoring and deployment is required.

The optional software packages installation are not discussed in this chapter. You can refer to chapter Chapter 10, “Monitoring” on page 213 to get details about how to install Premises Server using Tivoli software. Refer to Chapter 11, “Edge Controller Software installation and management” on page 219 to get details on WebSphere Everyplace Device Manager installation.

5.1.3 Edge Controller requirements

IBM WebSphere RFID Edge Controller software has the following software requirements:

- ▶ IBM WebSphere Everyplace Custom Environment V5.7.1
- ▶ IBM Service Management Framework V3.6
- ▶ SyncML/DM OSGi Agent V1.5.1

Any hardware device that is able to run the Java2 Micro Edition (J2ME) Connected Device Configuration (CDC) Configuration and Foundation Profile should be eligible to run the IBM WebSphere RFID Edge Controller software.

The IBM WebSphere RFID Edge Controller software currently supports the Arcom Viper Industrial Compact Enclosure Edge Controller hardware device. To get details regarding this enclosure, go to:

<http://www.arcom.com/ibm/rfid-edge-controller-IC.htm>

Refer to the following to get an up-to-date list of the supported Edge Controller hardware devices:

http://www.ibm.com/software/pervasive/ws_rfid_premises_server/technical_details

5.1.4 Supported RFID Edge Domain devices

The Edge Controller is a device, located at the edge domain of the RFID system, which controls devices such as I/O devices and RFID readers, and performs software processing towards the Premises Server.

The software components of the Edge Controller related to these devices include:

- ▶ Reader agents, which control the RFID readers
- ▶ Reader adapters, which use the RFID reader API to control the RFID reader

- ▶ I/O agents, which control the I/O devices
- ▶ I/O adapters, which use I/O device API to control the I/O devices

The I/O devices and RFID readers are linked together physically as shown in Figure 5-1.

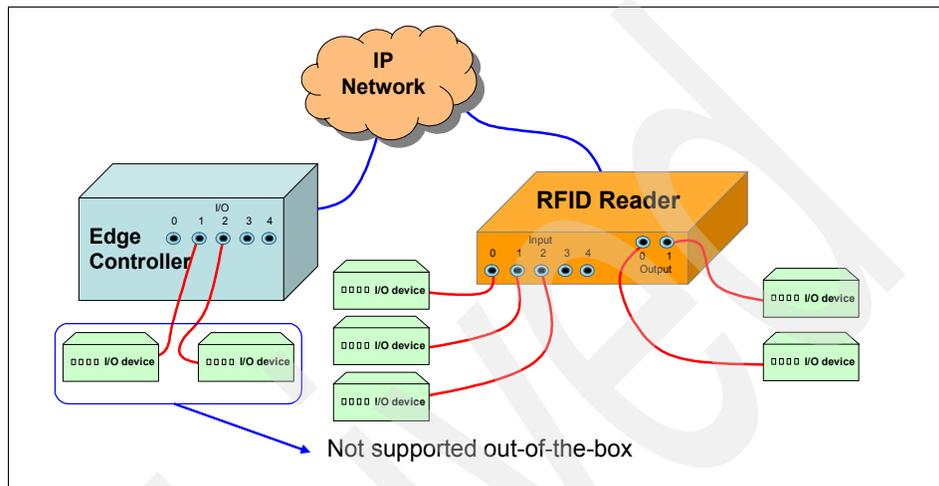


Figure 5-1 Edge Domain physical layout

The I/O devices are controlled physically by one of the following:

- ▶ The Edge Controller
 - In such a case, the piece of software controlling the I/O device is the Edge Controller I/O agent.
- ▶ The RFID reader
 - In such a case, the piece of software controlling the I/O device is the Edge Controller specific RFID reader agent.

I/O devices physically connected to the Edge Controller are not supported presently by the Edge Controller RFID without modifying the software.

The following RFID readers are supported presently by the IBM WebSphere RFID Edge Controller software:

- ▶ Alien ALR9780
- ▶ Intermec IF5
- ▶ Intermec ITRF
- ▶ Samsys MP9320 2.7EPC
- ▶ Symbol Matrics AR400

The following RFID readers are presently supported *as is* by the Edge software:

- ▶ Asyst ATR 9100
- ▶ Feig ISC.LRU1000
- ▶ Samsys MO9320 2.8EPC
- ▶ Symbol Matrics RDR-001
- ▶ Symbol Matrics XR400
- ▶ Tagsys medio L100

As is means that these readers driving software are delivered as is and are not supported.

Refer to the following to get an up-to-date list of the supported RFID readers:

http://www.ibm.com/software/pervasive/ws_rfid_premises_server/technical_details

5.2 Defining your RFID network topology

During the installation process, the so-called *network topology* is defined using the Premises Server Administrative Console.

The RFID network topology contains important information about the devices of the controlled RFID network, which are stored in the Premises Server configuration database. This information will be further retrieved by the Edge Controller and used to build its software stack according to what was actually defined.

The RFID network topology introduces the concepts of *location ID*, *reader ID*, and *Edge Controller ID*, which are defined in Table 5-1.

Table 5-1 RFID Network Topology Concepts

Concept	Meaning
Location ID	Defines the set of devices which are in the same area.
Reader ID	Defines the RFID readers in this area.
Edge Controller ID	Defines the Edge Controller driving these locations.

Refer to Chapter 7, “Administering the WebSphere RFID solution” on page 147 to get more details about these concepts.

5.2.1 Edge scalability

The Edge domain holds the highest degree of scalability. While it is possible to add a virtually unlimited number of controllers to a solution topology, it is

recommended that no more than 30 controllers be configured to communicate with a single Premises Server at any given time. Premises Server stability has been tested and maintained up to this level of topology complexity. It should be considered somewhat of a supported scalability threshold.

5.2.2 Device scalability

The Device domain, whether it be comprised of readers, printers, or any other RFID device, also has the potential to be highly scalable. However, the scalability of this domain is restricted by the computing power of the Edge domain because it is limited in the number of devices that it can communicate with simultaneously. While it is possible to add a virtually unlimited number of devices to a solution topology, it is recommended that no more than three be configured to communicate with a single Edge Controller at any given time. Satisfactory Edge operation has been verified at this level of topology complexity and it should be considered somewhat of a supported scalability threshold.

5.2.3 Supported layout

The supported layout is then to have the following distribution:

- ▶ One root location
- ▶ One location per reader
- ▶ Up to three RFID readers of the same type per Edge Controller
- ▶ Up to 30 Edge Controller per Premises Server

Note: It is recommended to one use distinct Edge Controller for each of the RFID reader types used in the solution. Indeed, downloading several types of RFID reader agents on the same Edge Controller would require custom programming for the Edge Controller.

Regarding the network, the following layout is supported:

- ▶ Each reader must have its own fixed IP address.
- ▶ Each Edge Controller must have its own fixed IP address.
- ▶ The Premises Server must have its own fixed IP address and its fully qualified DNS name, for example `bc1srv5.itso.ra1.ibm.com`.

Figure 5-2 summarizes the solution topology.

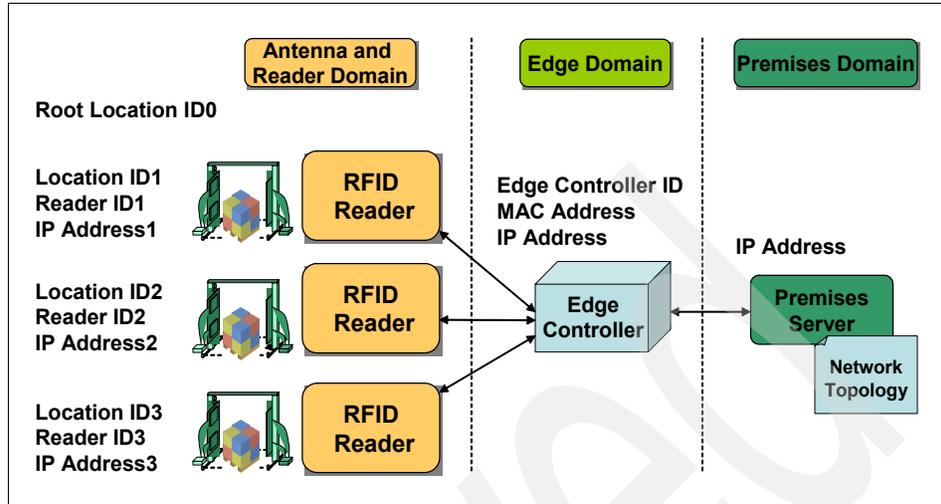


Figure 5-2 RFID solution topology

5.3 Pre-installation checklist

Table 5-2 gives a summary of the hardware items that you need to obtain prior to actually performing software installation of the IBM RFID solution.

Table 5-2 Software planning and hardware prerequisite checklist

Hardware pre-requisites	done
Premises Server: Intel® machine with following characteristics: <ul style="list-style-type: none"> • RAM: 2 GB • Processor: 3 GHz Pentium 4 • Free disk space: 8 GB • Temporary disk space during installation: 500 MB 	
Edge Controller hardware devices: One of the controllers that are defined by the following URL: http://www.ibm.com/software/pervasive/ws_rfid_premises_server/technical_details The only Edge Controller presently supported is the Arcom Viper Industrial Compact Enclosure.	

Hardware pre-requisites	done
<p>RFID readers: One of the readers that are defined by the following URL: http://www.ibm.com/software/pervasive/ws_rfid_premises_server/technical_details</p> <p>The presently supported RFID readers are:</p> <ul style="list-style-type: none"> • Alien ALR9780 • Intermec IF5 • Intermec ITRF • Samsys MP9320 2.7EPC • Symbol Matrics AR400 <p>The following readers are <i>as is</i> and are currently unsupported at the time of this writing:</p> <ul style="list-style-type: none"> • Asyst ATR 9100 • Feig ISC.LRU1000 • Samsys MO9320 2.8EPC • Symbol Matrics RDR-001 • Symbol Matrics XR400 • Tagsys medio L100 	
I/O devices: The ones supported by the chosen RFID readers	

Table 5-3 gives a summary of the hardware items which need to be obtained prior to actually performing software installation of the IBM RFID solution.

Table 5-3 Software planning - needed software packages checklist

Software packages	done
<p>Premises Server:</p> <p>Operating System: one of the following</p> <ul style="list-style-type: none"> • Windows 2000 Server Service Pack 4 • Windows Server 2003 Service Pack 1 <p>Ensure the following software programs are included in the IBM RFID Premises Server installation package:</p> <ul style="list-style-type: none"> • WebSphere Application Server V5.1 Fix Pack 1 • DB2 Workgroup Server V8.1.4 • WebSphere MQSeries V5.3.0.8 <p>Optional software programs:</p> <ul style="list-style-type: none"> • WebSphere Application Server Network Deployment V5.1 Fix Pack 1 • Tivoli Configuration Manager V4.2.1 • Tivoli Enterprise Console V3.9 • Tivoli Monitoring V5.1.2 • Tivoli Monitoring for Web Infrastructure Client V5.1.2 • Tivoli Monitoring for Database Servers Client V5.1.0 • Tivoli Monitoring for Business Integration V5.1.1 	
<p>Edge Controller:</p> <p>Ensure the following software programs are loaded on the Edge Controller hardware devices:</p> <ul style="list-style-type: none"> • IBM WebSphere Everyplace Custom Environment V5.7.1 • IBM Service Management Framework V3.6 • SyncML/DM OSGi Agent V1.5.1 	

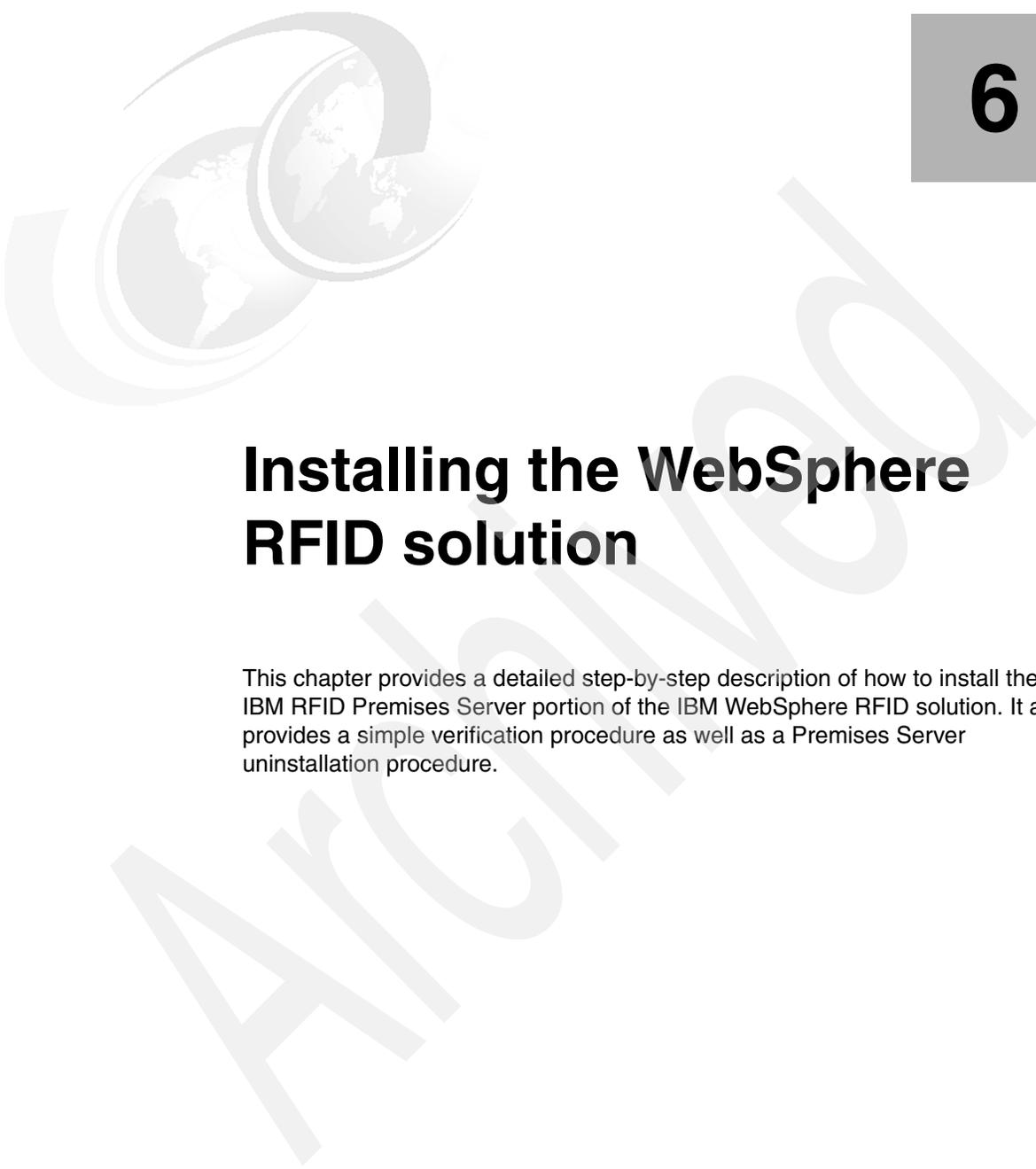
Note: Refer to 11.1.1, “WebSphere Everyplace Device Manager prerequisites” on page 220 to get the WebSphere Everyplace Device Manager installation checklist.

Table 5-4 gives a summary of the topology items that need to be defined prior to actually performing software installation of the IBM RFID solution.

Table 5-4 Software planning - needed topology items checklist

Topology item	done
RFID Topology: The following identifiers need to be defined: <ul style="list-style-type: none"> • Root location identifier • One location identifier per RFID reader • RFID reader identifier 	
Network topology: The following items need to be defined: <ul style="list-style-type: none"> • Premises Server fixed IP address • Premises Server DNS name • Edge Controller fixed IP address • RFID reader fixed IP address Network topology: The following addresses need to be obtained: <ul style="list-style-type: none"> • Edge Controller MAC address 	

Note: This chapter describes the planning tasks that are related to software installation. Planning an RFID solution also implies performing a site survey to better understand how the RFID devices can be laid out for best efficiency. Performing a site survey is discussed briefly in 1.4, “RFID solution design considerations” on page 30.



Installing the WebSphere RFID solution

This chapter provides a detailed step-by-step description of how to install the IBM RFID Premises Server portion of the IBM WebSphere RFID solution. It also provides a simple verification procedure as well as a Premises Server uninstallation procedure.

6.1 Installing the Premises Server software

This section provides instructions on how to install the Premises Server. It describes briefly the prerequisite software installation and more extensively, the installation steps specific to the Premises Server. Refer to Chapter 5., “Planning your WebSphere RFID solution” on page 101 for a detailed description of the WebSphere RFID solution hardware and software prerequisites as well as the necessary planning tasks.

The Premises Server machine must have a fixed IP address, a host name, and a DNS name. These should have been configured during the Windows operating system installation process. The Premises Server requires a database, which can be DB2 or Oracle. Only the DB2 installation process is presented in this chapter. Refer to the WebSphere RFID V1.0.2 Information Center for information about the Oracle installation steps, which is available at:

<http://publib.boulder.ibm.com/infocenter/pvcsensa/v1r2/index.jsp>

The IBM WebSphere RFID Premises Server installation package consists of twelve CDs, as shown in Table 6-1.

Table 6-1 IBM WebSphere RFID Premises Server CD package layout

CD #	CD Content
1	Documentation and Information Center
2	WebSphere Application Server V5.1 base
3	WebSphere Application Server V5.1 Network Deployment
4	WebSphere Application Server V5.1 Fix Pack 1
5	DB2 Universal Database V8.1.4 Workgroup Server Edition
6	WebSphere MQSeries V5.3.0.8 and Corrective Service Disk (CSD) 08
7	WebSphere Application Server V5.1 software package blocks for Tivoli Configuration Management
8	Network Deployment, WebSphere MQ software package blocks for Tivoli Configuration Management
9	WebSphere Application Server V5.1 Fix Pack 1 software package blocks for Tivoli Configuration Management
10	DB2 V8.1.4 software package blocks for Tivoli Configuration Management
11	RFID Premises Server V1.0.2
12	WebSphere Everyplace Device Management V5.0 Fix Pack 1 and OSGi support component

This chapter describes the mandatory installation steps, for which you need CD 2, CD 4, CD 5, CD 6, and CD 11.

The following is a synopsis of the Premises Server installation steps, which are described further in this chapter:

1. Copy the RFID Premises Server CD 11 to a local directory and make it read-write.
2. Verify the I/O ports proper settings.
3. Install the DB2 database.
4. Create the BMRFID database.
5. Install WebSphere Application Server and its fix pack.
6. Install WebSphere MQ and its Corrective Service Diskette (CSD).
7. Configure the environment variables.
8. Modify the properties files.
9. Run the Premises Server installation script.
10. Install Service Management Framework stack as a service.

Note: The Information Center for IBM WebSphere RFID Premises Server describes one step after the CD 11 copy where the I/O ports need to be verified. This step does not need to be done. The Premises Server Administrative Console now allows for an update of these I/O ports, if necessary, after the Premises Server is installed.

The sections that follow describe the steps to install the Premises Server.

6.1.1 Copy the Premises Server CD to a local directory

Copy the RFID Premises Server CD 11 to a local directory. Make sure this is a local directory and all its folders are read-write. Follow these steps:

1. From a Windows explorer window, right-click the local directory and select **Properties**.
2. Deselect **Read-only**, which is located at the bottom of the General tab and click **OK**.
3. In the Confirm Attribute Changes window, select **Apply changes to this folder, subfolders and files** and click **OK**.

In the remainder of this chapter, the RFID installation directory is referred to as *RFID_INSTALL_DIR*. Substitute your chosen local directory path in place of the *RFID_INSTALL_DIR* variable.

6.1.2 Install DB2 Universal Database

Install DB2 Universal Database V8.1.4 from installation package CD 5 by following these steps:

1. Execute the following:
[CD5] \DB2WSE814\wse\setup.exe
2. Perform a typical installation and use the default installation directories that are suggested during the installation.
3. Take note of the DB2 administrative user name and password that are created during the DB2 installation, because they are used later in the installation (in items 3 on page 115 and 5 on page 124).

6.1.3 Create the IBMRFID database

To create the IBMRFID database, follow these steps:

1. Open the DB2 database creation wizard:
 - a. Select **Start** → **Programs** → **IBM DB2** → **General Admin Tools** → **Control Center** to open the DB2 Control Center.
 - b. From the Control Center, select **All Catalog Systems** → **hostname** → **Instances** → **DB2**, then right-click **Databases** and select **Create** → **Database using wizard**.
2. Create the database:
 - a. Set database name as IBMRFID, then select **Finish**.
 - b. After several minutes, the database is created and a DB2 message asks whether Configuration Advisor should be run. Select **Yes** to tune the database.
3. Tune the database:
 - a. Select **Server** on the left pane of the Configuration Advisor wizard.
 - b. Slide the **Target** memory slider to 50%, then select **Finish**. A DB2 message should appear, which states that the command completed successfully. Close the window, then exit Control Center.

6.1.4 Import the IBMRFID database script files

To import the IBMRFID database script files, follow these steps:

1. Select **Start** → **Programs** → **IBM DB2** → **Command Line Tools** → **Command Center** to open the DB2 Command Center.
2. Click **Interactive**.

3. Connect to the IBMRFID database by typing the following into the Command field:

`connect to IBMRFID user userid using password`

where *userid* and *password* are the DB2 administration user ID and password that you defined during the DB2 installation in 6.1.2, “Install DB2 Universal Database” on page 114. You should now see a window similar to that shown in Figure 6-1.

4. Select **Interactive** → **Execute** from the top menu to execute the DB2 connection command.

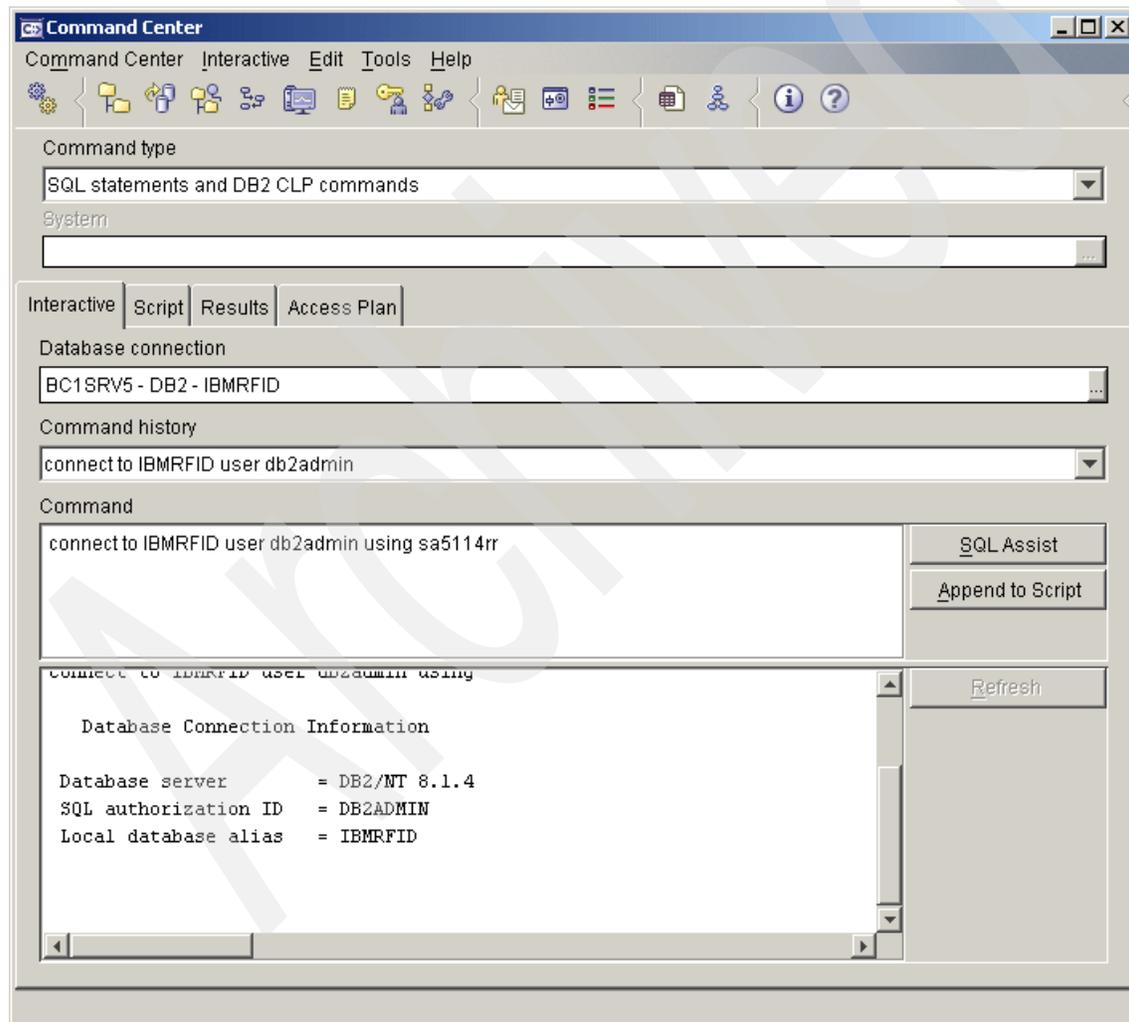


Figure 6-1 Connect to IBMRFID database using DB2 Command Center

5. Select **Script**, then select **Script** → **Import** from the top menu. You should see a window similar to that shown in Figure 6-2.
6. Select **File System**, then the host name from the System name menu.
7. Select the *RFID_INSTALL_DIR\IBM\RFID\premises\eventserver\db* directory and then the *ibm-premises-db2-dbdef.ddl* file. Then select **OK**.

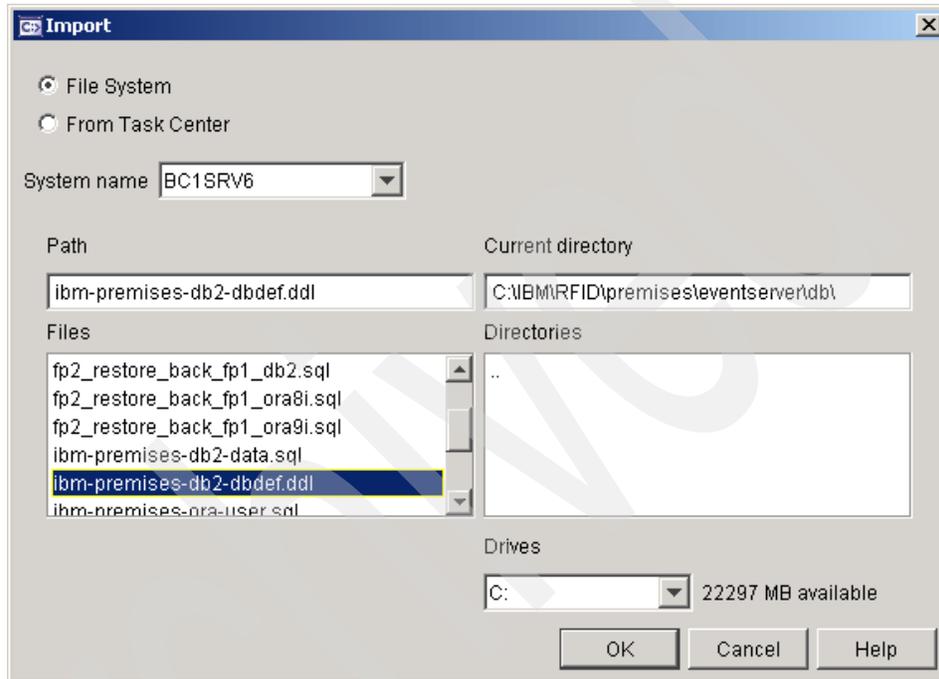


Figure 6-2 Importing RFID application DDL file

8. Select **Script** → **Execute** from the top menu to execute the script.
9. Repeat steps 5 on page 116 through 8 with the *ibm-premises-db2-data.sql* file to run the IBMRfid database SQL script.
10. Exit the DB2 Command Center.

6.1.5 Install WebSphere Application Server

Install WebSphere Application Server V5.1 from installation package CD 2 by following these steps:

1. Execute the following command:
`[CD2]\WAS51BASE\win\Install.exe`
2. Select the custom installation, then disable the installation of the Embedded Messaging component.

Important: Should you have an already existing WebSphere Application Server installation for which Embedded Messaging component was installed, it is recommended to re-install WebSphere Application Server.

3. You can then keep the default settings and use the suggested default directories.
4. Choose to run WebSphere Application Server and IBM HTTP Server as a Windows service.

The installation can last up to 10 minutes.

6.1.6 Install WebSphere Application Server Fix Pack

Install WebSphere Application Server V5.1 Fix Pack 1 from CD 4 of the installation package by following these steps:

1. Stop all the Java processes that use the IBM Software Development Kit. They should be WebSphere Application Server and IBM HTTP Server. You can check that no Java processes are running using the Windows Task Manager.
2. If necessary, remove the WebSphere MQ tray icon, which stops the `amqmtbrn.exe` process. To stop the process, click the WebSphere MQ tray icon and select **Stop WebSphere MQ** as shown in Figure 6-3.



Figure 6-3 Removing WebSphere MQ tray icon

3. From a command prompt:

- Execute the following command:

```
WAS_HOME\bin\setupCmdLine.bat
```

This command updates the necessary Java variables.

- Copy [CD4]WAS51FP1 to C:\WAS51FP1.

Restriction: Do not attempt to run the updateWizard.bat command directly from the CD-ROM drive as the updateWizard attempts to write to the location that it launched from. If you try running the updateWizard.bat command directly from the CD-ROM drive, it will fail with a access denied message.

- Execute the following to open the fix pack installation wizard:

```
C:\WAS51FP1\updateWizard.bat
```

You should see a window similar to Figure 6-4.

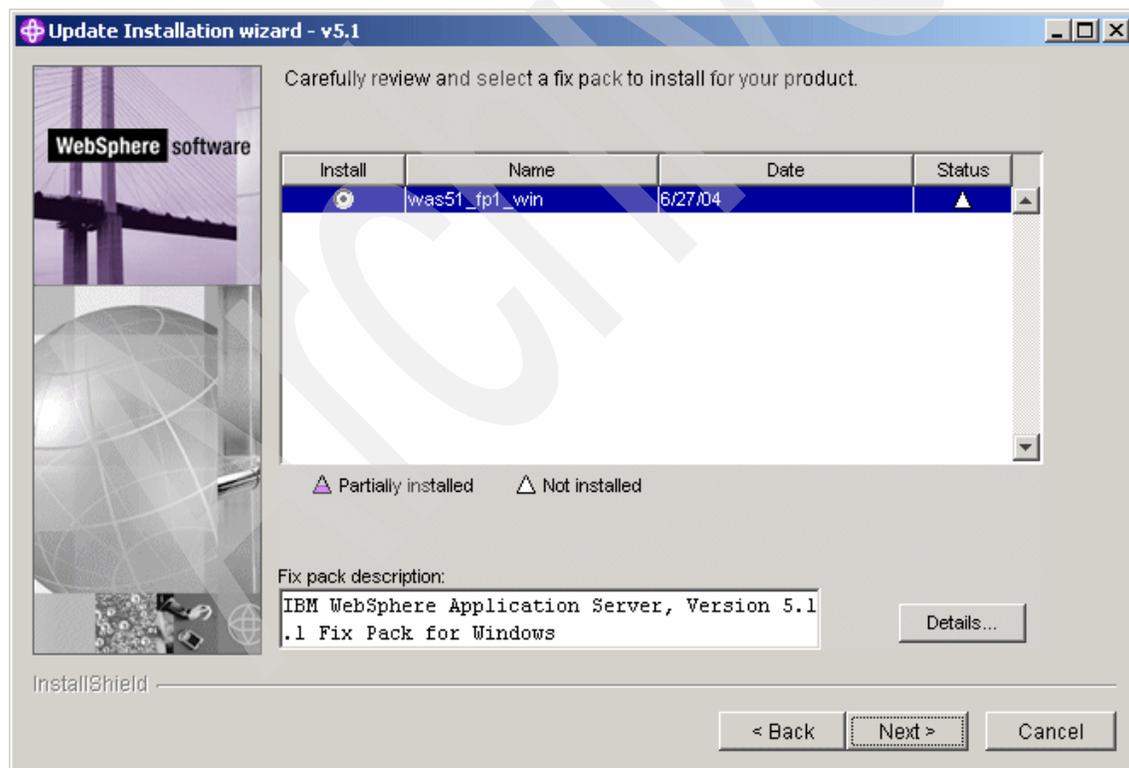


Figure 6-4 Update installation wizard window

4. After the wizard opens, perform the following:
 - Select **English** and click **OK**, then **Next**.
 - Select **WebSphere Application Server 5.1.0**, then click **Next**.
 - Select **install fix packs** and click **Next**.
 - Set fix pack directory to [CD4]\WAS51FP1\fixpacks and click **Next**.
 - Leave the defaults directory and embedded messaging settings and click **Next** twice to run the update.

The installation process can last more than five minutes. When the run is complete, click **Finish**.

Note: If you have an installation failure and need to restart the process, you can run the update wizard again to uninstall the fix pack and then run it again to re-install the fix pack.

5. Make sure the Windows service IBM WebSphere Application Server V5 - server1 is set to start automatically as shown in Figure 6-5.

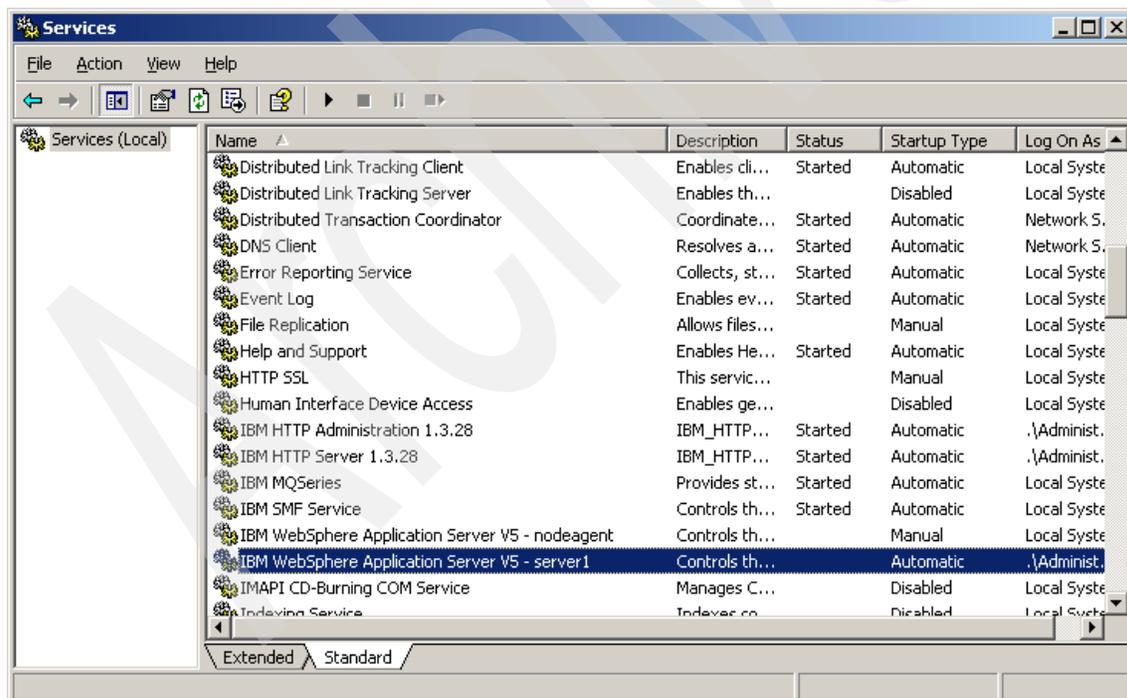


Figure 6-5 WebSphere Application Server set to start automatically
Screen shot reprinted by permission from Microsoft Corporation

6.1.7 Install WebSphere MQ

Install WebSphere MQ V5.3.0.8 from installation package CD 6 by following these steps:

1. Verify that the WebSphere Application Server Java path (*WAS_HOME*\java\bin) is in the Windows %PATH% variable to make Java recognized by the system.

If the variable is not set, then perform the following:

- Select **Start** → **Settings** → **Control Panel** → **System**.
- Select **Advanced**.
- Select **Environment Variables**.
- Select the **Path** System variable, and select **Edit**.
- Add path *WAS_HOME*\java\bin and click **OK**.

2. Execute the following from a command prompt to start the installation launchpad:

```
[CD6]\MQ5302\MQ5302\setup.exe
```

3. Chose option 3, **WebSphere MQ Installation**, then click **Launch WebSphere MQ Installer**.

Note: Ignore the software prerequisites error where Java JRE V1.3 or later is not found. The installation will still succeed.

4. Click **Next** and accept the agreements.
5. Chose custom installation option and click **Next**.
6. Leave the default directory setting and click **Next** three times. You should see a window similar to that shown in Figure 6-6.
7. Select **Java Messaging** and **Windows Client** options in the installation wizard and click **Next**, then **Install**.

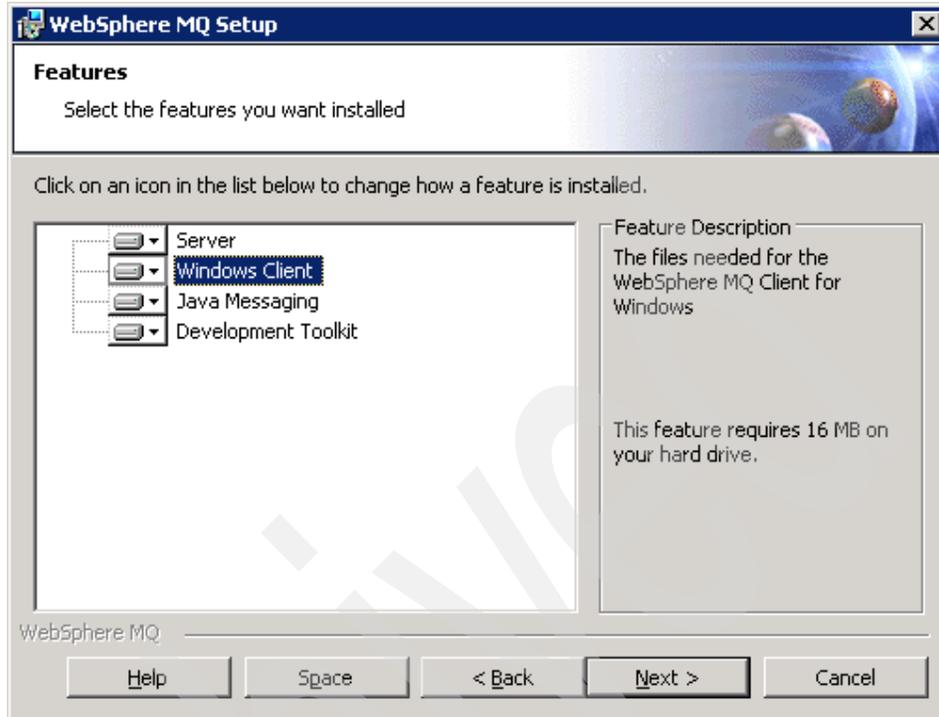


Figure 6-6 WebSphere MQ Setup settings

8. Select **Yes** when prompted for sufficient license units.
The installation begins, it can last several minutes.
9. Select **Finish** when the installation is complete.
10. There is no need to go through the Prepare WebSphere MQ Wizard to configure the default Queue Managers. Select **Cancel**.

6.1.8 Install WebSphere MQ CSD

Install WebSphere MQ V5.3.0.8 CSD08 from installation package CD 6 by following these steps:

1. Execute the following from a command prompt:
[CD6]\MQ5302\MQ_CSD08\U200215A.exe
2. Keep the default settings and proceed to the installation. You can use the suggested default directories.

The installation should last several minutes.

6.1.9 Configure the environment variables

To configure the environment variables, follow these steps:

1. Select **Start** → **Settings** → **Control Panel** → **System** to open the System Properties window.
2. Select the **Advanced** tab, then **Environment Variables** to open the Environment Variable window.
3. Ensure that CLASSPATH is a System variable.
4. Ensure that the following values are in the CLASSPATH variable:

```
C:\Program Files\IBM\WebSphere MQ\Java\lib\providerutil.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\com.ibm.mqjms.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\ldap.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\jta.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\jndi.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\jms.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\connector.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\fscontext.jar
C:\Program Files\IBM\WebSphere MQ\Java\lib\com.ibm.mq.jar
```

Note: Single quotation marks (') or double quotation marks (" ") should not appear in these CLASSPATH variables.

5. Ensure that values in the PATH variable contain the following value:
C:\Program Files\IBM\WebSphere MQ\Java\lib
6. Add system environment variable IVEHOME with the value set to the WebSphere Application Server installation directory, such as C:\Program Files\WebSphere\Appserver as shown in Figure 6-7.

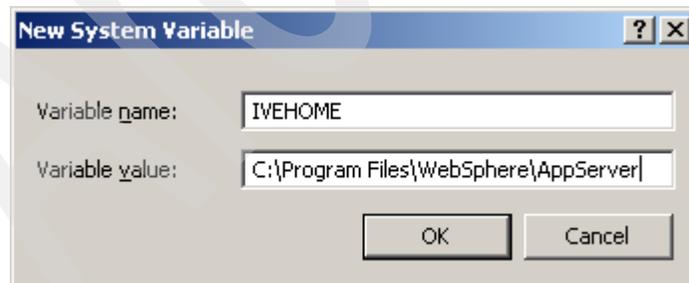


Figure 6-7 IVEHOME variable
Screen shot reprinted by permission from Microsoft Corporation

7. Add system environment variable MQ_JAVA_DATA_PATH with the value set to the WebSphere MQ installation directory (for example, C:\Program Files\IBM\WebSphere MQ) as shown in Figure 6-8.

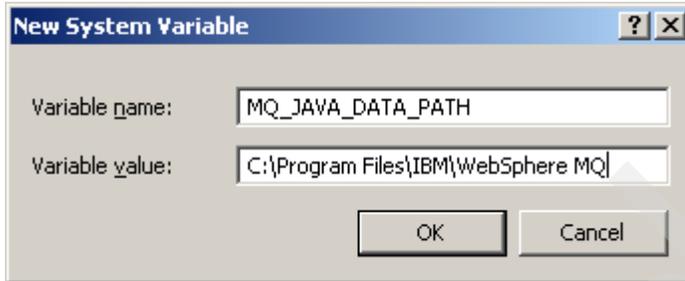


Figure 6-8 MQ_JAVA_DATA_PATH variable
Screen shot reprinted by permission from Microsoft Corporation

8. Add system environment variable MQ_JAVA_INSTALL_PATH with the value set to the WebSphere MQ installation Java directory (for example, C:\Program Files\IBM\WebSphere MQ\java) as shown in Figure 6-9.

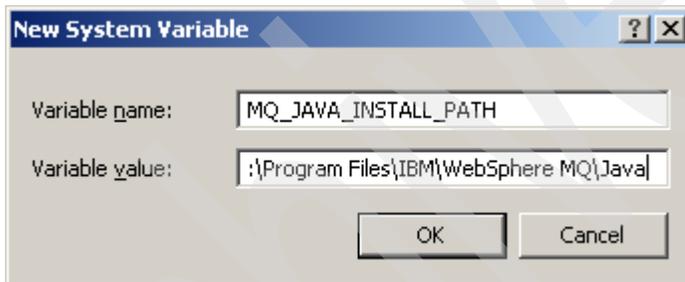


Figure 6-9 MQ_JAVA_INSTALL_PATH variable
Screen shot reprinted by permission from Microsoft Corporation

9. Add system environment variable IBM_RFID_HOME with the value set to RFID_INSTALL_DIR\IBM, as shown in Figure 6-10.

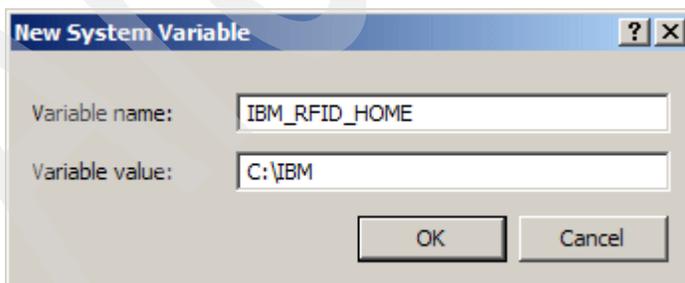


Figure 6-10 IBM_RFID_HOME variable
Screen shot reprinted by permission from Microsoft Corporation

6.1.10 Modify the Event Server configuration properties

The Event Server configuration properties are in file `event_svr_config.properties`, which are located in the following directory:

```
RFID_INSTALL_DIR\IBM\RFID\premises\install\event_server
```

To modify the Event Server configuration properties, do the following:

1. It is a good idea to save the original properties file as `event_svr_config.properties.orig` in case you would need to go back to the original file.
2. Modify all instances of `HOSTNAME` or `hostname` and set them to the short host name of the machine on which you are installing the Premises Server.
3. Modify the following variable:

```
wasvariable.1.value=MQ INSTALL DIRECTORY
```

Set this value to be the installation location of IBM WebSphere MQ (for example, `C:/Program Files/IBM/WebSphere MQ`).

Use the slash (/) and not the backslash (\) for directory paths.

4. Modify the following variable:

```
wasvariable.2.value=DB2 SQLLIB INSTALL DIRECTORY/java
```

This is the directory where the DB2 JDBC driver is found (for example, `C:/Program Files/IBM/SQLLIB`).

Use the slash (/) and not the backslash (\) for directory paths.

5. Modify the following variable:

```
jaasAuthAlias.0.user=DB_UserName
```

`DB_UserName` is the DB user ID that you defined in item 3 on page 114.

6. Modify the following variable:

```
jaasAuthAlias.0.password=DB_password
```

`DB_password` is the DB user password that you defined in item 3 on page 114.

7. Do *not* uncomment the following lines because WebSphere Application Server is installed in a non-distributed environment:

```
# nodeName <hostname>  
# cellName <hostname>
```

8. Modify the following variable:

```
jvm.0.property.value=IBM_RFID_HOME/RFID/premises/eventserver/  
properties/premises.properties
```

Replace *IBM_RFID_HOME* with *RFID_INSTALL_DIR/IBM*.

Note: You need to perform additional steps here when WebSphere Application Server is installed in a distributed environment. Refer to IBM WebSphere RFID Premises Server information center.

6.1.11 Run the RFID installation script

To run the RFID installation script, do the following:

1. Ensure that IBM HTTP Server, IBM WebSphere Application Server V5 - server1, and IBM MQ Series services are running.

2. From a command prompt, execute the following command:

```
WAS_HOME\bin\setupCmdLine.bat
```

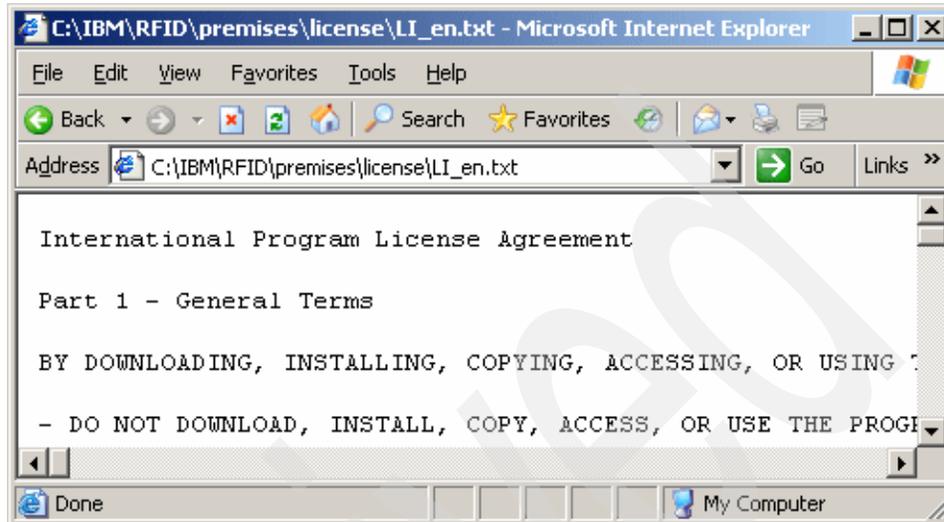
This command updates the required Java variables.

3. From the same command prompt, run *install.bat* that is located in the *RFID_INSTALL_DIR/IBM/RFID/premises* directory. You should get output that is similar to that shown in Example 6-1.

Example 6-1 RFID Installation script log - part 1

```
C:\IBM\RFID\premises>Install.bat  
Please read the License agreement.....  
Please Hit Ctrl+C if you do not agree with this license. Otherwise, hit Enter  
Press any key to continue . . .
```

- An Internet Explorer window opens that displays the licence agreements, as shown in Figure 6-11.



*Figure 6-11 IBM WebSphere RFID Premises Server license agreements
Screen shot reprinted by permission from Microsoft Corporation*

- Read the license agreements and accept them by pressing any key from the install.bat command prompt.
- The installation should last several minutes, and you should then see output that is similar to that shown in Example 6-2.

Example 6-2 RFID installation script log - part 2

```
Installing MQ resources
Installing WAS resources
Installing Event Server....Be patient...this may take a while
Installing kimono
Install Kimono MQ resources
Install Kimono Configuration and Application
Restarting WAS to complete installation
WAS restarted
Installation complete
```

7. Verify that the installation was successful in the log file, `RFID_INSTALL_DIR\RFID\logs\install.log`. This log file is rather long, but you should verify that the main phases listed in Example 6-3 show successful completion.

Example 6-3 Installation log phases

```
----- Starting creation of RFID MQ objects -----  
----- Creating MQ Queue Manager IBM.RFID.QM -----  
----- Finished creating MQ Queue Manager IBM.RFID.QM -----  
----- Setting MQ Queue Manager IBM.RFID.QM to autostart -----  
----- Finished setting MQ Queue Manager IBM.RFID.QM to autostart -----  
----- Starting MQ Queue Manager IBM.RFID.QM -----  
----- Started MQ Queue Manager IBM.RFID.QM -----  
----- Finished creation of RFID MQ objects -----  
----- Starting creation of Kimono RFID MQ objects -----  
----- Starting MQ Queue Manager IBM.RFID.QM -----  
----- Started MQ Queue Manager IBM.RFID.QM -----  
One valid MQSC command could not be processed.  
----- Finished creation of Kimono RFID MQ objects -----  
ADMU3000I: Server server1 open for e-business; process id is 3788
```

Note: If you encounter an installation problem during this phase, uninstall the RFID Premises Server using the `uninstall.bat` command, correct the problem, and then launch `install.bat` again.

6.1.12 Install the SMF stack as a Windows service

To install the SMF stack as a Windows service, execute the following steps:

1. From a command prompt, execute the following:

```
RFID_INSTALL_DIR\IBM\RFID\edgecontroller\premises\smf\smf_service -install
```

You should see messages similar to those shown in Example 6-4.

Example 6-4 SMF as a service installation log

```
C:\IBM\RFID\edgecontroller\premises\smf>smf_service -install  
smf_service: Calling ServiceInstall()  
smf_service: ServiceInstall() successful
```

Note: Should you encounter a problem during this phase, you can remove the IBM SMF Service using the following command:

```
RFID_INSTALL_DIR\IBM\RFID\edgecontroller\premises\smf\  
smf_service -uninstall
```

You can then correct the problem and run the installation procedure again (beginning with step 1 on page 127 to step 8 on page 130).

- The service, IBM SMF Service, should show on the Services Control Panel.
2. Select **Start** → **Run** → **regedt32.exe** to open the Windows Registry Editor.
 3. Go to the following key, as shown in Figure 6-12:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IBMSMFService

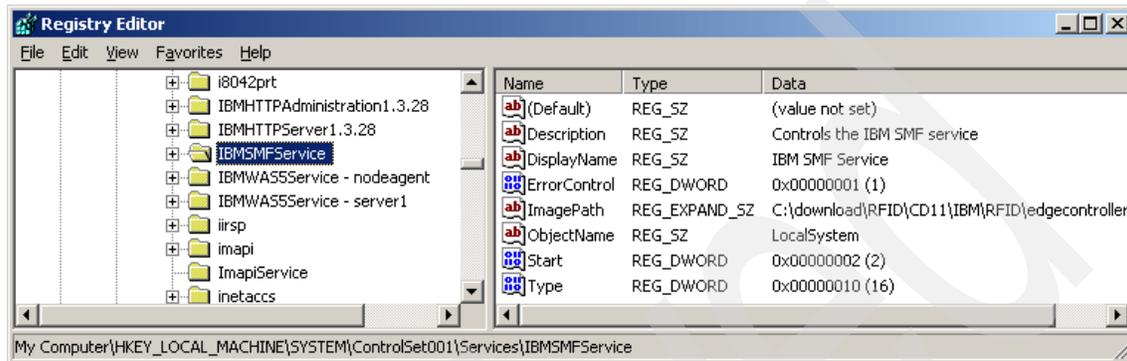


Figure 6-12 IBMSMFService registry key
Screen shot reprinted by permission from Microsoft Corporation

4. Add a new multi-string value key, and name it DependOnService (Figure 6-13).

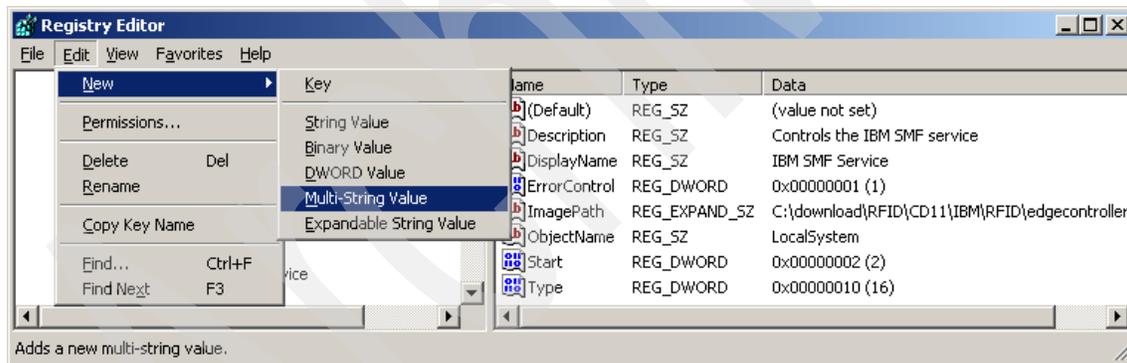


Figure 6-13 Adding a multi-string value key
Screen shot reprinted by permission from Microsoft Corporation

5. Modify the key, DependOnService, as shown in Figure 6-14.

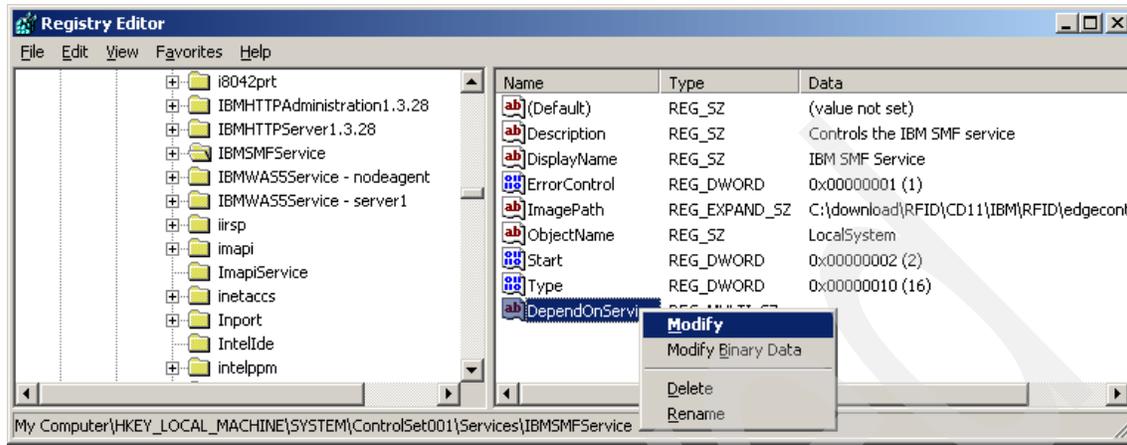


Figure 6-14 Modifying registry key `DependOnService`
Screen shot reprinted by permission from Microsoft Corporation

6. Set its value to MQSeriesServices, as shown in Figure 6-15.

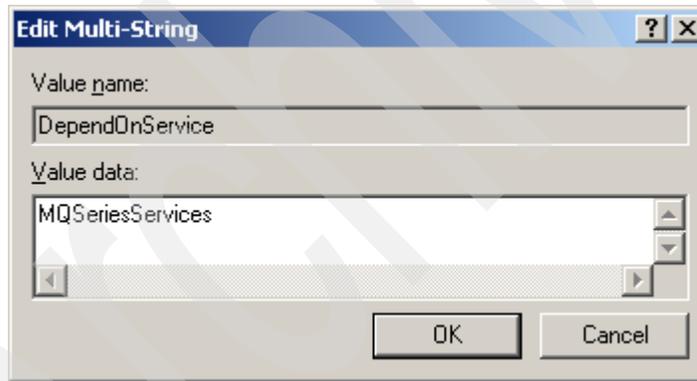


Figure 6-15 Setting multi-string key `DependOnService`

7. The new registry key now appears, as shown in Figure 6-16.

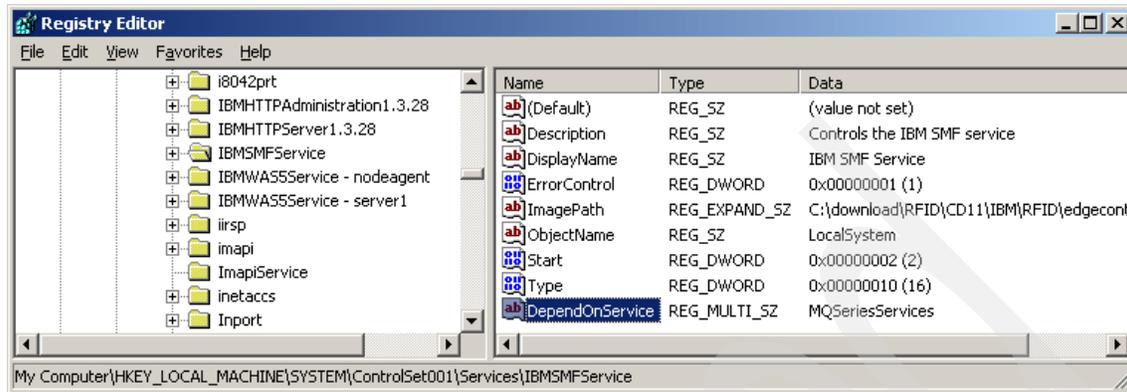


Figure 6-16 New registry key *DependOnService*
Screen shot reprinted by permission from Microsoft Corporation

8. Reboot the machine.

6.2 Verifying the Premises Server installation

This section discusses how to verify that the Premises Server installed successfully.

6.2.1 General purpose verifications

To verify the general installation, do the following:

1. Check the `IVEHOME` environment variable to ensure that it points to the WebSphere Application Server installation directory.
2. Make sure that the correct file paths are specified for the edge alerts and heartbeat log files in the file:

```
RFID_INSTALL_DIR\IBM\RFID\premises\eventserver\properties\  
premises.properties
```

You have to change `C:/ibm` to `RFID_INSTALL_DIR/ibm` throughout the file.

3. If necessary, start WebSphere Application Server from the services panel. Right-click **IBM WebSphere Application Server** and select **Start the service**.
4. Make sure that the IBM RFID Queue Manager is started:
 - a. Select **Start** → **Programs** → **IBM WebSphere MQ** → **WebSphere MQ Explorer** to open the WebSphere MQ console.

- b. Look for IBM.RFID.QM in the Queue Managers folder. If there is a green arrow to the left of IBM.RFID.QM, then it is running (Figure 6-17).
If it is not running, then start service IBM MQSeries from the services panel.

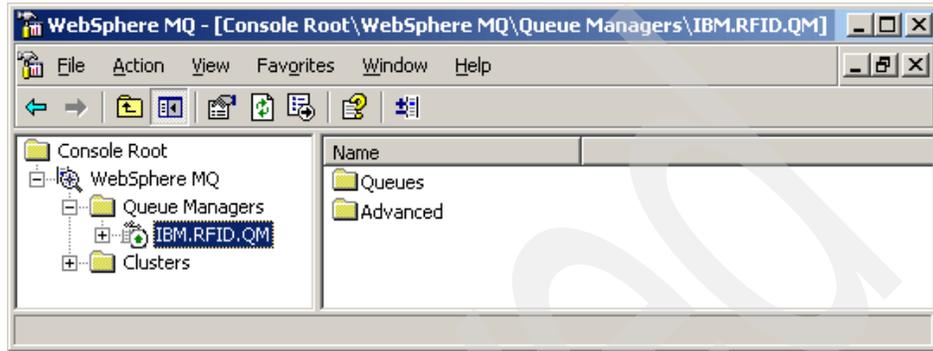


Figure 6-17 IBM.RFID.QM status on WebSphere MQ console

- c. The defined WebSphere MQ queues for the Premises Server should be similar to what is shown in Figure 6-18.

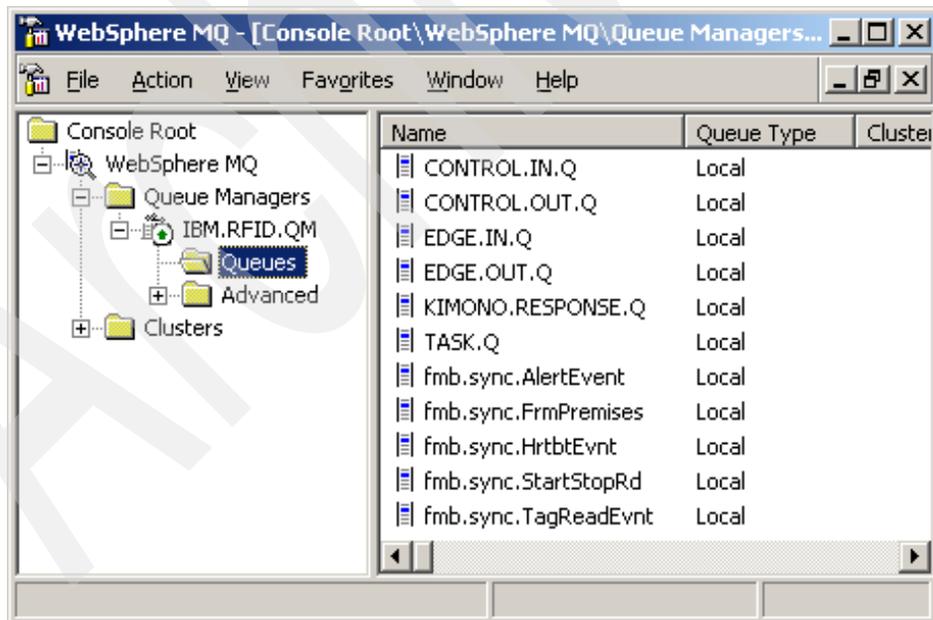


Figure 6-18 Premises Server defined WebSphere MQ queues

6.2.2 SMF stack verification

To verify the SMF stack, do the following:

1. Start the SMF service manually:
 - a. If SMF has been started as a Windows service (check the Windows services panel), stop the service.
 - b. From a command prompt, execute `smf.bat` in the directory `RFID_INSTALL_DIR\IBM\RFID\edgecontroller\premises\smf`. This batch file starts the SMF service. The SMF prompt is displayed after pressing the Enter key a second time, as shown in Example 6-5.

Note: In Example 6-5, the lines highlighted in **bold** are the ones you should look for to verify that your Edge Controller started correctly.

Example 6-5 SMF stack starting log

```
C:\IBM\RFID\edgecontroller\premises\smf>smf
=====
Use port 1457 when connecting to the SMF client from the WSDD IDE.
=====
```

Note: Some Java launch log lines are not shown here.

OSGi Service Platform Release 3

Licensed Materials - Property of IBM

(C) Copyright IBM Corp. 1999, 2004 All Rights Reserved.

IBM is a registered trademark of IBM Corp.

Service Management Framework is a trademark of IBM Corp.

[INFO] 2005-09-19 18:45:16.703 - OSGi Application Framework 3.2.0

Licensed Materials - Property of IBM

(C) Copyright IBM Corp. 2001, 2005 All Rights Reserved

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

**Mon Sep 19 18:45:17 EDT 2005 I FMBC1104 MicroBroker -----MicroBroker
starting-----**

Mon Sep 19 18:45:17 EDT 2005 I FMBC1030 MicroBroker WMQTT listener created successfully:

WMQTT:ALL:1883:MQTT-1883

Mon Sep 19 18:45:17 EDT 2005 I FMBC1030 MicroBroker WMQTTLocal listener created successfully:

WMQTTLocal:MQTT-LOCAL

Mon Sep 19 18:45:17 EDT 2005 I FMBC1001 **MicroBroker Plugin initialised successfully:**
 PersistenceInterface=com.ibm.micro.persist.LoggingPersistence
 Mon Sep 19 18:45:17 EDT 2005 W FMBC1412 MicroBroker Logging persistence will recover
 publications or subscriptions after a restart,but not a crash.
 Mon Sep 19 18:45:18 EDT 2005 W FMBC1413 MicroBroker Delivery of quality of service 1,2 messages
 cannot be assured.
 Mon Sep 19 18:45:18 EDT 2005 I FMBC1100 MicroBroker MicroBroker started: 1.0.2.5 - 200507041219

```
smf> Mon Sep 19 18:45:18 EDT 2005 I FMBB2000 bridge -----Bridge
starting-----
Mon Sep 19 18:45:18 EDT 2005 I FMBB2006 bridge Initializing Route: AlertEvent
Mon Sep 19 18:45:18 EDT 2005 I FMBB2006 bridge Initializing Route: HrtbtEvt
Mon Sep 19 18:45:18 EDT 2005 I FMBB2006 bridge Initializing Route: StartStopRd
Mon Sep 19 18:45:19 EDT 2005 I FMBB2006 bridge Initializing Route: TagReadEvt
Mon Sep 19 18:45:19 EDT 2005 I FMBB2006 bridge Initializing Route: FrmPremises
Mon Sep 19 18:45:19 EDT 2005 W FMBB2061 bridge No admin connection specified
Mon Sep 19 18:45:19 EDT 2005 I FMBB2004 bridge Starting Route: AlertEvent
Mon Sep 19 18:45:20 EDT 2005 I FMBC1300 MicroBroker Client connected: AlertEvent_localbroker,
Connection: /127.0.0.1
Mon Sep 19 18:45:20 EDT 2005 I FMBB2042 bridge Connected to tcp://localhost:1883 as
AlertEvent_localbroker [Will:Bridge/status:disconnected]
Mon Sep 19 18:45:20 EDT 2005 I FMBB2043 bridge Birth Certificate published to
tcp://localhost:1883, [Birth Certificate:Bridge/status:connected]
Mon Sep 19 18:45:20 EDT 2005 W FMBC1414 MicroBroker Client 'AlertEvent_localbroker' registered
a durable subscription. The subscription is not recoverable in all circumstances. See previous
persistence messages.
Mon Sep 19 18:45:20 EDT 2005 I FMBB2004 bridge Starting Route: HrtbtEvt
Mon Sep 19 18:45:20 EDT 2005 I FMBC1300 MicroBroker Client connected: HrtbtEvt_localbroker,
Connection: /127.0.0.1
Mon Sep 19 18:45:20 EDT 2005 I FMBB2042 bridge Connected to tcp://localhost:1883 as
HrtbtEvt_localbroker [Will:Bridge/status:disconnected]
Mon Sep 19 18:45:21 EDT 2005 I FMBB2043 bridge Birth Certificate published to
tcp://localhost:1883, [Birth Certificate:Bridge/status:connected]
Mon Sep 19 18:45:21 EDT 2005 W FMBC1414 MicroBroker Client 'HrtbtEvt_localbroker' registered
a durable subscription. The subscription is not recoverable in all circumstances. See previous
persistence messages.
Mon Sep 19 18:45:21 EDT 2005 I FMBB2004 bridge Starting Route: StartStopRd
Mon Sep 19 18:45:21 EDT 2005 I FMBC1300 MicroBroker Client connected: StartStopRd_localbroker,
Connection: /127.0.0.1
Mon Sep 19 18:45:21 EDT 2005 I FMBB2042 bridge Connected to tcp://localhost:1883 as
StartStopRd_localbroker [Will:Bridge/status:disconnected]
Mon Sep 19 18:45:21 EDT 2005 I FMBB2043 bridge Birth Certificate published to
tcp://localhost:1883, [Birth Certificate:Bridge/status:connected]
Mon Sep 19 18:45:21 EDT 2005 W FMBC1414 MicroBroker Client 'StartStopRd_localbroker' registered
a durable subscription. The subscription is not recoverable in all circumstances. See previous
persistence messages.
Mon Sep 19 18:45:22 EDT 2005 I FMBB2004 bridge Starting Route: TagReadEvt
Mon Sep 19 18:45:22 EDT 2005 I FMBC1300 MicroBroker Client connected: TagReadEvt_localbroker,
Connection: /127.0.0.1
Mon Sep 19 18:45:22 EDT 2005 I FMBB2042 bridge Connected to tcp://localhost:1883 as
TagReadEvt_localbroker [Will:Bridge/status:disconnected]
```

```

Mon Sep 19 18:45:22 EDT 2005 I FMBB2043 bridge Birth Certificate published to
tcp://localhost:1883, [Birth Certificate:Bridge/status:connected]
Mon Sep 19 18:45:22 EDT 2005 W FMBC1414 MicroBroker Client 'TagReadEvt_localbroker' registered
a durable subscription. The subscription is not recoverable in all circumstances. See previous
persistence messages.
Mon Sep 19 18:45:22 EDT 2005 I FMBB2004 bridge Starting Route: FrmPremises
Mon Sep 19 18:45:23 EDT 2005 I FMBC1300 MicroBroker Client connected: FrmPremises_localbroker,
Connection: /127.0.0.1
Mon Sep 19 18:45:23 EDT 2005 I FMBB2042 bridge Connected to tcp://localhost:1883 as
FrmPremises_localbroker [Will:Bridge/status:disconnected]
Mon Sep 19 18:45:23 EDT 2005 I FMBB2043 bridge Birth Certificate published to
tcp://localhost:1883, [Birth Certificate:Bridge/status:connected]
Mon Sep 19 18:45:23 EDT 2005 I FMBB2001 bridge -----Bridge
started-----
Mon Sep 19 18:45:23 EDT 2005 I FMBN102 BridgeMgr Started Bridge named PremisesBridge.

smf>

```

6.2.3 WebSphere Application Server verifications

To verify the WebSphere Application Server installation:

1. Check for errors in the WebSphere Application Server (Premises Server) log files. The log files are in the following directories:
 - directory *RFID_INSTALL_DIR*\IBM\RFID\logs for alert error and heartbeat and SMF
 - directory *WAS_HOME*\logs\server1 for WebSphere Application Server
 - files C:\Program Files\IBM\SQLLIB\DB2\db2diag.log and C:\Program Files\IBM\SQLLIB\DB2\jdbcerr.log for DB2.
2. Make sure all WebSphere Application Server applications are running:
 - a. Launch the WebSphere Application Server Administrative Console.
 - b. Expand **Applications**.
 - c. Click **Enterprise Applications** (Figure 6-19 on page 135). The following Applications should appear with green status arrows next to them:
 - Event_Server
 - adminconsole
 - ivtApp
 - kimono_input_channel
 - kimono_tasks

If one of them is not running, then select it and select **Start** to start it.

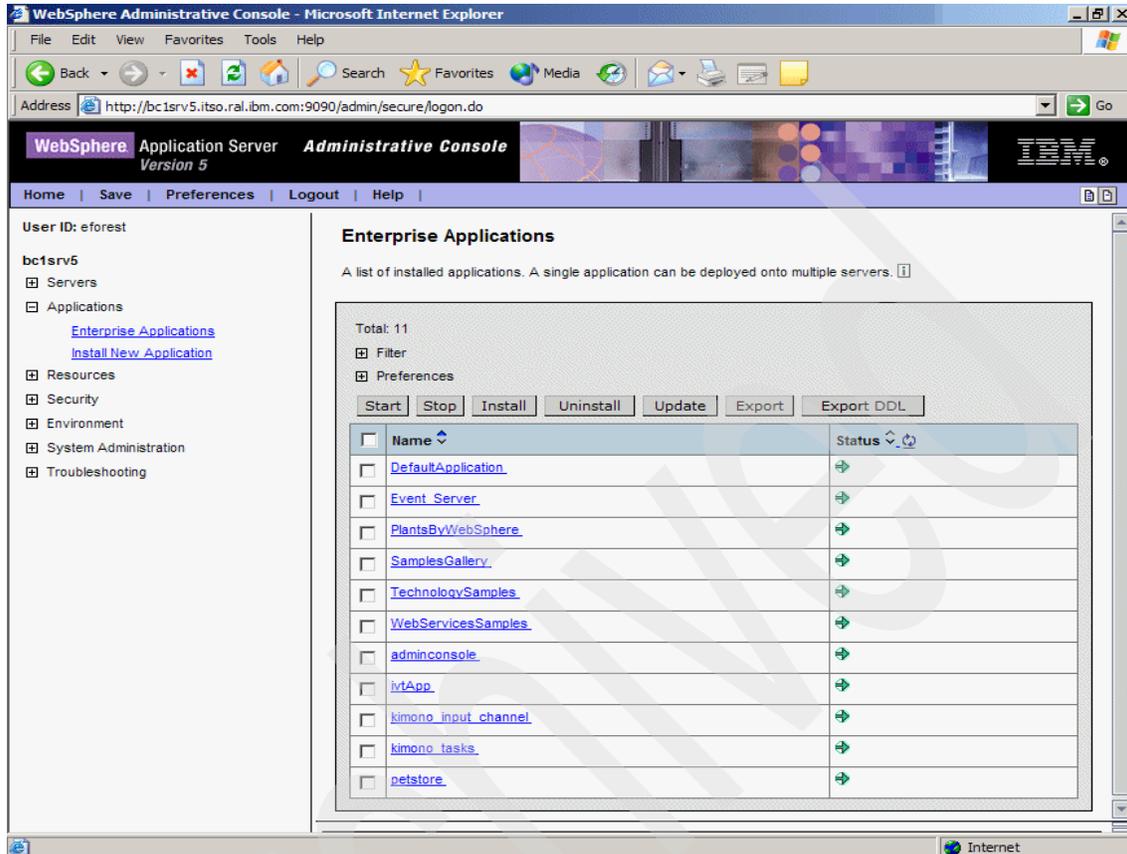


Figure 6-19 Premises Server WEB applications
Screen shot reprinted by permission from Microsoft Corporation

6.2.4 Premises Server verifications

To verify the Premises Server installation:

1. Check the Premises Server Administrative Console:

Log in to the Premises Server Administrative Console to verify that it is accessible. The default URL for the Premises Server Administrative Console is:

`http://premises_ip_address:9080/event_admin_web`

You should see something similar to that shown in Figure 6-20.



Figure 6-20 Premises Server Administrative Console
Screen shot reprinted by permission from Microsoft Corporation

2. Start the KimonoPremisesTest bundle:

- a. From the SMF command prompt, in the window where you started SMF, type `ss` to list the installed bundles. The text shown in Example 6-6 should be returned.

Example 6-6 SMF installed bundles in the Premises Server

```
smf> ss
```

Framework is launched.

id	Type	State	Bundle
27	.jar	ACTIVE	smfbd:/PremisesLoggingConnector [27]
26	.jar	RESOLVED	smfbd:/KimonoConsoleLog [26]
25	.jar	RESOLVED	smfbd:/KimonoPremisesTest [25]
24	.jar	ACTIVE	smfbd:/MBAF [24]
23	.jar	ACTIVE	smfbd:/win32service [23]
21	.jar	ACTIVE	smfbd:/MicroBrokerBridgeManager [21]
20	.jar	ACTIVE	smfbd:/KimonoPremisesBridge [20]
19	.jar	ACTIVE	smfbd:/Rfid [19]
18	.jar	ACTIVE	smfbd:/MicroBrokerBridgeJMS [18]
17	.jar	ACTIVE	smfbd:/MicroBrokerBridge [17]
15	.jar	ACTIVE	smfbd:/MicroBrokerManager [15]
14	.jar	ACTIVE	smfbd:/MicroBroker [14]
13	.jar	ACTIVE	smfbd:/ConfigurationAdmin [13]

12	.jar	ACTIVE	smfbd:/EventLog [12]
11	.jar	ACTIVE	smfbd:/MicroBrokerTrace [11]
10	.jar	ACTIVE	smfbd:/MQTelemetryTransport [10]
9	.jar	ACTIVE	smfbd:/MicroBrokerRegistry [9]
8	.jar	ACTIVE	smfbd:/OSGi-SPR3-ServiceTracker [8]
7	.jar	ACTIVE	smfbd:/LogService [7]
6	.jar	ACTIVE	smfbd:/SMFBundleMessages [6]
5	.jar	ACTIVE	smfbd:/PersistenceManager [5]
2	.jar	ACTIVE	smfbd:/OAF_Base [2]
1	.jar	ACTIVE	smfbd:/OSGi-SPR3-ServiceInterfaces [1]
0		ACTIVE	System Bundle [0]

b. To start a bundle, type the following command:

```
start bundle_number
```

To stop a bundle, type the following command:

```
stop bundle_number
```

Start the `KimonoPremisesTest` bundle by typing the following command:

```
start 25
```

The number 25 comes from the following line showing in Example 6-6 on page 136 (highlighted in bold blue):

```
25 .jar RESOLVED smfbd:/KimonoPremisesTest [25]
```

Starting the `KimonoPremisesTest` bundle enables an integrated end-to-end system simulation, including the simulation of the components external to the Premises Server.

Starting the `KimonoConsoleLog` bundle enables SMF to display all messages at this SMF prompt, which can be very useful in debugging the system.

6.2.5 Using KimonoPremisesTest bundle

The KimonoPremisesTest bundle (Figure 6-21 on page 138) allows for testing the Premises Server by simulating the domains external to the Premises Server through the three testing components that are shown in Table 6-2.

Table 6-2 *KimonoPremisesTest - testing components*

Component	Purpose
Event Simulator	This component simulates incoming message from an Edge Controller agent (for example tag read).
Event Handler	This component simulates the enterprise back-end.
Commands	This component runs the scenarios (for example one read cycle).

The KimonoPremisesTest bundle provides a configurable login facility.

The setting of either of these components is performed using the setting of the appropriate property in the file, `premises-test.properties`. More generally, any setting for this test bundle is done through this file. The default settings use the three components to simulate one set of read cycles.

The KimonoPremisesTest bundle runs on the Premises Server SMF stack and uses the following file to get its settings:

```
RFID_INSTALL_DIR\IBM\RFID\edgecontroller\premises\smf\
premises-test.properties
```

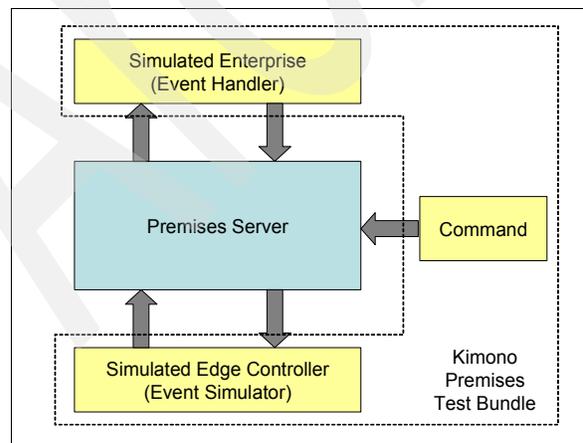


Figure 6-21 *KimonoPremisesTest bundle*

The sections that follow provide an overview of the three component features. Refer to the comments found in the premises-test.properties file to get details on these features and their settings.

Event simulator

The event simulator can send alert, heartbeat, agent start/stop, or tag reading events coming from the Edge Controller agents (Table 6-3).

Table 6-3 Event Simulator Events

Event	Meaning
Alert	Send a configurable alert message from an Edge Controller agent on a configurable topic.
Heartbeat	Send a configurable heartbeat message from an Edge Controller agent on a configurable topic.
StartStop	This simulates a complete read cycle: start, read cycle, stop. The read cycle, start message, stop message, and publish topic are configurable.
Tags	Sends tag read data (one to many) from a configurable reader, a configurable antenna, on a configurable topic.

Event handler

The event handler sends a IBMPremisesUnifiedMessage XML, a configurable event message on given configurable WebSphere MQ queue name of the Premises Server, of a configurable queue manager name (Table 6-4).

Table 6-4 Event handler event messages

Event Message	Meaning
start read	Send a start read event.
stop read	Send a stop read event.
dock door receiving	Send a dock door receiving event.
new tag	Send a new tag read event.
repeat tag	Send a repeat tag event.
heartbeat	Send a heart beat event.
alert	Send an alert event.

Command

The command simulator sends a configurable command using an IBMPremisesUnifiedMessage XML message on a given configurable WebSphere MQ queue name of the Premises Server, of a configurable queue manager name. The default settings allow for triggering of the Event simulator to run read cycles (Table 6-5).

Table 6-5 Command simulator commands

Command	Meaning
start/stop read	Send a start read command, wait for a read cycle amount of time, send a stop read command.

6.2.6 Edge Controller configuration verification

By default, the Premises Server is configured with an Edge Controller whose identifier is E1. Verify that you obtain valid XML when browsing to the following:

`http://hostname:9080/event_admin_web/premises.s1?action=getconfig&edge=E1`

where *hostname* is the Premises Server DNS name. Your XML should be similar to that shown in Figure 6-22 on page 141.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <edge>
- <propertyset id="MicroBrokerConfigurationAgent">
  <property key="server.ip" value="9.42.171.56" />
  <property key="portal.ids" value="P1" />
  <property key="server.port" value="1883" />
  <property key="edge.id" value="E1" />
</propertyset>
- <propertyset id="HeartbeatAgent">
  <property key="heartbeat.period.ms" value="60000" />
  <property key="edge.id" value="E1" />
</propertyset>
- <propertyset id="AlertAgent">
  <property key="threshold" value="error" />
  <property key="edge.id" value="E1" />
</propertyset>
- <propertyset id="RestartAgent">
  <property key="edge.id" value="E1" />
</propertyset>
- <propertyset id="SamsysReaderAgent">
  <property key="red" value="0" />
  <property key="io.id" value="P1_IO_ID" />
  <property key="heartbeat.period.ms" value="10000" />
  <property key="inputpins" value="switch,motion" />
  <property key="transport.connection" value="com.ibm.esc.tcpip.connection.TcpipConnection" />
  <property key="green" value="2" />
  <property key="amber" value="1" />
  <property key="motion" value="0" />
  <property key="selftestmode" value="OFF" />
  <property key="portal.id" value="P1" />
  <property key="reader.id" value="R1" />
  <property key="switch" value="1" />
  <property key="transport.host" value="127.0.0.1" />
  <property key="beep" value="3" />
  <property key="transport.remoteport" value="2101" />
</propertyset>
- <propertyset id="LightTreeAgent">
  <property key="duration.ms.beep" value="500" />
  <property key="io.id" value="P1_IO_ID" />
  <property key="selftestmode" value="OFF" />
  <property key="ignore.green.while.red" value="false" />
  <property key="portal.id" value="P1" />
  <property key="duration.ms.green" value="2000" />
  <property key="duration.ms.red" value="2000" />
</propertyset>
- <propertyset id="DutyCycleAgent">
  <property key="check.interval.ms" value="1000" />
  <property key="check.periodically" value="false" />
  <property key="selftestmode" value="OFF" />
  <property key="portal.id" value="P1" />
  <property key="sampling.period.ms" value="6000" />
  <property key="limit.percent" value="10" />
</propertyset>
- <propertyset id="MotionSensorAgent">
  <property key="delayafterquiet" value="2000" />

```

Figure 6-22 Edge Controller configuration on the Premises Server
 Screen shot reprinted by permission from Microsoft Corporation

6.3 Defining network topology

You need to define the following topology using the Premises Server Administrative Console:

1. Define one root location
2. Within this root location, define one location per reader. The mandatory parameters when defining a location are:
 - location identifier
 - location alias
3. For each defined location (except for root location), define one RFID reader. The mandatory parameters when defining a reader are:
 - reader identifier
 - reader type (such as Alien, Intermec, Matrics, Samsys)
 - reader location identifier
 - reader IP address
 - reader IP port number (according to reader specifications)

Restriction: Even though the Premises Server Administrative Console allows for the definition of several readers within the same location, the only supported RFID topology is when only one reader is defined within one location.

4. Define the Edge Controllers. The mandatory parameters when defining a controller are:
 - controller identifier

Restrictions:

- ▶ Even though the Premises Server Administrative Console allows for the definition of any number of readers within the same controller, the only supported RFID topology is when no more than three readers of the same type are defined within one controller.
- ▶ Even though the Premises Server Administrative Console allows for the definition of any number of controllers within the Premises Server, the only supported RFID topology is when no more than 30 controllers are defined within one Premises Server.

Refer to 5.2, “Defining your RFID network topology” on page 105 to get a definition of the RFID network topology.

Refer to 7.3, “Defining your RFID network topology” on page 150 to get the Premises Server Administrative Console detailed screens to define this topology.

6.4 Installing Edge Controller software

The Edge Controller software is provided as a set of bundles (JAR files) tailored to your specific RFID solution and can be deployed on any supported Edge Controller that has been properly configured to work on your network.

Restriction: The Edge Controller software is *not* included with the IBM WebSphere RFID solution. Contact your Systems Integrator or Edge Controller OEM for more information about obtaining this software.

WebSphere Everyplace Device Manager V5.0 is required to install the Edge Controller software as well as manage the Edge Controller devices. Refer to Chapter 11, “Edge Controller Software installation and management” on page 219 for details.

Restriction: WebSphere Everyplace Device Manager V5.0 is *not* included currently with the IBM WebSphere RFID solution. Contact your Systems Integrator for more information about obtaining this software.

6.5 Uninstalling the Premises Server software

Two scripts are available to uninstall the Premises Server software:

- ▶ A script called `RFID_INSTALL_DIR\IBM\RFID\premises\uninstall.bat` removes both the WebSphere Application Server and the WebSphere MQ Premises Server code, without removing the Premises Server configuration files.
- ▶ A script called `RFID_INSTALL_DIR\IBM\RFID\premises\smf\smf_service` removes the Services Management Framework as a service.

The following steps are needed to uninstall the Premises Server:

1. Ensure WebSphere Application Server and WebSphere MQ are both running.
2. Ensure IBM SMF Service is not running.
3. From a command prompt, execute the following command:
`RFID_INSTALL_DIR\IBM\RFID\premises\uninstall.bat`
4. From a command prompt, execute the following command:
`RFID_INSTALL_DIR\IBM\RFID\premises\smf\smf_service -uninstall`
Your output should be similar to that shown in Example 6-7.

Example 6-7 Premises Server uninstallation log

```
C:\IBM\RFID\premises>Uninstall.bat
Uninstalling Kimono Configuration data and Application
Uninstalling Event Server....Be patient...this may take a while
Uninstalling WAS resources
Restarting WAS to uninstall MQ resources
Uninstalling MQ resources
Restarting WAS to complete uninstallation
WAS restarted
Uninstall complete
```

6.6 Defining administrative roles and security

The Premises Server Administrative Console does not come with built-in user roles and security. Should you need those features, you would have to enable them using the WebSphere Application Server Administrative Console.

The overview of such a procedure is given below.

Note: This procedure is very complex and might lead to a malfunction if not done correctly. It is therefore strongly advised that you refer to *IBM WebSphere V5.0 Security WebSphere Handbook Series*, SG24-6573 for complete details about how to perform this procedure.

The following steps would have to be performed on the WebSphere Application Server Administrative Console:

1. Click **System Administration** → **Console Groups**, add the groups and the associated roles that you would need.
2. Click **Security** → **JAAS Configuration** → **J2C Authentication Data**, add a new alias which allows access to the MQ administration functions.
3. Click **Resources** → **WebSphere MQ JMS Provider**, for each of the existing connection factories, add previously created alias as component and container authentication alias, and specify Mapping Configuration Alias as `DefaultPrincipalMapping`.
4. Click **Security** → **Global Security**, enable security according to your specification (for example check **Enabled** and leave Local OS user registry default).
5. Restart WebSphere Application Server.

6. When security is set on WebSphere Application Server, the message driven beans will no longer be accessible by their calling EJBs.

Therefore, for each RFID enterprise application, you need to activate the *RunAs* feature as follows:

- a. Click **Applications** → **Enterprise Application** → *RFID enterprise application*.
- b. Go to Additional Properties and click **Map Security Roles to users/groups**.
- c. Add the *RunAs* administrative role defined in item 1 on page 144.

Repeat the process for all the RFID enterprise applications that contain message driven beans.

Archived

Archived

Administering the WebSphere RFID solution

This chapter offers a detailed discussion of the WebSphere RFID Premises Server Administrative Console and describes the administration tasks that you perform to define and to manage the components in your RFID solution. It shows you how to use this Web application to accomplish these tasks and explains why and when you need to do them. It covers how to:

- ▶ Open the Administrative Console
- ▶ Define and maintain your network topology
- ▶ Quickly view properties and tag data in the Premises Server database
- ▶ Configure extensions to your RFID solution

7.1 Before you begin

Before you begin administering your RFID solution, be sure you have installed the IBM WebSphere RFID solution properly, including the Edge Controller and the Premises Server. Refer to Chapter 6, “Installing the WebSphere RFID solution” on page 111 for complete instructions.

7.2 Administrative Console overview

This section gives a general overview of the Administrative Console and explains the tasks you can do. In subsequent chapters, we explain how to use the Administrative Console to configure the Premises Server for the Dock Door Receiving scenario.

7.2.1 Getting started with the Administrative Console

The Administrative Console is accessible from a Web browser.

Tip: For best results, use Internet Explorer 6.0 or higher.

To open the Administrative Console, with a Web browser, go to:

```
http://premises_server_ip:9080/event_admin_web
```

The *premises_server_ip* variable is the IP address of your Premises Server. The Home page and welcome message displays as shown in Figure 7-1 on page 149.

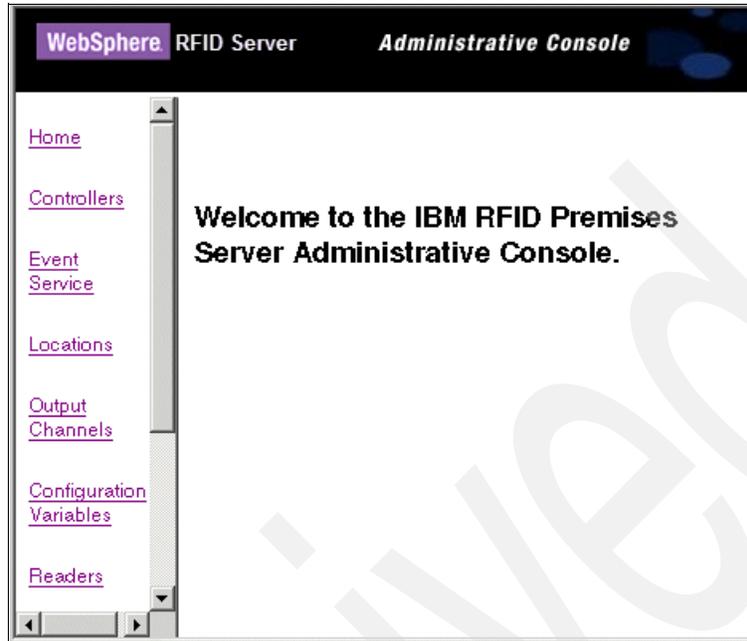


Figure 7-1 IBM WebSphere RFID Premises Server Administrative Console

The navigation frame on the left-hand side of the Administrative Console contains links for all the things you can configure, manage, or view from this application:

- ▶ Home
This displays a welcome message.
- ▶ Controllers
Edge Controllers are devices that control the I/O devices such as RFID readers and motion sensors. They also process tag data and communicate with the Premises Server. You can create, update, and delete controller definitions, and use them to restart the Edge Controller devices.
- ▶ Event Service
Event templates define events and how the information about events is transmitted across the appropriate communication channels and coordinated between the Edge Controller, Premises Server, and enterprise system. You can create, update, or delete event templates.
- ▶ Locations
A location represents a critical point of RFID data collection. In the IBM RFID solution, it is the physical location of an RFID reader (including its antenna,

light tree, and motion device). You can create, update, and delete location definitions.

- ▶ **Output Channels**

Output channels are paths used to send messages from the Premises Server to either the Edge Controller or the enterprise. You can create, update, and delete output channels.

- ▶ **Configuration Variables**

This displays the configuration variables and the values that are in the `premises.properties` file. You can only view this configuration data.

- ▶ **Readers**

RFID readers use radio frequency antennas to scan for RFID tags and read information encoded in the tags. The readers send the tag information through the WebSphere Connection Server Micro Edition (MicroBroker) to the Edge Controller. You can create, update, and delete reader definitions.

- ▶ **Tags**

This displays the RFID tags that have been read and saved in the Premises Server database. You can only view this tag data.

- ▶ **Tasks**

A task is a piece of software or event handler for coordinating the communication of an event between an Edge Controller, Premises Server, and enterprise system. You can create, update or delete tasks.

- ▶ **Agent Configuration**

Agents control the behavior of I/O devices and other system components, communicate with adapters, and perform other related processing tasks. You can change the property settings of the agents.

- ▶ **About**

This displays the version of the Premises Server that you are running.

7.3 Defining your RFID network topology

This section explains the tasks that you do to define your network topology and the order in which you do these tasks.

The first task that you do when you set up your RFID solution is define a network topology. An RFID network topology consists of a Premises Server and other physical components such as: Edge Controllers, RFID readers, light trees, switches, and motion sensors. In addition, there are logical components, such as locations, that provide associations between the physical devices, and software

components, such as agents, that enable them to communicate with the Premises Server, other servers in the solution framework, and the enterprise system.

Figure 7-2 shows a sample RFID network topology.

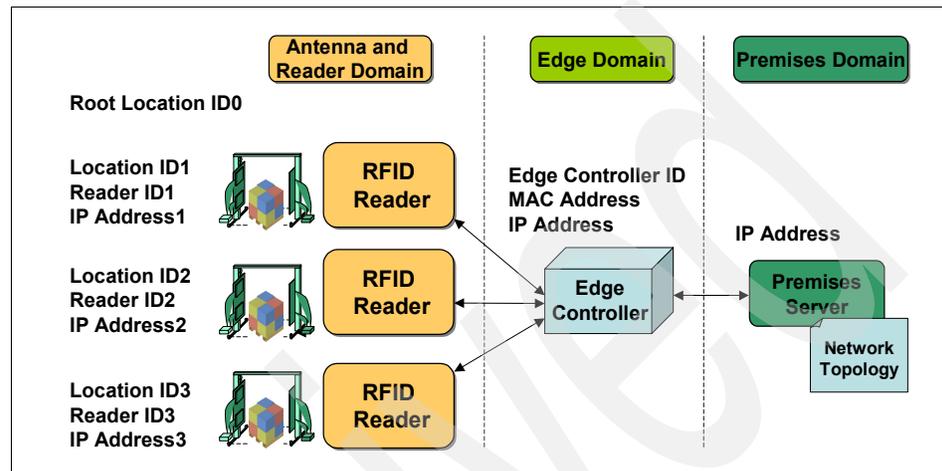


Figure 7-2 Sample RFID network topology

You can define and configure the network topology from the Administrative Console and store the information in a configuration database on the Premises Server. You can also restart Edge Controllers from the Administrative Console to immediately implement any modifications you make to the network topology configuration.

The sections that follow explain how to use the Administrative Console to define, configure, and then manage these components of your network topology:

- ▶ Locations
- ▶ RFID readers
- ▶ Reader Agents

7.3.1 Locations

Locations are the first thing that you define in the network topology. A location definition can include a location contact, which you can also define on the Locations page.

What is a location

A location represents the physical location of an RFID device and is part of the configuration definition for readers. Each reader is associated with one location.

A location is also part of a controller definition. It associates an Edge Controller with the reader at that location.

Locations are nested. There is one default Root Location that contains all the other locations. The root location can never be deleted or assigned to a device. All locations that you create are beneath this root.

To work with locations, click **Locations** in the Administrative Console navigation frame. The Locations page displays as shown in Figure 7-3.

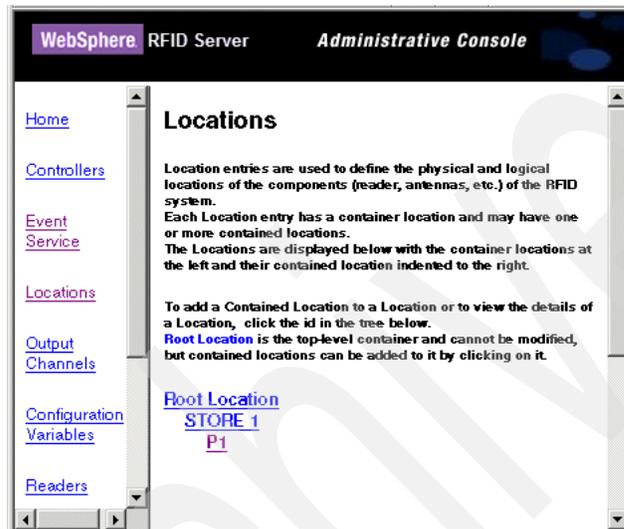


Figure 7-3 Locations

By default, the Premises Server has two locations defined: Root Location contains STORE 1, which in turn contains P1. These are provided as examples for testing purposes. You cannot change their names, but you can modify them for your own installation or delete them and create your own locations under the Root Location.

Location Contacts

Locations Contacts are the responsible people to contact for problems or questions about the devices at these locations. Figure 7-4 on page 153 shows an example of the Location Contacts window.

Note: Location contacts are not required, but if you want to maintain this information in the Premises Server configuration database, you need to define location contacts before you can associate them with a location.

Location Contacts

To view the contact details, click on the Name of the Contact. The details screen will allow you to update or delete the contact.
To create a new contact, press Create.

Create

Name	Phone	Mobile	Pager	Email
No Contacts Defined				

Figure 7-4 Locations contacts

By default, the Premises Server does not have any defined location contacts. You can add your location contacts at any time. If you define your contacts after you define your locations, you can go back and modify the location definitions to associate the contacts with a location.

Create new location contacts

To create new location contacts, follow these steps:

1. Click **Create**. The Create new Contact page displays.
2. Enter the information for this person and click **Create**. The new contact displays on the Locations page.

Display, update, and delete contacts

You can display a contact definition to review, update, or delete it. To display a location contact, click the contact name listed on the Locations page or anywhere it appears as a hot link in the Administrative Console pages.

Create new locations

To create a new location:

1. On the Locations page, click **Root Location** (or the location that will contain it). The Edit Locations details page displays (Figure 7-5).

WebSphere RFID Server Administrative Console

Home
Controllers
Event Service
Locations
Output Channels
Configuration Variables
Readers

Edit Location details

Root Location is the top-level container and cannot be modified, but children can be added to it by pressing Create Child.

Location Id	Root Location
Location Alias	
Description	The root location

Create Contained Location Cancel

Figure 7-5 Edit Location details

2. Click **Create Contained Location**. The Create new Locations page displays (Figure 7-6).

Create new Location

To create a new location, enter a non-empty unique Location Id, modify the other fields (optional), and press Create.
Root Location cannot be modified.

Location Id	
Location Alias	
Description	
Is Addressable	false
Is InSelfTestMode	false
Contact	
Container Location	Root Location

Figure 7-6 Create new Location

3. Enter a unique Location ID and Location Alias. The Premises Server uses the Location ID. The Location Alias is used for communication with the enterprise.

Note: These two fields are required. They can have the same value but they must be unique within your network topology.

4. Optionally, enter or modify the other data:
 - If you want to add contact information, change the value of *Is Addressable* to **true** and then select a contact from the menu list.
(You cannot add a contact unless *Is Addressable* is set to true.)
 - If you want to remotely test the RFID hardware from the Premises Server Administrative Console, change the value of *Is InSelfTestMode* to **true**.
 - Scroll down to enter the geographic Address fields: Street, City, State, Zip.
5. When you are done, click **Create**. The new location is added to the hierarchy of locations. It will be available for selection when creating definitions for readers and controllers.

Display, update, and delete locations

You can display a location definition to review, update, or delete it.

Display location details

To display a location definition, click **Location Id** in the list on the Locations page or anywhere it appears as a hot link in the Administrative Console pages.

Figure 7-7 is an example of what displays.

Location Id	P1
Location Alias	P1_Alias
Description	An imaginary portal for testing purposes.
Is Addressable	false
Contact	
Container Location	STORE 1
Controller	E1
Readers	R1
Antennas	
Address	
Street	
Street	
City	
State	
Zip	
<input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Create Contained Location"/> <input type="button" value="Cancel"/>	

Figure 7-7 Display location details

Update location data

You can update these fields in a location definition:

- ▶ Location Alias
- ▶ Description
- ▶ Is Addressable
- ▶ Is InSelfTestMode
- ▶ Address: Street, City, State, Zip

Delete location definition

You can delete a location definition only if it is not associated with a reader.

Tip: After a location is associated with a reader, you must delete the location association in the reader definition before you can delete the location definition.

7.3.2 Readers

The second thing to define in your network topology is your RFID readers. To define a reader, you give it an identifier, associate it with a location, and specify its manufacturer type, IP address, and port number. You should have this information at hand before you start.

To work with reader definitions, click **Readers** in the Administrative Console navigation frame. Figure 7-8 shows the Readers page.

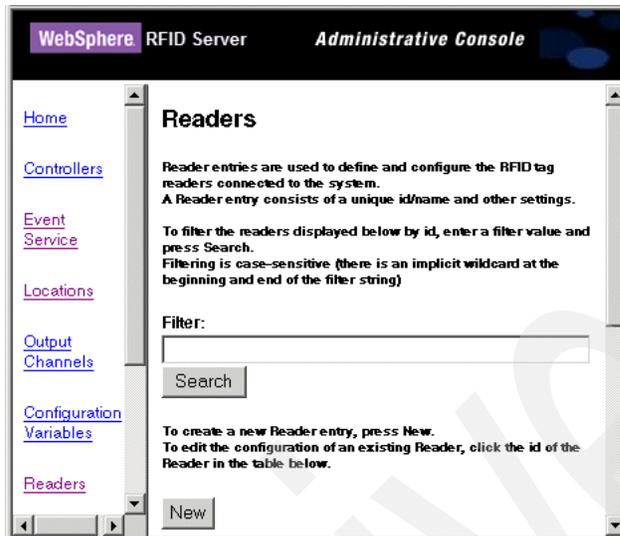


Figure 7-8 Readers page

In the middle of the page are the defined readers. By default, the Premises Server comes with one reader definition, R1, for a Samsys reader at location P1, as shown in Figure 7-9.

Reader id	Location
R1	P1

Figure 7-9 An example of a reader definition

You cannot change the name of this R1 reader definition, but you can modify it for your own installation, or delete it and create your own.

At the bottom of the Readers page is a listing of available Reader Types for creating reader definitions as shown in Figure 7-10.

Reader Types	
The following list contains the Reader Types that are currently defined.	
Reader Type	Description
AlienType	
IntermecType	
MatricsType	
SamsysType	
TagSysType	
SymbolType	
FeigType	

Figure 7-10 Reader types

Important: Alien, Intermec, Matrics, and Samsys reader types are supported. The other reader types are available *as-is* and are not supported.

Create new readers

To create new readers, follow these steps:

1. To create a new reader, click **New** from the Readers page. The Create a new Reader page displays (Figure 7-11).

Create a new Reader	
A Reader consists of a non-empty unique ID, and other characteristics.	
To create a new reader, enter an ID, set the other values and press 'Create'.	
Reader Id	<input type="text"/>
Reader Type	<input type="text" value=""/>
Location Id	<input type="text" value=""/>
Reader IP address	<input type="text"/>
Reader IP Port Number	<input type="text"/>
<input type="button" value="Create"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

Figure 7-11 Create a new Reader page

2. Enter or select the following information:
 - Reader ID
A unique reader identifier
 - Reader Type
One of the available reader types (by manufacturer)

– Location ID

An available location in your topology that has not been assigned to another device.

Note: Although the Administrative Console allows you to choose a location that has already been assigned, the device infrastructure only allows one reader to one location.

– Reader IP Address

The IP address on your network.

– Reader IP Port Number

The correct port number for the type of device, for example:

- Alien = 23
- Intermec = 2189
- Samsys = 2101
- Matrics = 3000

Note: Be sure to verify the port number with your device manufacturer's documentation.

3. Click **Create**. The reader displays in the list on the Readers page. It will be available for selection when creating controller definitions.

Note: You can create and save a reader definition that only has a reader ID in it, but you will have to return and add the other information before you can test or use your RFID network.

Search, display, update, and delete readers

You can display a reader definition to review, update, or delete it.

Search for readers

If you have many readers and the list on the Readers page is long, you can search for specific readers and display just those in the list. The search filter is case-sensitive and assumes a wild card at the beginning and end of the search string.

For example, by entering just the letter *R*, you can display all the readers that are defined in the default Premises Server configuration database that have the capital letter *R* anywhere in the name as shown in Figure 7-12.

To filter the readers displayed below by id, enter a filter value and press Search. Filtering is case-sensitive (there is an implicit wildcard at the beginning and end of the filter string)

Filter:

To create a new Reader entry, press New.
To edit the configuration of an existing Reader, click the id of the Reader in the table below.

Reader Id	Location
R1	P1

Figure 7-12 Reader search

Display reader details

To display a reader definition, click the Reader ID listed on the Readers page, or anywhere it appears as a hot link in the Administrative Console pages. The Edit Reader details page displays as shown in Figure 7-13.

Edit Reader details
To modify the attributes of this reader, enter the new values and press 'Update'. The reader id and other fields that are greyed-out cannot be modified.
If the reader is associated to a controller, the controller id is not editable and it is determined by the controller.
To delete this reader, press Delete.

Reader Id	Alien1A
Controller	ArcomUno
Reader Type	AlienType
Location	A305-21-1A
Reader IP address	9.42.171.97
Reader IP Port Number	23

Figure 7-13 Edit Reader details page

Update reader data

You can update these fields in a reader definition:

- ▶ Reader type
- ▶ Location
- ▶ IP address
- ▶ Port number

Note: You cannot edit the Controller value because the association between a reader and its Edge Controller is maintained in the controller definition. You must change it there.

Delete reader definitions

You can delete a reader definition at any time; you do not have to remove the locations associations first.

7.3.3 Agents

The Premises Server comes with agents for all the supported devices, for the devices that are available *as is*, and for other system components and functions. They include reader agents, I/O agents, controller agents, and filter agents.

Agents perform several functions. They connect I/O device adapters to the WebSphere Everyplace Connection Server, act as controllers for the I/O environment, and perform processing appropriate for the device or connection. For example, the SamsysReaderAgent connects the Samsys reader adapter to the Connection Server and filters tag information. In a similar way, the LightTreeAgent controls the number of milliseconds between beeps.

Each agent has a configurable set of properties. Whenever you install a reader, you should review the default settings of the properties for its associated device agents. They might be configured to suit your needs. If not, you can modify their settings. Be sure to refer to Appendix C, “Agents, properties, and values” on page 253 for a complete list of agents, their properties, and values.

To work with agent properties, click **Agent configuration** in the Administrative Console navigation frame. The Edit Agent Properties page displays as shown in Figure 7-14.

Edit Agent Properties	
To update an Reader Agent property, use the pulldowns to select the desired combination of Reader Agent, Locations and Properties, press 'Update'.	
Reader Agent	AlienReaderAgent
Reader Location	*
Agent Property	green
Property Value	2
Update	Cancel

Figure 7-14 Edit Agent Properties page

View and edit agent properties

You can access all the agent properties from the Administrative Console, but there are many that you would never want to change. And others, such as changing the pin to which particular light is mapped, require that you also modify the hardware. Refer to Appendix C, “Agents, properties, and values” on page 253 for details on all agents, their properties, valid values, and defaults.

To view or change agent properties:

1. Select the agent for your output device from the Reader Agent menu list.

Note: The Reader Agent menu list contains all agents, not just reader agents.

2. Select the property from the Agent Property menu list that you want to verify or change. The properties in the list will vary according to the agent you selected.
3. To change the value, enter a new Property Value and click **Update**.

Note: Reader Location is not used. For device agents, the property settings apply to all occurrences of the device type, regardless of location. For other agents, location does not apply.

7.3.4 Controllers

After you define the locations and the readers, you can define the Edge Controllers and associate them with locations and the devices at those locations. When you define an Edge Controller, you select the locations for this Edge Controller and the readers at that location. Edge Controllers can have multiple locations in their definition because they can control more than one reader of the same type.

To work with controller definitions, click **Controllers** in the Administrative Console navigation frame. The Controllers page displays (Figure 7-15).

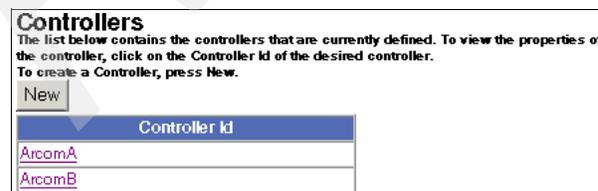


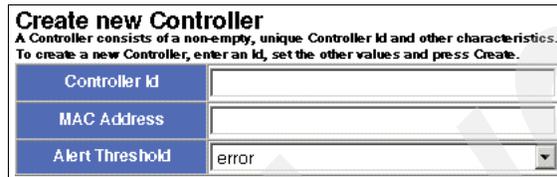
Figure 7-15 Controllers

By default, the Premises Server comes with one controller definition, E1. You cannot change the name of this E1 controller definition, but you can modify it for your own installation, or delete it and create your own.

Create new controllers

To create a controller, you give it an identifier and specify the locations and devices with which it will communicate. You can also specify the MAC address and change the alert threshold. To create new controllers, follow these steps:

1. To create a new controller, click **New**. The Create new Controller page displays as shown in Figure 7-16.



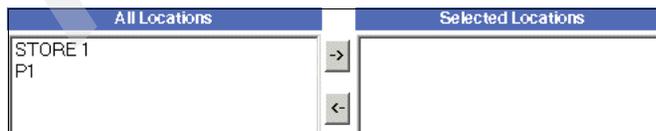
Create new Controller	
A Controller consists of a non-empty, unique Controller Id and other characteristics. To create a new Controller, enter an Id, set the other values and press Create.	
Controller Id	<input type="text"/>
MAC Address	<input type="text"/>
Alert Threshold	error

Figure 7-16 Create new Controller

2. Enter a unique identifier for the Controller ID and, optionally, the controller's MAC address. The MAC address field is for your own information. It is not used by the system at this time.
3. Select an alert threshold to determine the level of information to be included in the Edge Controller log file on the Premises Server: error, warning, debug, or info.

Important: An error Alert Threshold is the least verbose setting and info is the highest. The info Alert Threshold setting sends information about more events to the log file, which generates a large amount of traffic and will significantly impact network performance.

4. In the list of All Locations (Figure 7-17), click to select the locations of the readers for this Edge Controller, then click the right-arrow, **->**, to add them to the list of Selected Locations. For multiple locations, you must add them one at a time. Use the left-arrow, **<-**, to remove a location from the list of Selected Locations.



All Locations		Selected Locations
STORE 1	->	
P1	<-	

Figure 7-17 Controller locations

5. In the list of All Readers (Figure 7-18), select the readers at the associated locations, then click the right arrow, **->**, to add them to the list of Selected Readers. For multiple readers, you must add them one at a time. Use the left arrow, **<-**, to remove a reader from the list of Selected Readers.

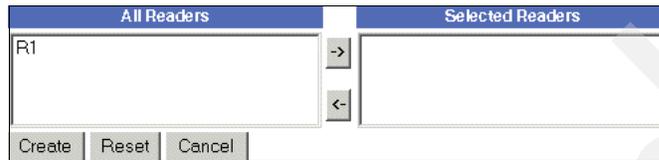


Figure 7-18 Controller readers

6. When you are done, click **Create**. The new controller is listed on the Controllers page.

Display, update, and delete controllers

You can display a controller definition to review, update, or delete it.

Display controller details

To display a controller definition, click the Controller ID listed on the Controllers page, or anywhere it appears as a hot link in the Administrative Console pages.

Update controller data

You can update any fields in a controller definition except the Controller ID.

Delete controller definitions

You can delete a controller only if it does not have any associated locations or readers. To delete a controller:

1. Remove the locations from the list of Selected Locations.
2. Remove the readers from the list of Selected Readers.
3. Click **Update**.
4. Return to the controller definition and click **Delete**.

Restart an Edge Controller

Whenever you make a change to a network topology, you must restart the Edge Controller to reload the new values that were saved in the Premises Server configuration database. You can do that right from the Administrative Console. For example, if you change a reader's IP address, modify an agent property, or change the alert threshold for an Edge Controller, you can implement them quickly without restarting the system.

To reload an Edge Controller's configuration:

1. Display the controller definition (Figure 7-19).

Edit Controller Details	
To update a controller, modify the fields and press Update. The controller id cannot be modified. To delete a controller, press Delete.	
Controller Id	E1 <input type="button" value="Reload Config"/>
MAC Address	<input type="text"/>
Alert Threshold	error

Figure 7-19 Reload Config

2. Click **Reload Config**. A verification message displays as shown in Figure 7-20.



Figure 7-20 Restart verification

3. Click **OK**.

Note: This series of steps restarts SMF that is running on the Edge Controller. When SMF starts, it loads the configuration from the Premises Server configuration database. The Edge Controller is not operational during this time, which is typically less than one minute.

View network topology configuration data

After an Edge Controller is started, you can use a Web browser to view the network topology configuration information that is stored in the Premises Server configuration database.

To view the default configuration data, with a Web browser, go to:

```
http://localhost:9080/event_admin_web/premises.s1?action=
getConfig&edge=E1
```

To view the configuration data for a controller that you defined, replace E1 with your Controller ID.

7.4 Viewing configuration and tag data

This section explains the configuration variables and tag data that you can view with the Administrative Console.

7.4.1 Configuration variables

You can display the parameters and their values that are in your Premises Server `premises.properties` file. These parameters control many functions of the Premises Server.

To view these parameters, click **Configuration Variables** in the Administrative Console navigation frame. The Configuration variables page displays as shown in Figure 7-21.

Configuration variables	
The following variables are currently defined to configure the components of the Premises Server.	
Name	Value
<code>com.ibm.rfid.premises.sysman.logging.alert.formatter.delimiter.open</code>	[
<code>com.ibm.rfid.premises.sysman.logging.heartbeat.formatter.message.down</code>	DOWN
<code>com.ibm.rfid.premises.sysman.logging.heartbeat.handler.pattern</code>	C:\download\RFID\CD11\AB\heartbeats.log
<code>com.ibm.rfid.premises.sysman.logging.heartbeat.formatter.delimiter.field</code>	
<code>autoid.core.savantid</code>	premises_cit
<code>com.ibm.rfid.premises.sysman.logging.heartbeat.handler.limit</code>	500000
<code>com.ibm.rfid.premises.sysman.logging.heartbeat.formatter.timestamp</code>	MM d HH:mm:ss yyyy
<code>com.ibm.rfid.premises.tag.persistence</code>	false
<code>com.ibm.rfid.premises.sysman.logging.alert.handler.limit</code>	2000000
<code>com.ibm.rfid.location.external.format</code>	alias

Figure 7-21 Configuration variables

You cannot change these properties on this page. To change any of these properties, you must edit the `premises.properties` file, save it, then stop and restart the Premises Server. This file is located in the following directory on the Premises Server:

`C:\IBM_RFID_HOME\RFID\premises\eventserver\properties`

This file contains detailed comments that provide descriptions of all the properties and the effect they have on the Premises Server performance.

7.4.2 Tags

You can view the RFID tags that have been read and stored in the Premises Server database.

Important: Tag data is only stored if the `com.ibm.rfid.premises.tag.persistence` property in your `premises.properties` file is set to `true`. Use this setting with caution. It can result in an enormous amount of data being stored on the Premises Server and can affect its performance.

To view tag data, click **Tags** in the Administrative Console navigation frame. The Tags page displays (Figure 7-22).

Tags

A Tag consists of a unique ID and a history.
To filter the tags displayed below by ID, enter a filter value and press Search.
Filtering is case-sensitive (there is an implicit wildcard at the beginning and end of the filter string)

Filter:

To view the details of a Tag, click on its Tag Id in the table below.
To view the tag's history, click on its Tag History link.

Tag Id	Tag History	Date/Time Stamp
30544b5a1c61d50000000000	premises_cit112981891265622	Thu Oct 20 03:35:16 EDT 2005
30144b5a1cc3780000000001	premises_cit112981890940620	Thu Oct 20 03:35:13 EDT 2005
907ce30e1c000000	premises_cit112981890167218	Thu Oct 20 03:35:05 EDT 2005
807ce61bac000001	premises_cit112981888229716	Thu Oct 20 03:34:44 EDT 2005
31144e4e45ca3c004000000	premises_cit112934498273412	Fri Oct 14 22:56:21 EDT 2005
31144e4e45ca3c002000000	premises_cit112933600067212	Fri Oct 14 20:26:40 EDT 2005

Figure 7-22 Tags

Search tag data

The tag list is likely to be long. You can search for specific tags by their Tag ID and display just those tags in the list. The search filter is case-sensitive and assumes a wild card at the beginning and end of the search string.

Here we found and displayed just the tags with e4e4 in the Tag ID (Figure 7-23).

Tags

A Tag consists of a unique ID and a history.
 To filter the tags displayed below by ID, enter a filter value and press Search.
 Filtering is case-sensitive (there is an implicit wildcard at the beginning and end of the filter string)

Filter:

To view the details of a Tag, click on its Tag Id in the table below.
 To view the tag's history, click on its Tag History link.

Tag Id	Tag History	Date/Time Stamp
31144e4e45ca3c004000000	premises_cit112934498273412	Fri Oct 14 22:56:21 EDT 2005
31144e4e45ca3c002000000	premises_cit112933600067212	Fri Oct 14 20:26:40 EDT 2005

Figure 7-23 Tag search

Display tag details

If you click the link in the Tag ID field, the View Tag Details page displays as shown in Figure 7-24. The fields on this page are not used at this time.

View Tag Details

The following table contains detailed information about the tag you have selected.
 To view detailed information about the tag's parent or children, click on the name of the parent or child in the correct section below.

Tag Id	30544b5a1c61d50000000000
Tag History	premises_cit112981891265622
Object Link Id	
Object Link Description	
Contained In	
Contained Tags	

Figure 7-24 View Tag Details page

If you click the link in the Tag History field, the View Tag History page displays (Figure 7-25). This page shows the location at which a tag was read, which reader and which antenna read it, and the number of times it was read. This information can be useful for troubleshooting antenna placement or problems.

View Tag History

Tag History for Tag Id: 30544b5a1c61d50000000000

UID	EventID	Location	Reader	Contained In	Antenna	Count	DTS
premises_cit112981891265622	Event_112981891257821	DEMO	DEMO		0	1	Thu Oct 20 03:35:16 EDT 2005

Figure 7-25 View Tag History page

If you click the link in the EventID field, the View Tag Metadata page displays as shown in Figure 7-26. The fields on this page are not used at this time.

View Tag Metadata	
Tag Metadata for Event Id: Event_112934498182811	
Tag Metadata	
Name	Value
No Metadata for this Event Id.	
Tags in this Event Id	
Tag Id	
3114f4e4e45ca3c004000000	

Figure 7-26 View Tag Metadata page

7.5 Configuring your RFID solution extensions

This section explains the things that you can do from the Administrative Console to configure your Premises Server and extend its function. You can:

- ▶ Define output channels for Premises Server message communication
- ▶ Define event templates to determine how event information is transmitted across communication channels
- ▶ Define tasks, or event handlers, for coordinating the communication of an event between an Edge Controller, Premises Server, and your enterprise system

Task definitions, event templates, and output channels are all related. In the Premises Server, processing is performed by discrete tasks that are known events. The outputs from these events are transmitted as messages. In the Administrative Console, task definitions associate tasks with event templates. The event templates act as an intermediary to route the event messages to their destination through their associated output channels (Figure 7-27).



Figure 7-27 Relationship between tasks, event templates and output channels

7.5.1 Output channels

Output channels are paths used to send messages from the Premises Server to either the Edge Controllers or the enterprise. Table 7-1 shows the types of output channels.

Table 7-1 Types of Output Channels

Output Channel	Purpose
Email	For e-mail messages
HTTP	For HTTP messages
JMS	For Java Message Service messages
JMS Topic	For Java Message Service topic messages
MQ	For WebSphere MQ messages

The JMS, JMS Topic, and MQ output channels communicate with specific messaging systems. If you do not have these messaging systems, or do not choose to use them for Premises Server communication, you can define e-mail or HTTP output channels to route event messages.

To work with output channels, click **Output Channels** in the Administrative Console navigation frame. The Output Channels page displays as shown in Figure 7-28.

Output Channels

The list below contains the output channels that are currently defined. To view the properties of the output channel, click on the 'Channel Id' of the desired channel.

Channel Id	Description	XSL Transform	Type
control.outchannel	output channel to enterprise		JMS
edge.outchannel	output channel to edge		JMS

Create Output Channel

To create an output channel, select the channel type from the list below and click on the 'New' button. You will be redirected to a page where you can set the properties of the output channel.

Email Output Channel

Figure 7-28 Output channels

By default, the Premises Server comes with two predefined output channels that use a WebSphere MQ Java Messaging System (JMS) for routing, as described in Table 7-2.

Table 7-2 Premises Server Predefined Output Channels

Predefined Output Channel	Purpose
control.out.channel	Allows messages to be communicated between the Premises Server and your enterprise.
edge.out.channel	Allows tag data to be communicated between the Premises Server and Edge Controllers.

These predefined channels are used by the Premises Server internal processing and the Dock Door Receiving scenario. We recommend that you do not change or delete them. You can define additional output channels to use other JMS systems, e-mail, or HTTP for routing this data.

Create new output channel

To create a new output channel:

1. Select the output channel type from the menu list and click **New**. The Create new Output Channel page for that type displays.
2. Enter a unique Channel ID and the other data for that type of output channel.
3. Click **Create**. The new output channel displays in the list on the Output Channels page. It will be available for selection when creating event templates.

Each output channel has data that is common to all channels and data that is specific to its channel type, so there are different input fields on the Create Output Channel pages. For example, Figure 7-29 on page 171 shows the Administrative Console page for creating a new e-mail output channel. The fields on this page are specific to e-mail format.

Create new Email Output Channel
 An Email Output Channel consists of a non-empty, unique channel id and other characteristics.
 To create a new Email Output Channel, enter an id, set the other values and press Create

Channel Id	<input type="text"/>
Description	<input type="text"/>
XSL Transform	<input type="text"/>
JNDI Session	<input type="text"/>
Recipient	<input type="text"/>
From Address	<input type="text"/>
Subject	<input type="text"/>

Create Reset Cancel

Figure 7-29 Create new Email Output Channel

It is helpful to have the information you will need at hand when creating a new output channel definition. Some of the output channel definitions require data that is related to preconfigured parameters in your WebSphere Application Server. Review the input fields for the particular output channel type before creating a new output channel. Refer to Table 7-3 for information about fields and their definitions for the various types of output channels.

Table 7-3 Field definitions for various types of output channels

Field	Definition
Fields common to all types of output channels	
Channel ID	Unique identifier for the output channel
Description	Text to describe the channel's function
XSL Transform	URL of the XSL style sheet, if needed to transform the outgoing message into a format required by the target application
Email output channel fields	
JNDI Session	Java Naming and Directory session for the e-mail output channel
Recipient	E-mail address of the message recipient
From Address	E-mail address of the sender
Subject	Subject of the e-mail
HTTP output channel field	
URL	The URL of the destination server
JMS output channel fields	
Connection Factory	Java Message Service connection factory that has objects used to create connections to JMS destinations
Queue	WebSphere MQ queue that defines a point-to-point destination type
JMS topic output channel fields	
Topic Factory	Java Message Service topic connection factory that has objects used to manage connections between JMS topics

Field	Definition
Topic	Java Message Service topic that has objects used to manage message flow from publishers to subscribers
MQ output channel fields	
Queue	WebSphere MQ queue to define a point-to-point destination type
Queue Manager	WebSphere MQ queue manager to control access to queues and serve as a transaction coordinator for all queue operations
Channel	WebSphere MQ channel to provide a communication path between queue managers
Hostname	Host name of the MQ Manager server
Port	Port number of the MQ Manager server
Uid	User ID of the MQ Manager server
Pwd	Password of the MQ Manager server

Display, update, and delete output channels

You can display an output channel definition to review, update, or delete it.

Note: We recommend that you do not update or delete the predefined output channels. If you do, the Dock Door Receiving scenario might not work as expected.

Display output channel details

To display an output channel definition, click **Channel Id** in the table on the Output Channels page. The Modify Output Channel details page displays as shown in Figure 7-30.

Modify JMS Output Channel Details	
<small>To modify the attributes of this output channel, enter the new values and press Update. The Channel Id and other fields that are greyed-out cannot be modified. To delete the output channel, press Delete.</small>	
Channel Id	control.out.channel
Description	output channel to enterprise
XSL Transform	
Connection Factory	jms/ibm.rfid.qm
Queue	jms/control.out.q
Update	Delete Cancel

Figure 7-30 Modify JMS Output Channel Details

Update output channel data

You can change any output channel data except the Channel ID.

Delete output channels

You can delete output channels at any time.

7.5.2 Event services (templates)

In the RFID network topology, an event is a type of action that takes place in the RFID network, such as a new tag read. Event templates define these actions and also define how the event information is transmitted across the appropriate communication channels and coordinated between the Edge Controller, Premises Server, and enterprise system.

You can associate an existing event with any of your defined output channels. Then, in a task definition, you can associate a task with this event template to route the task event messages to a particular output channel.

If you are introducing new events (such as commands, requests, and responses) you can create a new event template and then, in a task definition, associate it with a predefined task.

To work with event templates, click **Event Service** in the Administrative Console navigation frame. The Event Templates page displays as shown in Figure 7-31.

Event Templates		
The list below contains the event templates that are currently defined. To view the properties of an event template, click on the 'View Template Parameters' link.		
Event Template Name	Description	Actions
heartbeat	Heartbeat messages sent by an Edge device. These messages contain the status of the Edge as well as any connected readers. This is a primary event.	View Template Properties,
alert_error	Error messages sent by an Edge device. This is a primary event.	View Template Properties,
alert_warning	Warning messages sent by an Edge device. This is a primary event.	View Template Properties,
alert_info	Informational messages sent by an Edge device. This is a primary event.	View Template Properties,

Figure 7-31 Event Templates

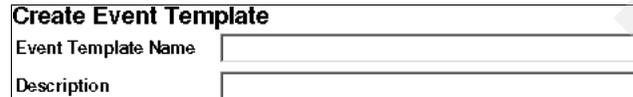
By default, the Premises Server comes with several predefined event templates, which are listed here. Note that some of predefined event templates have associated output channels:

- ▶ alert_info
- ▶ alert_warning
- ▶ alert_debug
- ▶ alert_error
- ▶ heartbeat
- ▶ started_reading
- ▶ stopped_reading
- ▶ started_stopped_reading
- ▶ tag_read new_tag
- ▶ repeat_tag
- ▶ start_reading
- ▶ stop_reading
- ▶ start_stop_reading_internal (associated with edge.out.channel:JMS)
- ▶ external_validation
- ▶ external_validation_internal (associated with edge.out.channel:JMS)
- ▶ reload_configuration_internal (associated with edge.out.channel:JMS)
- ▶ dock_door_receiving (associated with control.out.channel:JMS)
- ▶ dock_door_receiving_response

Create event templates

To create a new event template:

1. Scroll to the bottom of the Event Templates page and click **New**. The Create Event Template page displays as shown in Figure 7-32.



Create Event Template	
Event Template Name	<input type="text"/>
Description	<input type="text"/>

Figure 7-32 Create Event Template

2. Enter a unique Event Template Name and Description.
3. Select a defined channel from the list of Available Channels (see Figure 7-33), and then click the right-arrow, **->**, to add it to the list of Selected Channels. Use the left-arrow, **<-**, to remove a channel from the list of Selected Channels.



Available Channels		Selected
control.out.channel : JMS	->	
edge.out.channel : JMS	<-	
Create Event Template		Cancel

Figure 7-33 Event Template channels

4. Click **Create Event Template**. Your new event template displays at the bottom of the list on the Event Templates page. It will be available for selection when creating tasks.

Display, update, and delete event templates

You can display an event template definition to review, update, or delete it.

Note: We recommend that you do not update or delete the predefined event templates. If you do, the Dock Door Receiving scenario might not work as expected.

Display event template details

To display an event template, click **View Template Properties** in the corresponding table row. The Event Template Detail page displays as shown in Figure 7-34.

Event Template Detail	
Event Template Name	started_stopped_reading_external
Description	A started or stopped reading event whose location ha:
Available Channels	Selected
edge.out.channel : JMS	control.out.channel : JMS
Update Event Template	Delete Event Template
Cancel	

Figure 7-34 Event Template Detail

Update event template data

You can update all event template data except the Event Template Name. You can associate the event template with the predefined output channels or with new ones you created to communicate task event messages between the Premises Server, Edge Controller, and the enterprise.

We recommend that you do not update the predefined event templates.

Delete event templates

You can delete an event template at any time. You do not have to remove the output channel associations first.

We recommend that you do not delete the predefined event templates.

7.5.3 Tasks (event handlers)

A task is a piece of software, or event handler, for coordinating the communication of an event between an Edge Controller, Premises Server, and enterprise system. For example, the Dock Door Receiving Event Handler directs the delivery of tag information received from the Edge Controller to the Premises Server and out to the enterprise.

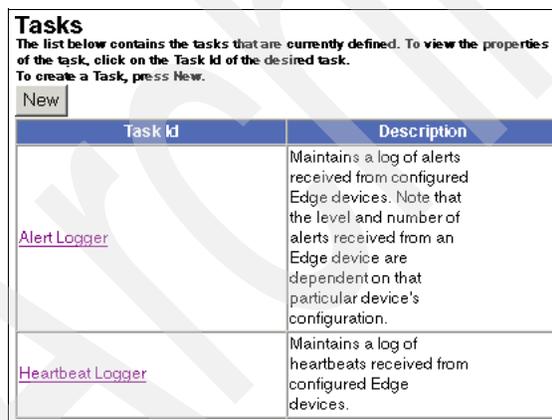
By default, the Premises Server includes many predefined tasks:

- ▶ Alert Logger
- ▶ Heartbeat Logger
- ▶ Tag Read Event Handler
- ▶ Started Stopped Reading Event Handler
- ▶ External Validation Response Handler
- ▶ Start Stop Reading Command Handler
- ▶ Dock Door Receiving Event Handler
- ▶ Dock Door Receiving Response Handler

These tasks are used in the Premises Server internal processing and the Dock Door Receiving scenario. We recommend that you do not change or delete them.

If you are extending your Event Server architecture, you might have additional event handlers. This type of custom code is written typically by people who are helping you to customize your RFID solution, delivered as message-driven beans, and deployed on WebSphere Application Server. In these cases, you need to create new task definitions for any of these custom tasks or event handlers.

To work with tasks, click **Tasks** in the Administrative Console navigation frame. The Tasks page displays as shown in Figure 7-35.



Task Id	Description
Alert Logger	Maintains a log of alerts received from configured Edge devices. Note that the level and number of alerts received from an Edge device are dependent on that particular device's configuration.
Heartbeat Logger	Maintains a log of heartbeats received from configured Edge devices.

Figure 7-35 Tasks

The Tasks page lists all the defined tasks and their descriptions.

Create new tasks

To create a new task:

1. Click **New**. The Create Tasks page displays as shown in Figure 7-36.

Create new Task
A Task consists of a non-empty, unique Task Id and other characteristics.
To create a new Task, enter an Id, set the other values and press Create.

Task Id	Description

Available Events	Selected Events
heartbeat alert_error alert_warning alert_info alert_debug started_reading stopped_reading started_stopped_reading_external tag_read new_tag	

Create Reset Cancel

Figure 7-36 Create new Task

2. Enter a unique Task ID and Description.

Note: The task identifier must match the naming in the message-driven bean deployment descriptor for this event handler. Consult with the provider of this custom code.

Select the events that this task handles from the list of Available Events, and then click the right-arrow, **->**, to add them to the list of Selected Events. For multiple events, you must add them one at a time. Use the left arrow, **<-**, to remove an event from the list of Selected Events.

3. When done, click **Create**.

Display, update, and delete tasks

You can display a task definition to review, update, or delete it.

Note: We recommend that you do not update or delete the predefined tasks. If you do, the product scenarios (Dock Door and Print, Verify, Ship) might not work as expected.

Display task details

To display a task, click its Task ID in the table on the Tasks page. The Edit Task Details page displays as shown in Figure 7-37.

Task Id	Alert Logger
Description	Maintains a log of alerts received from configured Edge devices
Available Events	heartbeat started_reading stopped_reading started_stopped_reading_external tag_read new_tag repeat_tag start_reading stop_reading start_stop_reading_internal
Selected Events	alert_info alert_warning alert_error alert_debug

Update Delete Cancel

Figure 7-37 Edit Task Details

Update task data

You can update any task data except the Task ID. You can associate a task with any defined event templates.

We recommend that you do not delete the predefined tasks.

Delete tasks

You can delete a task at any time; you do not have to remove the event template associations first.



Running the Dock Door Receiving scenario

In this chapter, we explain the Dock Door Receiving scenario that comes with the Premises Server. We tell you what you have to do to run it without modification and discuss the way we ran it in our ITSO lab.

8.1 Before you begin

Before you run the Dock Door Receiving scenario and attempt to read RFID tags, be sure you have done the following:

- ▶ Installed all the Premises Server according to the instructions in Chapter 6, “Installing the WebSphere RFID solution” on page 111.
- ▶ Installed the edge devices and Device Infrastructure software: an Edge Controller and at least one RFID reader with reader I/O devices.

You need a way to send (or simulate) input for switch and motion activity to the reader and send (or simulate) light or beep output from the reader.

- ▶ Obtained RFID tags to read.

In order to test RFID tag readers, you need to have the appropriate tags. Not all readers read all tags. Consult with the manufacturer of your RFID reader or your IBM representative to determine the types of tags you need and where you can get them.

At some point, before you actually use the RFID system in your environment, you will want to provide communication between the Premises Server and your enterprise system. This communication enables your RFID solution to respond appropriately to expected and unexpected tags. Because each customer enterprise system varies greatly, integration of an enterprise system with the IBM RFID solution is beyond the scope of this book.

If your RFID solution is not yet integrated with your back-end system, you can still run the Dock Door Receiving scenario using the simulator that is included with the Premises Server installation. In the sections that follow, we tell you how to do this.

8.2 Overview of the scenario

The Dock Door Receiving scenario is a sample set of basic activities that are commonly performed to receive items that have been labeled with RFID tags. It includes reading tags using the devices at a defined location, validating the tags data in some way, and showing the results of that validation at the same location.

This scenario represents a very small part of any real-life, end-to-end supply chain process. Nevertheless, it is a useful scenario to run. It gives you a quick and simple way to test your devices and helps you to become familiar with the IBM RFID solution. It also provides you with a base infrastructure which can be extended and customized to suit individual needs.

The IBM WebSphere RFID solution includes the Premises Server software that is required to run this scenario. It also includes a license for you to download the required edge software, called RFID Reference Sample Implementation V1.0 (Kimono). Many of the components of this sample implementation use *Kimono* as part of their name.

Figure 8-1 provides a visualization of the scenario at an entry into a warehouse, such as the doorway of a receiving dock. In our scenario, this entry area, which is sometimes called a *portal*, has an ON/OFF switch, a motion sensor, an RFID reader with one or more antennas, and a light tree.

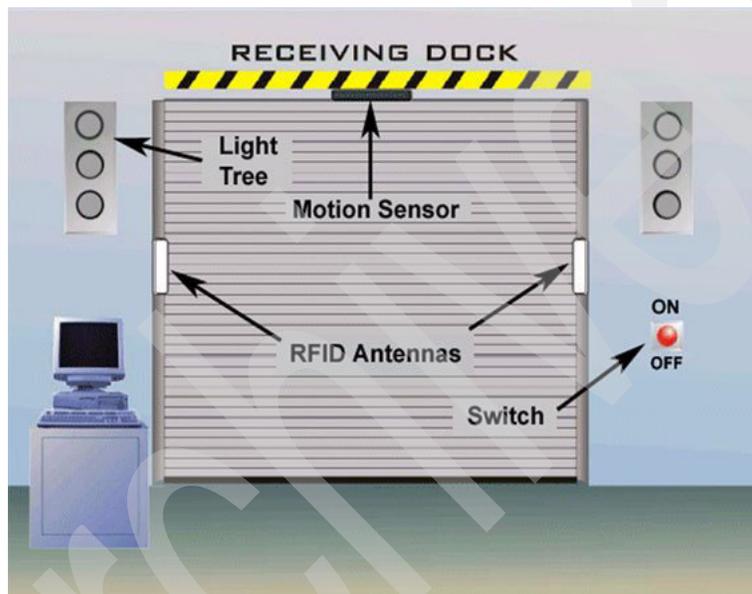


Figure 8-1 Dock Door Receiving Scenario

Using the agents and associations that are defined in the network topology, the devices at this location communicate with the Edge Controller and this, in turn, communicates with the Premises Server.

If all the devices are in place and the network components are configured properly, the scenario follows this course as illustrated in Figure 8-2 on page 184:

- ▶ You turn on a switch to open the door and notify the Edge Controller.
- ▶ When the first item passes through the door, the motion sensor activates the reader.
- ▶ As the RFID tags are read, the tag data is captured by the antennas and the reader sends it to the Edge Controller.

- ▶ The Edge Controller filters the tags and sends the filtered tag data to the Premises Server.
- ▶ The Premises Server sends the tag data to the enterprise system.
- ▶ The enterprise system evaluates the tags and sends results back to the Edge Controller.
- ▶ The light tree beeps or flashes the appropriate color.

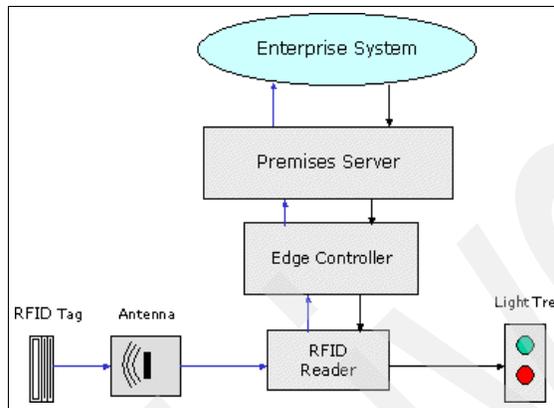


Figure 8-2 End-to-end message flow

For a more detailed discussion of these events, refer to 8.2.5, “Behind the scenes” on page 199.

8.2.1 Using the default network topology

To run the Dock Door Receiving scenario requires minimal configuration changes. The Premises Server comes with default definitions for a location (P1), a reader (R1), and edge controller (E1). It also has defined tasks, event templates, and output channels to enable communication from the Edge Controllers to the Premises Server, the enterprise, and back.

If you want to use the default definitions, you need to:

- ▶ Modify the R1 reader definition to specify the correct Reader Type, its IP address on your network, and correct port number for that reader type.
- ▶ Verify the input and output settings on your reader agent, and change them if necessary.
- ▶ Verify your filter agent settings, and change them if necessary. These settings determine which tag data is not passed to the Premises Server.
- ▶ Enable or simulate a back-end system so that the Premises Server knows which tags to expect and can respond accordingly.

See “Search, display, update, and delete readers” on page 159 for information about modifying a reader definition.

See “View and edit agent properties” on page 162 for information about modifying agents.

8.2.2 Configuring our network topology

In the ITSO lab, we had two readers and two Edge Controllers, so we deleted the defaults and created our own definitions for locations, readers, and controllers.

We chose to use location contacts and location, reader, and controller names that reflected our actual environment, as shown in Table 8-1.

Table 8-1 Locations and contacts for the ITSO lab controllers and readers

Controllers	Readers	Locations	Contact
--	--	Bldg662	James
ArcomA	Alien1A	A305-21-1A	Sam
ArcomB	Intermec1B	A305-21-1B	Eric

Locations and location contacts

We created three location definitions (Figure 8-3):

- ▶ Bldg662 represents the building where our lab was located. This could be the equivalent of a warehouse.
- ▶ A305-21-1A and A305-21-1B are the alcoves in the lab where our readers were located. These could be the equivalent of specific dock door areas.

Locations

Location entries are used to define the physical and logical locations of the components (reader, etc.) of the RFID system.
Each Location entry has a container location and may have one or more contained locations.
The Locations are displayed below with the container locations at the left and their contained location indented to the right.

To add a Contained Location to a Location or to view the details of a Location, click the id in the tree below.
Root Location is the top-level container and cannot be modified, but contained locations can be added to it by clicking on it.

[Root Location](#)
 [Bldg662](#)
 [A305-21-1A](#)
 [A305-21-1B](#)

Figure 8-3 ITSO lab locations

We created three location contacts (Figure 8-4), one for each of those locations.

Location Contacts

To view the contact details, click on the Name of the Contact. The details screen will allow you to update or delete the contact.
To create a new contact, press Create.

Create

Name	Phone	Mobile	Pager	Email
No Contacts Defined				

Figure 8-4 ITSO lab location contacts

Readers

We had two readers, an Alien and an Intermec. We created two reader definitions, one at each location (Figure 8-5).

Restriction: There cannot be more than one reader at a location.

Readers

Reader entries are used to define and configure the RFID tag readers connected to the system. A Reader entry consists of a unique id/name and other settings.

To filter the readers displayed below by id, enter a filter value and press Search.
Filtering is case-sensitive (there is an implicit wildcard at the beginning and end of the filter string)

Filter: Search

To create a new Reader entry, press New.
To edit the configuration of an existing Reader, click the id of the Reader in the table below.

New

Reader id	Location
Alien1A	A305-21-1A
Intermec1B	A305-21-1B

Figure 8-5 ITSO lab readers

Reader agents

We looked at the reader agents for our Alien and Intermec readers to verify their property settings.

AlienReaderAgent

An Alien reader can have four possible pin settings (Figure 8-6) for input (0-3) and four possible pin settings for output (0-3). The default settings for inputs (motion and switch) are 1 and 0. The default settings for outputs (red, amber, green, beep) are 0, 1, 2, 3.

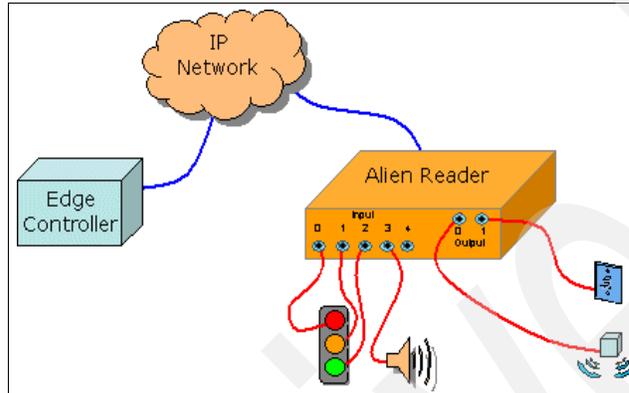


Figure 8-6 Alien reader pin settings

We accepted all the defaults for the AlienReaderAgent, as shown in Example 8-1.

Example 8-1 Default values for AlienReaderAgent

```
green = 2
red = 0
amber = 1
heartbeat.period.ms = 10000
alien.pollingreadrate = 666
alien.pollingppiorate = 250
motion = 0
transport.connection = com.ibm.esc.tcpi.connection.TcpiConnection
inputpins = switch,motion
switch = 1
io.type = reader
beep = 3
```

IntermecReaderAgent

By default, the IntermecReaderAgent has the same values as the AlienReaderAgent but, because the Intermec reader has a different physical I/O port configuration, these values are problematic. The Intermec reader has four possible settings (0-3) for both the inputs and the output pins, so we made the changes (highlighted in bold as shown in Example 8-2).

Example 8-2 IntermecReaderAgent settings

```
green = 2
red = 3
amber = 99
heartbeat.period.ms = 10000
alien.pollingreadrate = 666
alien.pollingppiorate = 250
motion = 0
transport.connection = com.ibm.esc.tcpip.connection.TcpipConnection
inputpins = switch, motion
switch = 1
io.type = reader
beep = 99
```

Note: The value 99 is out of range for a pin setting. It ensures that this property is disabled.

Filtering agent

We had three Intermec RFID tags to use for testing the Dock Door Receiving scenario, two case tags and one pallet tag, and we wanted to read both types of tags.

FilterAgent

We looked at filter agent to check its default property settings:

```
duplicates.decay.limit.sec = 5
duplicates.decay.cleanup.sec = 2
filters = Duplicates,CaseTags
```

The filters property setting specifies what the Edge Controller should ignore and not send to the Premises Server, therefore we changed the setting of this property to filter out duplicates but allow case tags to be read:

```
filters = Duplicates
```

Note: A tag is considered a *duplicate* if it is read more than once during the same on/off switch session.

Controllers

We had two different types of readers, Alien and Intermec. Therefore, we needed an Edge Controller for each one. We created two controller definitions as shown in Figure 8-7.

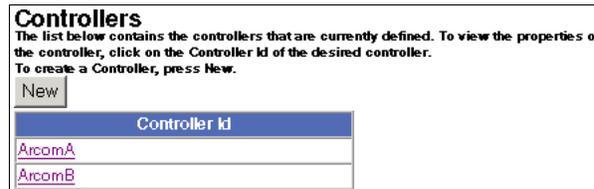


Figure 8-7 ITSO lab controllers

We associated ArcomA with Alien1A reader at location A305-21-1A and ArcomB with Intermec1B at location A305-2-1B.

Other configuration tasks

We also used the Administrative Console to view the tags that were read and saved in the Premises Server database. We followed this process to edit the `premises.properties` file and to change the tag persistence parameter:

1. Open the following:
`IBM-RFID-HOME\RFID\premises\eventserver\properties\premises.properties`
2. Set the following:
`com.ibm.rfid.premises.tag.persistence=true`
3. Restart WebSphere Application Server.

Example 8-3 shows the section of the file that we edited.

Example 8-3 Setting tag persistence

```
#####  
# com.ibm.rfid.premises.tag.persistence  
#  
# Indicates if the Premises Server should persist tag and tag  
# history information in the database. Values:  
# true = persist tag and tag history  
# false = no persistence  
#  
# Default = false  
#####  
com.ibm.rfid.premises.tag.persistence=true  
#####
```

8.2.3 Simulating a back-end system

Because we do not have the equivalent of an enterprise system in our ITSO lab, we used the `PremisesTest` bundle, which you can configure through the `premises-test.properties` file that comes with the Premises Server. This bundle can simulate three of the RFID domains: a reader, an Edge Controller, and a back-end system. We used this file to simulate a back-end enterprise system. This section explains how we did this.

For more information about this file, see 6.2.5, “Using `KimonoPremisesTest` bundle” on page 138.

Find RFID tag data

We read our RFID tags to determine their actual values, so we could put these values in the simulator file:

```
RFID_HOME\RFID\edgecontroller\premises\smf\premises-test.properties
```

View the Premises Server SMF console

To view the Premises Server SMF console, follow these steps:

1. On the Premises Server, stop the SMF service.
2. Start SMF by running `smf.bat`:
 - From a command prompt, change to the SMF directory:
`IBM-RFID-HOME\edgecontroller\premises\smf`
 - Then enter:
`smf.bat`
3. To show the bundles that are running, enter:
`SMF> ss`
4. Verify that `KimonoConsoleLog` [26] and `KimonoPremisesTest` [25] are `RESOLVED`, as shown in Example 8-4.

Example 8-4 Premises Server SMF console

```
Framework is launched.
id      Type      State      Bundle
27      .jar      ACTIVE    smfbd:/PremisesLoggingConnector [27]
26      .jar      RESOLVED  smfbd:/KimonoConsoleLog [26]
25      .jar      RESOLVED  smfbd:/KimonoPremisesTest [25]
24      .jar      ACTIVE    smfbd:/MBAF [24]
23      .jar      ACTIVE    smfbd:/win32service [23]
21      .jar      ACTIVE    smfbd:/MicroBrokerBridgeManager [21]
20      .jar      ACTIVE    smfbd:/KimonoPremisesBridge [20]
19      .jar      ACTIVE    smfbd:/Rfid [19]
```

18	.jar	ACTIVE	smfbd:/MicroBrokerBridgeJMS [18]
17	.jar	ACTIVE	smfbd:/MicroBrokerBridge [17]
15	.jar	ACTIVE	smfbd:/MicroBrokerManager [15]
14	.jar	ACTIVE	smfbd:/MicroBroker [14]
13	.jar	ACTIVE	smfbd:/ConfigurationAdmin [13]
12	.jar	ACTIVE	smfbd:/EventLog [12]
11	.jar	ACTIVE	smfbd:/MicroBrokerTrace [11]
10	.jar	ACTIVE	smfbd:/MQTelemetryTransport [10]
9	.jar	ACTIVE	smfbd:/MicroBrokerRegistry [9]
8	.jar	ACTIVE	smfbd:/OSGi-SPR3-ServiceTracker [8]
7	.jar	ACTIVE	smfbd:/LogService [7]
6	.jar	ACTIVE	smfbd:/SMFBundleMessages [6]
5	.jar	ACTIVE	smfbd:/PersistenceManager [5]
2	.jar	ACTIVE	smfbd:/OAF_Base [2]
1	.jar	ACTIVE	smfbd:/OSGi-SPR3-ServiceInterfaces [1]
0		ACTIVE	System Bundle [0]

5. To start the SMF console log, enter:

```
SMF> start 26
```

The KimonoConsoleLog [26] should now be active, as shown here:

```
26 .jar ACTIVE smfbd:/KimonoConsoleLog [26]
```

Scan RFID tags

To scan RFID tags, do the following:

1. At your edge location, turn your switch ON and activate your motion sensor.
2. Scan each tag once.
3. To stop reading, deactivate your motion sensor and turn your switch OFF.

Note: The `delayafterquiet` property of the `MotionSensorAgent` sets the period of time after the motion sensor starts that a reader attempts to read tags. Depending on this setting, your reader might stop by itself.

Copy the tag data

After we read the tags we copied the tag data from the console to put in our list of accepted tags. Because we wanted to test an invalid tag, we only copied one pallet tag and one case tag.

1. Return to the Premises Server SMF console and look through the output for XML messages with `type='tag_read'` as shown in Example 8-5.

Example 8-5 XML message example with type='tag_read'

```
Intermec1Bx
[INFO] 2005-10-20 12:23:48.359 - TagReadTransformation@44625bb6: doTransform() -> transformed
message (after decoding) = properties
{msgDuplicate=false, qos=1, msgID=4,
topicProperty=receiving/portal/A305-21-1B/signal/tags},resourceName=EDGE.IN.Q,
sourceResourceName=receiving/portal/A305-21-1B/signal/tags,body=<?xml version='1.0'
encoding='UTF-8'?><ibmprem:ibm-premises-unified-format dts='2005
10-20T12:23:48' xmlns:ibmprem='http://www.ibm.com'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xsi:schemaLocation='http://
www.ibm.com IBMPremisesUnifiedMessageFormat.xsd'><event location='A305-21-1B'
type='tag_read'><rfid-tag-data antenna='0' count='1'
discovered='960282721899' reader='Intermec1B' tagid='ef04304000000020' /></event>
</ibmprem:ibm-premises-unified-format>
```

2. Copy the value of tagid. For example:

ef04304000000020

Edit the premises-test.properties file

We pasted the tag data into the simulator file. We also made other necessary changes to use just the back-end simulation and not use the Premises Server and Edge Controller simulation. To edit the `premises-test.properties` file:

1. Open the following file for editing (Example 8-6):
`IBM-RFID-HOME\RFID\edgecontroller\premises\smf\premises-test.properties`
2. Look for the list of expected tags in the properties section of `DockDoorReceivingEventHandler.java`.

Example 8-6 premises-test.properties file

```
#####
#
# Properties for DockDoorReceivingEventHandler.java
#
# id      - the ID of the handler
#          (DEFAULT = <fully qualified class name of the handler>)
# message-the format of the validation response sent to the Premises Server
```

```

#           (DEFAULT = <?xml version="1.0"
encoding="UTF-8"?><dock-door-receiving-response><response
#           result="~" location="~" original_timestamp="1085558889990"
date="2004-07-01T21:32:53"
#           status="2"/></dock-door-receiving-response>)
# message.regex-the regular expression indicating the places where dynamic data should be
inserted
#           (DEFAULT = ~)
# queue - the queue on which the validation response should be placed
#           (DEFAULT = KIMONO.RESPONSE.Q)
# queue.manager-the queue manager on which the queue resides
#           (DEFAULT = IBM.RFID.QM)
# tag.<#>-a tag ID (e.g., EPC) that should be indicated as valid by the handler; see General
Note (1);
#           note that these values are treated as Strings and must represent what will
actually be
#           reported by the Premises Server (e.g., if the Premises Server reports EPCs in
hexadecimal format
#           then these values should also be given in hexadecimal format)
#####
premises.test.event.handler.dockdoorreceiving.id=DockDoorReceivingEventHandler
premises.test.event.handler.dockdoorreceiving.message=<?xml version="1.0"
encoding="UTF-8"?><dock-door-receiving-response><response result="~" location="~"
original_timestamp="~" date="~" status="~"/></dock-door-receiving-response>
premises.test.event.handler.dockdoorreceiving.message.regex=~
premises.test.event.handler.dockdoorreceiving.queue=KIMONO.RESPONSE.Q
premises.test.event.handler.dockdoorreceiving.queue.manager=IBM.RFID.QM
premises.test.event.handler.dockdoorreceiving.status=2
premises.test.event.handler.dockdoorreceiving.tag.0=3114f4e4e45ca3c002000000
premises.test.event.handler.dockdoorreceiving.tag.1=3114f4e4e45ca3c003000000
premises.test.event.handler.dockdoorreceiving.tag.2=3114f4e4e45ca3c005000000
#####

```

3. Paste your tag values into the list. You can modify the tags that are there or add additional tags. Example 8-7 shows the changes that we made so that our list had two of our tags in it.

Example 8-7 premises-test.properties file (modification 1)

```

#####
premises.test.event.handler.dockdoorreceiving.id=DockDoorReceivingEventHandler
premises.test.event.handler.dockdoorreceiving.message=<?xml version="1.0"
encoding="UTF-8"?><dock-door-receiving-response><response result="~" location="~"
original_timestamp="~" date="~" status="~"/></dock-door-receiving-response>
premises.test.event.handler.dockdoorreceiving.message.regex=~
premises.test.event.handler.dockdoorreceiving.queue=KIMONO.RESPONSE.Q
premises.test.event.handler.dockdoorreceiving.queue.manager=IBM.RFID.QM
premises.test.event.handler.dockdoorreceiving.status=2
premises.test.event.handler.dockdoorreceiving.tag.0=ef04304000000020
premises.test.event.handler.dockdoorreceiving.tag.1=ef0430400000001f
#####

```

4. Scroll to the properties for Activator.java at the top of the file. This section defines the simulator classes used in the file:
 - EventHandlerFactory simulates a back-end enterprise system
 - CommandFactory simulates the Premises Server
 - EventSimulatorFactory simulates the Edge Controller
5. Comment out the lines with the CommandFactory and EventSimulatorFactory that simulate the Premises Server and Edge Controller, as shown in Example 8-8.

Example 8-8 premises-test.properties file (modification 2)

```
#####
#
# Properties for Activator.java
#
# To disable a particular component in its entirety, comment out the appropriate line.
#####
premises.test.factory.class.0=test.com.ibm.kimono.premises.event.handler.EventHandlerFactory
#premises.test.factory.class.1=test.com.ibm.kimono.premises.command.CommandFactory
#premises.test.factory.class.2=test.com.ibm.kimono.premises.event.simulator.EventSimulatorFacto
ry
#####
```

This leaves just the EventHandlerFactory, which simulates a back-end enterprise system to tell us if a tag we read is expected.

6. Save your changes and close premises-test.properties.

Enable the simulator

Then we returned to the SMF console to start the simulator. To enable the simulator:

1. Show the bundles that are running (Example 8-9) by typing the following command:

```
SMF> ss
```

Example 8-9 Output from starting the simulator

Framework is launched.

id	Type	State	Bundle
27	.jar	ACTIVE	smfbd:/PremisesLoggingConnector [27]
26	.jar	ACTIVE	smfbd:/KimonoConsoleLog [26]
25	.jar	RESOLVED	smfbd:/KimonoPremisesTest [25]
24	.jar	ACTIVE	smfbd:/MBAF [24]
23	.jar	ACTIVE	smfbd:/win32service [23]
21	.jar	ACTIVE	smfbd:/MicroBrokerBridgeManager [21]

20	.jar	ACTIVE	smfbd:/KimonoPremisesBridge [20]
19	.jar	ACTIVE	smfbd:/Rfid [19]
18	.jar	ACTIVE	smfbd:/MicroBrokerBridgeJMS [18]
17	.jar	ACTIVE	smfbd:/MicroBrokerBridge [17]
15	.jar	ACTIVE	smfbd:/MicroBrokerManager [15]
14	.jar	ACTIVE	smfbd:/MicroBroker [14]
13	.jar	ACTIVE	smfbd:/ConfigurationAdmin [13]
12	.jar	ACTIVE	smfbd:/EventLog [12]
11	.jar	ACTIVE	smfbd:/MicroBrokerTrace [11]
10	.jar	ACTIVE	smfbd:/MQTelemetryTransport [10]
9	.jar	ACTIVE	smfbd:/MicroBrokerRegistry [9]
8	.jar	ACTIVE	smfbd:/OSGi-SPR3-ServiceTracker [8]
7	.jar	ACTIVE	smfbd:/LogService [7]
6	.jar	ACTIVE	smfbd:/SMFBundleMessages [6]
5	.jar	ACTIVE	smfbd:/PersistenceManager [5]
2	.jar	ACTIVE	smfbd:/OAF_Base [2]
1	.jar	ACTIVE	smfbd:/OSGi-SPR3-ServiceInterfaces [1]
0		ACTIVE	System Bundle [0]

2. Start KimonoPremisesServerTest bundle by typing the following command:

```
SMF> start 25
```

At this point, we were ready to use the Dock Door Receiving scenario to read tags and receive responses, and test our RFID solution.

8.2.4 Using our Dock Door Receiving scenario

We attached our three tags to their respective containers, knowing that two tags were valid (they were in the list of expected tags) and that one was not. We began the receiving process at location A305-21-1B. We then moved the containers past the reader antennas manually.

Open the dock door

To open the dock door, we took the following action:

Action

We clicked the ON/OFF switch  to open the dock door and to notify the Edge Controller (Figure 8-8).

We then moved one container with a valid pallet tag through the door with the following result:

Result

The motion sensor above the door detected the movement and our reader was activated.

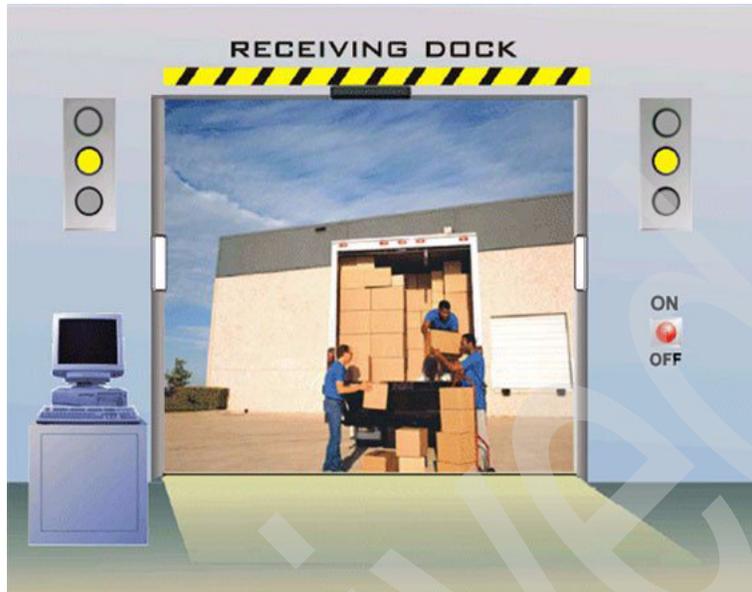


Figure 8-8 Activating the edge devices

Read valid tags

To read valid tags, we took the following action:

Action We moved that container with a valid pallet tag past the antennas (Figure 8-9 on page 197).

We know that if a tag was not filtered by the Edge Controller, the tag data is passed to Premises Server and then to the (simulated) back-end system, which has a list of expected tags. The back-end system lets the Premises Server know whether the tag was expected (valid). The Premises Server sends this message back to the edge devices, with the following result:

Result The first tag that was read was an expected pallet tag. The green light came on, indicating that the pallet should be unpacked and processed into inventory.

We repeated this process for the valid case tag with the same result.



Figure 8-9 Reading valid tags

Read an invalid tag

To read an invalid tag, we took the following action:

Action We passed the container that has the invalid case tag through the door (see Figure 8-10 on page 198).

We know that we did not enter this case tag in the list of valid tags in the premises-test.properties file and it should be rejected, with the following result:

Result This tag was not expected and the red light came on, indicating that the case should be put aside.



Figure 8-10 Reading an invalid tag

Viewing the tag data

Because we set `tag.persistence=true` in the `premises.properties` file, we could view the tags that were read from the Administrative Console. This view does not show whether the tags were expected or unexpected, just that they were read. To view the tag data, you need to do the following:

1. To open the Administrative Console, use a Web browser and go to the following URL:
`http://premises_server_ip:9080/event_admin_web`
where `premises_server_ip` is the IP address of your Premises Server.
2. Click **Tags** in the Administrative Console navigation frame. The Tags page is displayed.

Figure 8-11 shows the tag data that we saw on the Administrative Console Tags page.

Tags

A Tag consists of a unique ID and a history.
 To filter the tags displayed below by ID, enter a filter value and press Search.
 Filtering is case-sensitive (there is an implicit wildcard at the beginning and end of the filter string)

Filter: Search

To view the details of a Tag, click on its Tag Id in the table below.
 To view the tag's history, click on its Tag History link.

Tag Id	Tag History	Date/Time Stamp
ef04304000000020	premises_cit11298336969842	Tue Jun 06 07:29:49 EDT 2000
ef0430400000001f	premises_cit11298336932971	Tue Jun 06 07:29:46 EDT 2000
ef04304000000025	premises_cit11298336886560	Tue Jun 06 07:29:40 EDT 2000

Figure 8-11 Viewing tag data

8.2.5 Behind the scenes

This list of events explains what happens throughout the Dock Door Receiving scenario. It includes the activities that you can see and those that occur behind the scenes at a software component level.

Note: In this description, the term *Connection Server* refers to the WebSphere Everyplace Connection Server. All the agents reside on the Edge Controller.

The list of events is as follows:

1. An attendant at the location turns the switch ON.
2. The switch agent changes the status to ON.
3. The motion sensor is tripped by the movement of an item at that location.
4. The motion sensor agent changes its status to ON.
5. The controller agent, which subscribes to the motion sensor events, sends a command to the reader agent to begin reading tags.
6. The reader agent starts reading and sends the RFID tag data to the Connection Server.
7. The Connection Server sends the tag information to the filter agent.
8. The filter agent removes duplicate reads or case tags, as needed, and sends a filtered set of tags to the Connection Server.
9. The Connection Server sends the filtered tag data to the Premises Server.
10. The Event Server application running on the Premises Server receives the tag information.
11. The data for the tags and the locations from which they were read are sent to the enterprise back-end system for evaluation. This evaluation is a comparison against a list of expected tags/items.

12. The enterprise back-end system responds with an ACCEPT or REJECT message for the items in the list.
13. The Premises Server formats the response and sends it to the Connection Server.
14. The controller agent receives the response.
15. If the response is ACCEPT, a green light command is sent through the Connection Server to the light tree agent associated with the location, and the green light turns on. If the response is REJECT, a red light command is sent through the Connection Server to the light tree agent associated with the location and the red light turns on.
16. After a period of motion sensor inactivity, the motion sensor agent changes its status to OFF and the controller agent sends a command to the reader agent to stop reading tag data from the reader. If the attendant turns the switch OFF before this period has expired, the reader agent would stop reading at that point. When the reader stops, the filter agent is reset.

8.2.6 Extending the Dock Door Receiving scenario

Dock Door Receiving is a non-modified, simple, and limited process that offers the fundamental building blocks of an RFID system. There are many ways that you can extend it and customize it to make it an integrated and valuable part of your enterprise environment. Here are a few suggestions:

- ▶ Integrate with your back-end enterprise system and databases. Connect to your store of shipment and inventory information.
- ▶ Expand the current validation, which only indicates expected tags and unexpected tags at a package or container level. For example, you could add validation of the associations, such as between packages and their containers.
- ▶ Modify the processing logic to trigger business work flow and perform specific actions for expected or unexpected tag reads.
- ▶ Build a graphical user interface to remotely administer the devices at a location. This could enable you to activate edge devices and view their status without having to be in their proximity.
- ▶ Add a reporting mechanism to save the tag data that you read and present the results in a meaningful way.

Note: Keep in mind that to implement any extensions of this type requires the assistance of an IBM Services team or an IBM Business Partner.



Part 3

Advanced deployment topics

In this part, we explain what we did to configure our WebSphere RFID system and run the Dock Door Receiving sample scenario. We show you the variables that we set, the tasks that we performed, and the results that we obtained. We also offer suggestions for extending this reference implementation to meet the needs of your enterprise.

Archived

Performance

This chapter discusses the performance of an RFID solution, both from an end-to-end system perspective as well as at the component level. It introduces topics that you should consider when planning an RFID solution. In addition, it provides information about how to optimize the performance of each domain in order to achieve the most efficient RFID solution possible.

We consider the following areas:

- ▶ System-wide performance
- ▶ Premises Server performance
- ▶ Edge Controller performance
- ▶ Reader performance

9.1 RFID system performance

Many factors influence the performance of a fully deployed RFID solution, such as the following:

- ▶ The type of network over which the solution is deployed
- ▶ The nature of the back end system (Integration Domain)
- ▶ The intricacy of message routing, between and within system domains
- ▶ The performance of the individual components
- ▶ The load under which the system is stressed (tag reads, I/O usage, and so forth)

While the investigation of these and other performance impacts can become a science in and of itself, the answers to some questions are obvious. For example, a solution that is deployed over a LAN performs better than over a WAN, a Premises Server achieves higher performance with a faster CPU, and so on. However, some areas are not so clear. This chapter aims to define and to offer insight into such areas.

9.1.1 System performance breakdown

As with any component based system, the individual parts of an RFID solution must operate at peak performance in order to achieve the best possible system performance. This section investigates the overall system performance, and in doing so, reveals how each domain (Integration, Premises, Edge, and Reader) contributes to the overall system performance.

In order to perform this breakdown, we put a simple tag validation scenario under the microscope. This scenario starts with a tag read event at some location and ends with the visible or audible response from the outputs at the same location where the tag was read. Basically, the flow of information between domains in this scenario is as follows:

1. A tag is read by the RFID reader, *R*.
2. The tag data is filtered by the Edge Controller, *E*.
3. The filtered tag data is sent from *E* to the Premises Server, *P*.
4. The tag data is filtered further by *P* and sent to the Integration Domain.
5. The filtered tag data is evaluated by the Integration Domain, and a response is sent back to *P* from the Integration Domain.
6. The response is formatted by *P* and sent to *E*.

7. The response is converted into an output trigger by *E* and is displayed at the outputs of *R*.

In this manner, the tag data flows up the domains until it reaches the Integration Domain, otherwise known as the Enterprise back-end system, where it is converted into a response that flows back down the domains. Refer to Chapter 8, “Running the Dock Door Receiving scenario” on page 181 for more information about tag reads and message flow in this scenario, which constitutes a single run of the Dock Door Receiving scenario.

Figure 9-1 shows a typical breakdown of the relative performance of various components during the scenario described above.

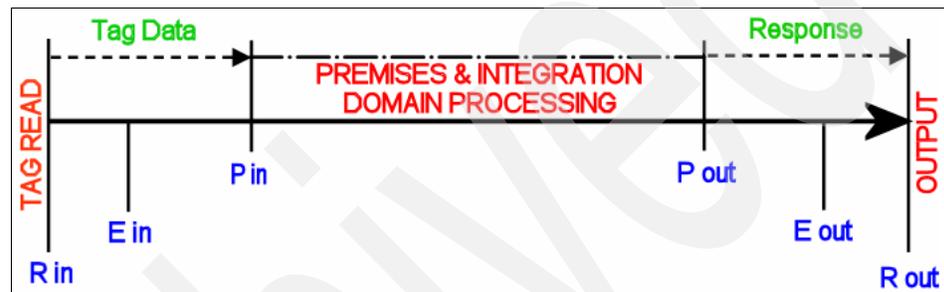


Figure 9-1 System performance breakdown

This time line represents the relative performance in a fully optimized system. Most users are concerned mainly with the end-to-end system response time, meaning the time it takes for the response to be displayed at the reader after a tag is read by that reader, represented here by (R out) - (R in). However, the optimization of some of the segments in between these two points can have a drastic effect on this total response time.

Table 9-1 defines the stages from left to right in Figure 9-1.

Table 9-1 Stages in system performance breakdown

Stage	Definition
R in	The RFID reader reads a tag.
E in	The Edge Controller receives the tag read.
P in	The Premises receives the filtered tag data from the Edge Controller.
P out	The Premises forwards the response to the Edge Controller.
E out	The Edge Controller enables the RFID reader output.
R out	The final output by the RFID reader.

Obviously, the bulk of the end-to-end message flow is concentrated in the Premises and Integration Domain Processing, which greatly depends on the nature of the enterprise back-end system that is used by the Integration Domain to evaluate the tag read and issue a response. For example, if the tag data is compared to a short list that is stored locally, the processing time is a great deal faster than using a nationwide enterprise application that accesses a remote database to evaluate the tag data and issue a response.

Likewise, the RFID readers, Edge Controllers, and Premises Servers all communicate through TCP/IP so the networks that connect each of these components can either enhance or hinder the performance of the segment of the time line associated with that domain.

The remainder of this chapter discusses some end-to-end system test results as well the performance of each of the domains that are represented on the time line.

9.1.2 System performance test results

In our testing, we executed some simple end-to-end performance tests on a complete IBM WebSphere RFID solution that we configured to use a single Premises Server, a single Edge Controller, and a single RFID reader. All of these components ran on the same network, and tag validation was done locally by a back-end simulator running on the Premises Server. In this manner, we isolated test results to the IBM WebSphere RFID solution that were independent of any user specific Integration Domain processing performance. We used the following hardware:

- ▶ RFID reader simulator running on Windows 2000 Server machine
- ▶ Edge Controller running on the Arcom Viper platform (embedded Linux)
- ▶ Premises Server running on Windows 2000 Server with Service Pack 4
 - Dual 2.8 GHz CPUs
 - 3.5 GB RAM

End-to-end system response time

In this test, the RFID reader simulator sent tag reads and measured response times. We found that the end-to-end system response time for a single tag read was well below two seconds, such that (R out) - (R in) was less than two seconds.

Premises Server tag handling rate

The RFID reader simulator also had the ability to send tag reads at ever increasing rates in order to stress the system. In this way, we found that the

maximum tag read rate that the Premises Server could handle was around five tags per second. Thus, the rate at which tag data reaches the Premises Server (P in) cannot exceed five times per second in order to uphold the end-to-end system response time of two seconds.

9.2 Premises Server performance

The Premises Server and back-end processing domains consume the most time during end-to-end message transmission. As a result, this segment of message flow has the highest impact on system performance. Thus, optimizing these legs of the trip is of utmost concern. While the back-end system is largely dependent on the RFID user's existing enterprise infrastructure, some measures can be taken to optimize Premises Server performance. This section shows you how.

9.2.1 Running SMF as a service

By default, SMF runs as a service. If SMF is being run from the command line for any reason, it will degrade Premises Server performance, especially if the ConsoleLog bundle is running. For this reason, it is recommended that SMF always be run as a service, unless it is absolutely necessary to run it from a command prompt for testing or debugging purposes.

9.2.2 Reducing and eliminating logging

Extensive logging can bog down any application, and the Premises Server is no exception. It is recommended that you set all logging on the Premises Server to the most restrictive levels and even disable logging when possible to ensure peak performance in a production environment. Of course, for testing and debugging situations, you can enable and configure logging as needed.

The sections that follow explain how to configure the Premises Server logging for optimal performance.

Restrict SMF Logging

By default, SMF logging is set to the most restrictive level. However, this level might have been modified during installation or testing. If this is the case, you can follow these steps to limit the amount of logging performed by the Services Management Framework runtime on the Premises Server:

1. Open the PremisesLoggingConnector.properties file for editing. It is located in the *IBMRFD*\edgecontroller\premises\smf directory.

2. Edit the last line in the file to read as follows:

```
com.ibm.rfid.premises.logging.file.level=SEVERE
```
3. Save and close the file.
4. Restart SMF if it is already running, for the new logging level to take effect.

Disable WebSphere Application Server Diagnostic Trace

There are many levels of diagnostic tracing available on WebSphere Application Server, and extensive use can impede application performance. For optimum performance, follow these steps to ensure that all Diagnostic Trace is disabled.

1. Access the WebSphere Application Server Administrative console at:
<http://PremisesIP:9090/admin>
2. Click **Troubleshooting** → **Logs and Trace** → **server1** → **Diagnostic Trace** to arrive at the Diagnostic Trace Service page, as shown in Figure 9-2.

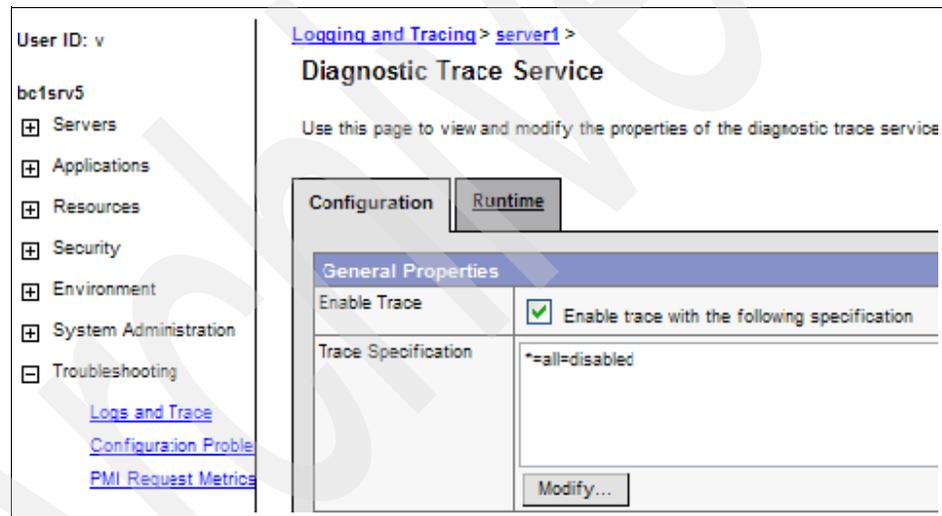


Figure 9-2 WebSphere Application Server Diagnostic Trace settings

3. Switch to the Runtime tab if you want to apply changes to the current WebSphere Application Server runtime. Otherwise, you must restart WebSphere Application Server for any changes to take effect.
4. Deselect Enable Trace or ensure that the Trace Specification is:

```
*=all=disabled
```

Attention: This Diagnostic Trace Service Trace Specification is applied to *all* applications that run on WebSphere Application Server. Alternatively, you can click **Modify** to fine-tune the Trace Specification values for each group or component of any application as needed. Keep in mind, however, that completely disabling the Diagnostic Trace service results in the best possible Premises Server performance.

5. Apply and save any changes.
6. Restart WebSphere Application Server if necessary.

Restrict Premises test logging

If you happen to be using the Premises Test bundle in the Premises Server SMF runtime, you might want to restrict its logging level to improve performance. Generally, this bundle is only used for testing purposes, so it should not be running in your production environment. Nonetheless, to change its logging level, you need to follow these steps:

1. Open the `premises-test.properties` file for editing from the following directory:
`IBMRFID\edgecontroller\premises\smf`
2. Edit the last line in the file to read as follows:
`premises.test.logging.level=error`
3. Save and close the file.
4. Restart the Premises Test bundle if SMF is running from the console. If it is running as a service, you must restart SMF altogether.

Refer to 8.2.3, “Simulating a back-end system” on page 190 for more information about the Premises Test bundle.

Disable Application Level Event logging

If you are using the PVS Technical Preview on your Premises Server, you might want to disable ALE logging as well. By default, ALE logging is set to the most verbose level. Follow these instructions to suppress its output:

1. Open the `premises-test.properties` file for editing from the following directory:
`IBMRFID\edgecontroller\premises\smf`
2. Edit the last line in the file to read as follows:
`com.ibm.rfid.ale.engine.logLevel=OFF`
3. Save and close the file.
4. Restart WebSphere Application Server to employ the new logging level.

9.2.3 Known issues and limitations

Due to the nature of tag validation, the Premises Server deals with messages in a first in first out (FIFO) fashion. The value of the `maxsessions` and `maxmessages` parameters of the WebSphere Application Server message listener ports are set to one and must not be changed. These settings require that messages be processed one at a time in the order that they are received to support this design. This is the main influence on the maximum tag handling rate of the Premises Server, as described in the 9.1.2, “System performance test results” on page 206.

9.3 Edge Controller performance

The performance of an Edge Controller depends almost solely on the number of messages that it must process. These messages are those on the topics to which Edge Controller agents are either published or subscribed, and their number is determined by the following factors:

- ▶ Agent design (for example the topics to which they publish and subscribe)
- ▶ RFID solution scale (for example the number of devices per Edge Controller)
- ▶ RFID solution stress (for example device activity, tag read rates, and so forth)

While these factors differ for every solution, it should be obvious that the fewer devices an Edge Controller must control and less active those devices are, the fewer messages the Edge Controller must process, and the better it performs.

One common performance factor that the Edge Controller can control, however, is the amount of alerts it generates for each message that it processes. The generation of these alerts is moderated by the alert agent based on the value of the `Alert Threshold` parameter. This value is defined in the Edge Controller details on the Premises Server Administrative console, and can have a major impact on the overall performance of the Edge Controller. Refer to 7.3.4, “Controllers” on page 162 for details on assigning this value.

Table 9-2 shows the amount of alerts that are generated for each of the basic events necessary to read a single tag. The average size of the alerts is also included, and the results are presented for each of the various possible Alert Threshold values.

Table 9-2 Alerts generated for the basic events necessary to read a single tag

Alert Threshold	messages per switch	messages per motion	messages per tag read	Average Msg Size (bytes)
error	1	1	1	630
warning	1	1	1	630
info	2	2	2	685
debug	4	22	12	715

Clearly, the greatest Edge Controller performance can be expected with the alert threshold set to either error or warning, and while the debug level can be useful for testing, it can degrade Edge Controller performance seriously.

Attention: Under no circumstances should the Alert Threshold be set to debug in a production environment.

9.4 RFID Reader performance

The performance of an RFID reader depends greatly on a number of factors.

- ▶ Reader API
- ▶ Reader type
- ▶ Antenna construction
- ▶ Tag types
- ▶ Environment in which the reader is being used

While a number of combinations of these factor might produce suitable results, The perfect combination for your particular RFID infrastructure can be achieved through testing, such as is provided by the IBM Wireless Center of Excellence located in Research Triangle Park, North Carolina.

Archived

Monitoring

This chapter provides information about how to monitor the various domains in your IBM WebSphere RFID solution. It describes the various monitoring methods that are available and contains detailed instructions concerning their use.

We discuss the following areas that are related to monitoring:

- ▶ Premises Server monitoring with Tivoli Enterprise Console
- ▶ Edge Controller monitoring with Tivoli Enterprise Console
- ▶ Monitoring other components of your IBM WebSphere RFID solution

10.1 Tivoli Enterprise Console

Tivoli Enterprise Console is an event management application that collects and processes events from a multitude of sources. Ultimately, Tivoli Enterprise Console allows you to monitor the status of virtually any application. This functionality makes Tivoli Enterprise Console the perfect monitoring tool for your RFID environment.

With the installation of the proper logfile adapters that are provided with the IBM WebSphere RFID solution, Tivoli Enterprise Console displays the status of the Premises, Edge, and any corresponding RFID devices.

Verify the following prerequisites before continuing with the installation:

- ▶ A Tivoli Server with Tivoli Enterprise Console is installed and running.
- ▶ Tivoli Endpoint software is installed on each Premises Server.

Refer to the online help or to the book on Tivoli Enterprise Console for any additional information.

10.1.1 Installing the Logfile Adapters

This section describes how to install the proper logfile adapters for your RFID monitoring needs. You must load the adapters into Tivoli Enterprise Console and then distribute them to the Premises Server. The adapters then run as services on the Premises Server, allowing alerts, events, and exceptions to be viewed from Tivoli Enterprise Console.

WebSphere Application Server Logfile Adapters

The status of the Premises Server is determined by the health of the WebSphere Application Server instance that is running on that machine, because it hosts the Premises Server applications. Follow these instructions to install the WebSphere Application Server Logfile Adapters on one or more Premises servers using Tivoli Enterprise Console:

1. Copy the contents of CD11 to the C: drive of the Tivoli Server.
2. Edit the file, C:\IBM\RFID\monitoring\wasjava.conf:
 - a. Set the ServerLocation variable equal to the Tivoli Enterprise Console Server Host Name.
 - b. Adjust the value of the PollInterval attribute to the rate at which you want Tivoli Enterprise Console to receive and display new events from the Premises Server.
 - c. Modify the value of the BufEvtPath attribute if the file named is already in use by another adapter.

- d. Set the LogSources variable equal to the WebSphere Application Server log file(s) you want Tivoli Enterprise Console to monitor.
3. Create the Profile Manager
 - a. Open the Tivoli Desktop.
 - b. Select an existing policy region, or create a policy region, to contain the profile manager for logfile monitoring.
 - c. Add an Adaptor Configuration Profile (ACP) to the selected region as a managed resource type.
 - d. Add Profile Manager to the selected region as a managed resource type.
 - e. Open the policy region and create a new Profile Manager.
4. Create and Configure the Profile
 - a. Open the newly created Profile Manager and create a new ACP profile
 - b. Open the new ACP profile for editing and add a tecad_win entry.
 - c. Click **General** for the new tecad_win entry and click **Identifier**.
 - d. Enter a descriptive name in the Identifier Name field.
 - e. Click **Distribution** and double-click the C/tecad_win.fmt entry. You are now able to edit the entry.
 - f. Edit the value to reflect the location of the supplied wasjava.fmt file, and click the check mark button to save the changes.
 - g. Enter tecad_win.cds as a property name, enter the path of the supplied wasjava.cds file, and click the check mark button to add the property.
 - h. Enter tecad_win.conf as a property name, enter the path of the supplied wasjava.conf file, and click the check mark button to add the property.
 - i. Click **Save and Close** to save the entry.
5. Set the subscribers for the profile manager to include the Premises Server from which you want to monitor the WebSphere Application Server log file.
6. Import the supplied wasjava.baroc file.
7. Compile the Rule Base and load it into the Event Server.
8. Distribute the ACP profile to the Premises Server.

After distribution, a new service should exist on the Premises Server. It should have an ID equal to the Identifier Name that is given to the ACP entry that was defined above. This service monitors the WebSphere Application Server logfile entered into the LogSources variable value field in the wasjava.conf file. Exceptions logged to this log file are converted by the logfile adapter into an instance of the appropriate class and sent to the Tivoli Enterprise Console server.

Edge Logfile Adapters

The status of Edge Controllers and their corresponding devices is derived from the edge-heartbeats.log file that resides on the Premises Server.

Follow these instructions to install the Edge Logfile Adapters on Premises Server using the Tivoli Enterprise Console:

1. Copy the contents of CD11 to the C: drive of the Tivoli Server if you have not done so already.
2. Edit the file, C:\IBM\RFID\monitoring\tecad_win.conf:
 - a. Set the ServerLocation variable equal to the Tivoli Enterprise Console Server Host Name.
 - b. Adjust the value of the PollInterval attribute to the rate at which you want Tivoli Enterprise Console to receive and display new events from the Premises Server.
 - c. Modify the value of the BufEvtPath attribute if the file named is already in use by another adapter.
 - d. Set the LogSources variable equal to the edge-heartbeats.log file you want Tivoli Enterprise Console to monitor.
3. Create the Profile Manager
 - a. Open the Tivoli Desktop.
 - b. Select an existing policy region or create a policy region to contain the profile manager for logfile monitoring.
 - c. Add ACP to the selected region as a managed resource type.
 - d. Add Profile Manager to the selected region as a managed resource type.
 - e. Open the policy region and create a new Profile Manager.
4. Create and Configure the Profile
 - a. Open the newly created Profile Manager and create a new ACP profile
 - b. Open the new ACP profile for editing and add a tecad_win entry.
 - c. Click **General** for the new tecad_win entry and click **Identifier**.
 - d. Enter a descriptive name in the Identifier Name field.
 - e. Click **Distribution** and double-click the C/tecad_win.fmt entry. You are now able to edit the entry.
 - f. Edit the value to reflect the location of the supplied tecad_win.fmt file, and click the check mark button to save the changes.
 - g. Enter tecad_win.cds as a property name, enter the path of the supplied tecad_win.cds file, and click the check mark button to add the property.

- h. Enter `tecad_win.conf` as a property name, enter the path of the supplied `tecad_win.conf` file, and click the check mark button to add the property.
 - i. Click **Save and Close** to save the entry.
5. Set the subscribers for the profile manager to include the Premises Server from which you want to monitor the `edge-heartbeats.log` file.
6. Import the supplied `premises.baroc` file.
7. Compile the Rule Base and load it into the Event Server.
8. Distribute the ACP profile to the Premises Server.

After distribution, a new service should exist on the Premises Server. It should have an ID equal to the Identifier Name that is given to the ACP entry that was defined above.

This service monitors the `edge-heartbeats.log` file entered into the `LogSources` variable value field in the `tecad_win.conf` file. New heartbeats that are logged to this file are converted by the logfile adapter into an appropriate class and sent to the Tivoli Enterprise Console server.

10.1.2 Monitoring the RFID environment using Tivoli Enterprise Console

Monitoring your RFID environment using Tivoli Enterprise Console is quite simple. After the Logfile Adapters are installed, Tivoli Enterprise Console receives all of the events that those adapters collect from the Premises Server log files. These events are color coded based on severity and can be filtered for your viewing convenience.

10.2 Monitoring Other components

Due to the dependence of any system on the proper operation of all its working parts, it is very useful to monitor other key components of an RFID solution, in addition to those in the RFID specific domains. You can refer to the documentation that is included with the software that is used to install the following components for information about how to monitor those components:

- ▶ Web Infrastructure
- ▶ DB2
- ▶ MQ

Archived



Edge Controller Software installation and management

This chapter describes how to install and manage the Edge Controller software through the use of WebSphere Everyplace Device Manager V5.0. This is the only method of installing the Edge Controller software that is supported by the IBM WebSphere RFID solution.

We discuss the following topics:

- ▶ Installing WebSphere Everyplace Device Manager V5.0
- ▶ Using WebSphere Everyplace Device Manager V5.0
- ▶ Deploying and managing the Edge Controller software
- ▶ Enrolling and configuring the Edge Controller
- ▶ Verifying the Edge Controller setup

11.1 Installing WebSphere Everyplace Device Manager

WebSphere Everyplace Device Manager facilitates the management of a wide array of wireless and mobile devices, and it will allow you to easily reconfigure and deploy software on the Edge Controller devices that are used in your IBM WebSphere RFID solution. It is necessary to use WebSphere Everyplace Device Manager because it is the only supported way to install and to manage the Edge Controller software.

This section discusses the installation and configuration of WebSphere Everyplace Device Manager for use with the IBM WebSphere RFID solution.

11.1.1 WebSphere Everyplace Device Manager prerequisites

The following operating systems are supported:

- ▶ AIX
- ▶ Windows 2000 Server
- ▶ Solaris™

The following minimum hardware configuration is required:

- ▶ RAM: 512 MB
- ▶ Processor: 600 MHZ
- ▶ Free disk space: 1 GB

The following software is required:

- ▶ WebSphere Application Server V5.0.2.8
- ▶ One of the following database servers:
 - DB2 Universal Database V8.1.2 or V7.2
 - Oracle 9i or Oracle 8i

For detailed hardware and software requirements about WebSphere Everyplace Device Manager V5.0, refer to the Information Center that is provided with the product.

Pre-installation checklist

Before proceeding with installation, ensure the following:

- ▶ The network properties contain the fully qualified name of your machine.

Right-click **My Computer** and select **Properties** → **Network Identification** → **Properties** and verify that the full computer name is displayed. If it is not, you might need to set the primary DNS suffix.

To do this, click **More...** and enter the primary DNS suffix in the field. Click **OK** and the full computer name should now be properly displayed.

- ▶ Any Microsoft IIS service is disabled on the machine.

From the Windows desktop, select **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**. If a Microsoft IIS service exists, it is necessary to disable it, due to an incompatibility with WebSphere.

To do this, right-click the service, select **Properties**, and change the Startup type to Disabled.

- ▶ Any existing VMware service on the machine is disabled.
- ▶ The language settings of the machine are set to English.

From the Desktop, select **Start** → **Settings** → **Control Panel** → **Regional** → **Options** → **General**. Under **Your locale (location)**, select English (US).

- ▶ WebSphere Application Server is started.
- ▶ Java is installed and the JAVA_HOME environment variable is set to the Java installation directory. For instance, you can use the WebSphere Application Server Java home directory.

Right-click **My Computer** and select **Properties** → **Advanced** → **Environment Variables**. If the JAVA_HOME variable does not exist under System variables, click **New...** and create it, as shown in Figure 11-1. Click **OK**.

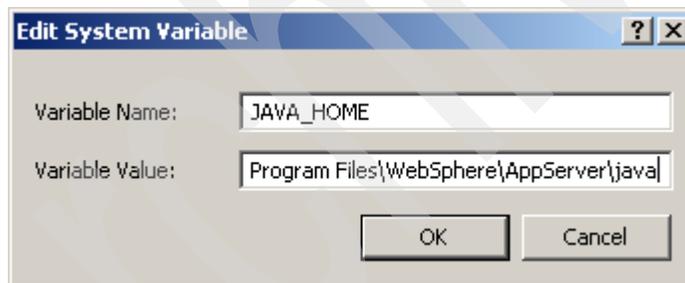


Figure 11-1 Setting the JAVA_HOME environment variable

- ▶ You have your WebSphere Everyplace Device Manager V5.0 CD.

Important: WebSphere Everyplace Device Manager V5.0 is not included with IBM WebSphere RFID solution and must be obtained separately.

- ▶ You have production bundles which you plan to deploy to Edge Controllers.

Note: You should obtain these bundles from your Systems Integrator or Edge OEM. Contact your IBM representative for more information.

- ▶ You have the required WebSphere Everyplace Device Manager patch and OSGi plug-in, located on CD 12 of the IBM WebSphere RFID product CD set.
- ▶ You have the required XML configuration and JAR files, located in the IBM\RFID\premises\WEDMJobUtility directory on CD11 of the IBM WebSphere RFID product CD set.
- ▶ The Premises Server is properly installed, configured, and running. Refer to Chapter 6.1, “Installing the Premises Server software” on page 112 for more information.
- ▶ A solution topology is defined using the Premises Server Administrative Console. Refer to Chapter 7, “Administering the WebSphere RFID solution” on page 147 for more information.

11.1.2 Installing WebSphere Everyplace Device Manager

Follow these instructions to install and to configure WebSphere Everyplace Device Manager for use with your IBM WebSphere RFID solution:

1. Create a new administrator user, hereby referred to as *dmsadmin*, by following these steps:
 - a. Right-click **My Computer** and select **Manage** → **Local Users** → **Groups**.
 - b. Right-click **Users** and select **New User**.
 - c. Set the following parameters in the New User input box:
 - User name: *dmsadmin*
 - Password: *dmspassword*
 - d. Select **Password never expires** and click **Create**.
 - e. Right-click the *dmsadmin* user and select **Properties** → **Member Of**.
 - f. Click **Add...** → **Administrators** → **OK**.
2. Insert the WebSphere Everyplace Device Manager V5.0 installation CD and launch the install file, *install.bat*. Then, follow these steps:
 - a. The Device Manager V1.5 Installer welcome page displays. Click **Next**.
 - b. Change the install directory if necessary, and click **Next**.
 - c. Ensure that both the database and server will be installed. Click **Next**.
 - d. Set IBM HTTP Server home, select your database type, and Click **Next**.

- e. Set the WebSphere Everyplace Device Manager database home, select the proper JDBC driver for your database type, and set the user name and password, using the *dmsadmin* user name and *dmspassword* as shown in Figure 11-2. Click **Next**.

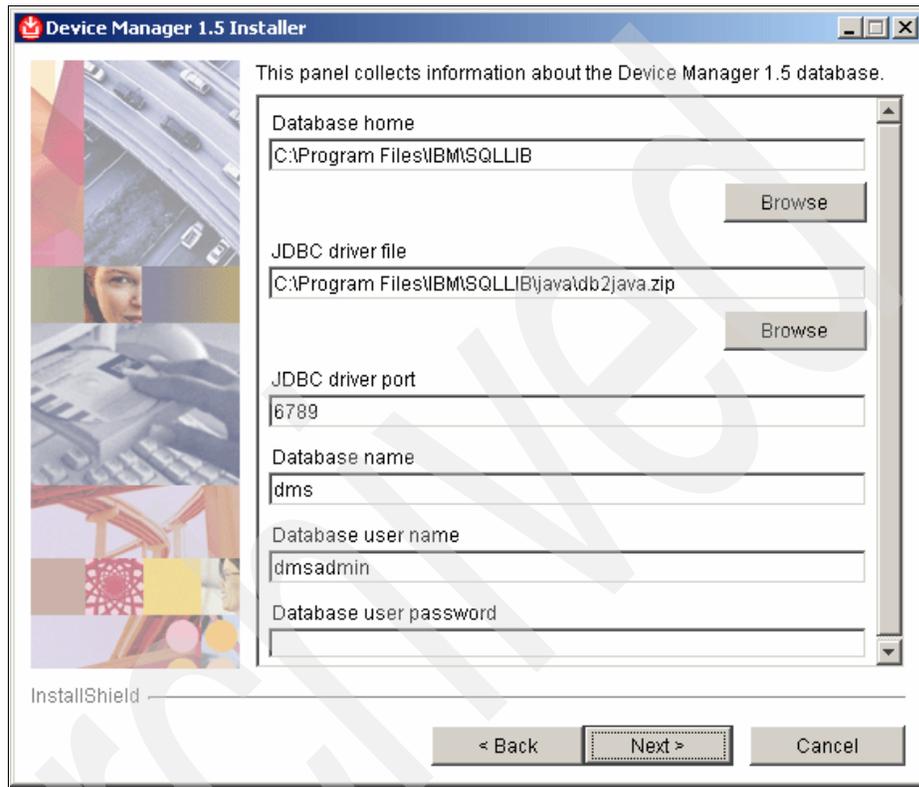


Figure 11-2 Configuring the WebSphere Everyplace Device Manager Database

Note: It is recommended to use the same user name and password as you did when creating the *dmsadmin* user for the server.

- f. The installation summary displays. Click **Next**.

Note: Two portions of the installation (at 2% and 38%) take some time to complete. The total WebSphere Everyplace Device Manager installation should take about ten minutes.

- g. When the installation completes, click **Finish**.

3. Install the WebSphere Everyplace Device Manager V5.0 Fix Pack 1.
 - a. Copy the `Tivoli_Device_Manager_Patch.zip` file from the `fp1` directory on RFID CD12 to a directory on the WebSphere Everyplace Device Manager server and unzip the file.
 - b. From a command prompt, `cd` to the directory where you unzipped the file, and run the following command:

```
installpatch.bat "C:\Program Files\TivDMS15"
```

The patch is installed and confirmed.
 - c. Reinstall the WebSphere Everyplace Device Manager console, on any machines on which it was previously installed, by accessing:

```
http://Device_Manager_Server_IP/dmsserver/DMconsole
```

Note: You do *not* need to complete this step if you are using the console from the WebSphere Everyplace Device Manager server. If this is the case, you can run the console from the `DMConsole.bat` file that is located in the `TivDMS_Install/console` directory.

4. Install the Simple OSGi Plug-in Component.
 - a. Copy the OSGi plug-in file, `SimpleOSGiPluginComponent.jar`, from the plug-in directory on IBM WebSphere RFID CD12 to a local directory
 - b. From a command prompt, `cd` to `c:\program files\tivdm15\bin` and run the following command:

```
compinstall -file c:\SimpleOSGiPluginComponent.jar
```
5. Configure WebSphere Application Server security.
 - a. Open the WebSphere Application Server Administrative Console with **Start** → **Programs** → **IBM WebSphere** → **Application Server v5.0** → **Administrative Console**.
 - b. In the left frame, click **Security** → **User Registries** → **Local OS**.

Note: You can alternatively select any other user registry that is supported by WebSphere Application Server security.

- c. Fill in the Server User ID and Password using `dmsadmin` and `dmspassword`.
- d. Click **OK**, and click the Global Security link in the left frame.

- e. In Global Security, select **Enabled** and deselect **Enforce Java 2 Security**, as shown in Figure 11-3, and click **OK**.

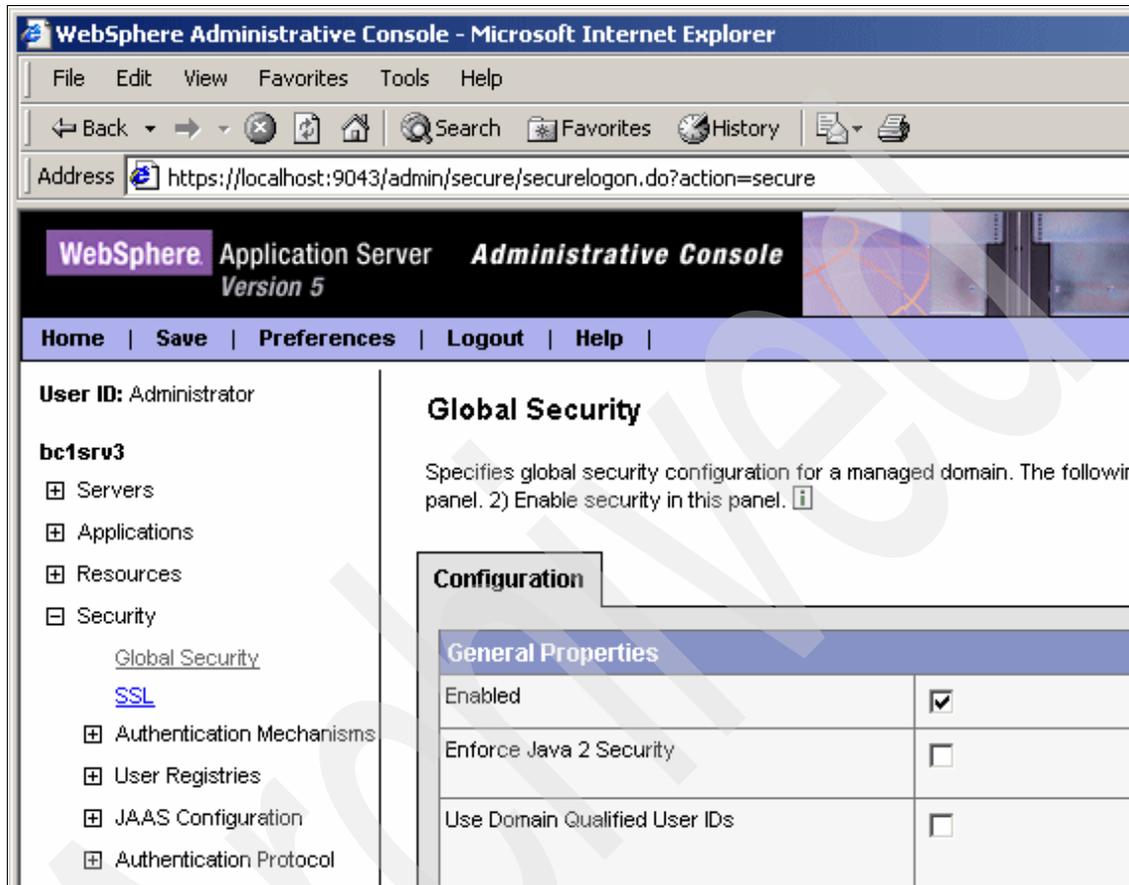


Figure 11-3 Setting Global Security

- f. Click **Save** and then click **Save** again.
6. Add security parameters to server start and stop shortcuts.

Note: This step is necessary only if you want to stop and start the WebSphere Application Server manually.

- a. Click **Start** → **Programs** → **IBM WebSphere** → **Application Server V5.0**. Right-click **Stop The Server** and select **Properties**.
- b. Under the shortcut tab, add the following text to the Target field.
- ```
-user dmsadmin -password dmspassword
```

- c. Repeat steps a and b for the **Start The Server** shortcut.
7. Create WebSphere Everyplace Device Manager Server start and stop scripts.

**Note:** This step is necessary only if you want to stop and start the DMS\_AppServer manually.

- a. Create a startDMS.bat file, as shown in Example 11-1, to launch the DMS\_AppServer server.

*Example 11-1 startDMS.bat*

---

```
cd "C:\Program Files\WebSphere\AppServer\bin"
startServer DMS_AppServer -user dmsadmin -password dmsspassword
```

---

- b. Create a stopDMS.bat file, as shown in Example 11-2, to stop the DMS\_AppServer server.

*Example 11-2 stopDMS.bat*

---

```
cd "C:\Program Files\WebSphere\AppServer\bin"
startServer DMS_AppServer -user dmsadmin -password dmsspassword
```

---

- c. Again, user and password are the Server User ID and Password specified previously for the WebSphere Application Server security settings.
8. Ensure that the following services are set to start automatically:
  - All necessary DB2 services
  - IBM HTTP Server
  - IBM WebSphere Application Server V5 - server1
  - IBM WebSphere Application Server V5 - DMS\_AppServer
9. Reboot the machine.
10. If you want to restart WebSphere Application Server and the DMS\_AppServer at any other time, do so in the following order
  - a. **Start** → **Programs** → **IBM WebSphere** → **Application Server V5.0** → **Stop The server.**
  - b. Execute stopDMS.bat.
  - c. Execute startDMS.bat.
  - d. **Start** → **Programs** → **IBM WebSphere** → **Application Server V5.0** → **Start The server.**

WebSphere Everyplace Device Manager is now installed and configured to manage OSGi devices. Continue on to the next sections to prepare WebSphere Everyplace Device Manager for Edge Controller enrollment. You will find

instructions on how to use the Device Manager console and how to incorporate Edge Controllers into your IBM WebSphere RFID solution.

## 11.2 Using WebSphere Everyplace Device Manager

After WebSphere Everyplace Device Manager 5.0 is installed and configured properly, you can use it to manage devices and distribute software. However, devices will first need to enroll with the WebSphere Everyplace Device Manager server. This section shows you how to do the following:

Enroll your Edge Controllers with WebSphere Everyplace Device Manager  
Create a distributable software package from your production bundles  
Update Edge Controller parameters and distribute software

Figure 11-4 illustrates the enrollment and software distribution process, which should clarify how WebSphere Everyplace Device Manager fits into the WebSphere RFID solution. You can then follow the remainder of this section to get your solution up and running.

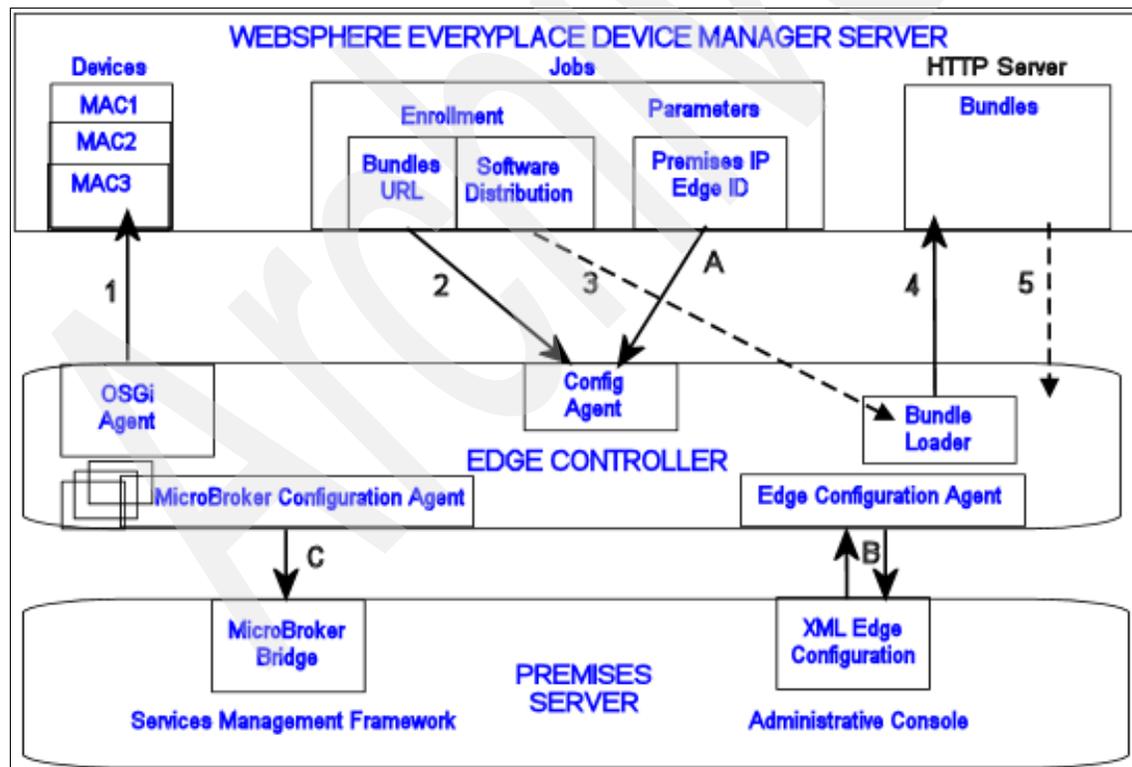


Figure 11-4 WebSphere Everyplace Device Manager and the RFID solution

## 11.2.1 The Device Manager console

The Device Manager console is used to manage devices. It displays and allows access to all of the current information pertinent to the WebSphere Everyplace Device Manager server, including the following:

- ▶ Job status
- ▶ Device properties
- ▶ Available software

You can access the console from the **DMConsole.bat** command located in the *TivDMS\_Install\console* directory on the WebSphere Everyplace Device Manager server. For example:

```
C:\Program Files\TivDMS15\console\DMconsole.bat
```

If you are using the console from a remote machine, this file is located in the directory to which you installed the Device Manager console. Login to the Device Manager console, using `dmadmin/dmadmin` as the User ID/Password combination, if these default credentials were not modified. You are also prompted for the Device Manager server and can simply enter `localhost` if you are accessing the console directly from the Device Manager server.

After logging into the Device Manager console you can click the Device Classes (Figure 11-5) link in the left frame, to verify that the file `SimpleOSGiPluginComponent.jar` installed earlier has in fact added OSGi to the Device Classes list.

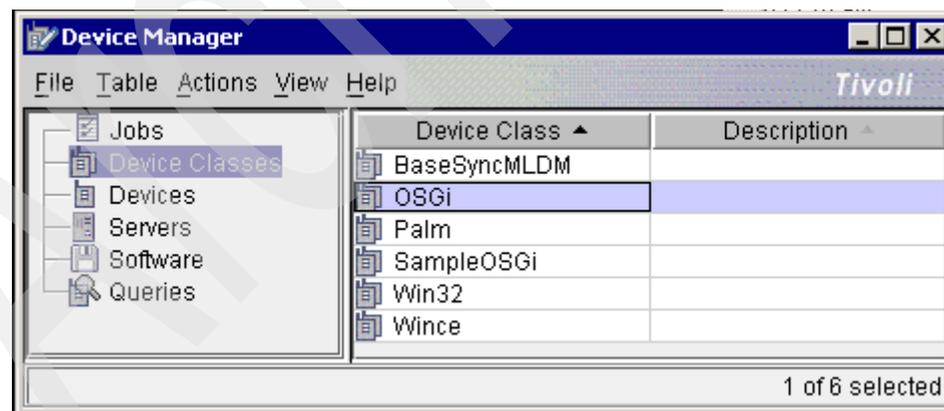


Figure 11-5 Verifying the OSGi Device Class in the Device Manager Console

You can click the links in the left frame to access the corresponding view. When using the Device Manager console to manage Edge Controllers, we are primarily concerned with the following views:

- ▶ Jobs

The Jobs view contains a list of all jobs that have been submitted to the Device Manager Server along with their status and type. You can access more detailed information about a particular job, including its history, progress, and target devices, by right-clicking it.

- ▶ Devices

The Devices view contains a list of all devices that have enrolled with the Device Manager server along with their MAC Address and Device Class. You can access more detailed information about a particular device, including the its specific job progress, by right-clicking it.

- ▶ Software

The Software view contains a list of all software that is available for deployment along with its version and type. You can access more detailed information about a software package, including its location and applicable jobs, by right-clicking it.

## 11.2.2 Preparing the WebSphere Everyplace Device Manager Server

In order for the Edge Controllers to enroll and receive software distributions properly, some preparations must be made on the WebSphere Everyplace Device Manager server. Follow these instructions to prepare the server:

1. Copy the contents of CD11 of the IBM WebSphere RFID solution to a local directory, referred to as *RFID\_INSTALL\_DIR*.
2. Copy your production bundles from the provided CD or download to the HTTP server on the WebSphere Everyplace Device Manager server, preferably in the following directory:

`HTTP_HOME\htdocs\en_US\bundles`

**Note:** The production bundles are provided as a set of JAR files, by your Systems Integrator. Without these bundles, you cannot create and distribute Edge Controller Software. Contact your IBM representative or Edge Controller OEM for more information about these bundles.

3. Create the software you will deploy on your Edge Controllers.
  - a. Open createWedmSoftware.xml for editing (Example 11-3). It is located in the following directory:

*RFID\_INSTALL\_DIR\IBM\RFID\premises\WEDMJobUtility\xml*

*Example 11-3 createWedmSoftware.xml*

---

```
<?xml version="1.0" encoding="UTF-8"?>

<dms-task>
 <server uid="<USERID>" passwd="<PASSWORD>">
 <url value="http://<HOSTNAME>/dmsserver/servlet/rpcrouter"/>
 </server>

 <!--CREATE SOFTWARE-->
 <software action="install" type="OSGiBundle">
 <url value="http://<HOSTNAME>/<DIRECTORY>/KIMONO_<READER
TYPE>_LOADER+<VERSION_NUMBER>.jar"/>
 </software>
</dms-task>
```

---

- b. Modify the variables shown in Table 11-1.

*Table 11-1 Variables*

Variable	Value
USERID	dmsadmin as defined in 11.1.2, "Installing WebSphere Everyplace Device Manager" on page 222
PASSWORD	<i>dmspassword</i> as defined in 11.1.2, "Installing WebSphere Everyplace Device Manager" on page 222
HOSTNAME	The Device Manager server host name or IP address
DIRECTORY	The directory in which the production bundles are stored, relative to the HTTP server's document root
READER TYPE <sup>1</sup>	The type of reader used in your production environment
VERSION_NUMBER <sup>2</sup>	The software version number (for example 1_0_28); this value can be taken from the end of the file name of the .jar that is used to create the software

**Note 1:** Currently, `READER TYPE` can only be one of the following fully supported reader type values: `ALIEN`, `INTERMEC`, `MATRICES`, or `SAMSYS`. Software for unsupported and *as is* reader types cannot be distributed from WebSphere Everyplace Device Manager unless customized production bundles have been provided.

**Note 2:** It is safer to use the name of the JAR file that will be used to create the software in place of `KIMONO_READER TYPE_LOADER+VERSION_NUMBER.jar`. This is a necessary step if you are using customized production bundles that do not follow the naming convention that is used in this XML file.

- c. Save and close the XML file.
- d. Open a command prompt and `cd` to the following directory:  
`RFID_INSTALL_DIR\IBM\RFID\premises\WEDMJobUtility`
- e. Execute the following command:  
`xmlConfig.bat xml\createWedmSoftware.xml`

The Edge Controller software can now be deployed from the WebSphere Everyplace Device Manager server. You can verify the creation of this software package by accessing the Software view on the Device Manager console.

4. Create an enrollment job that will be executed on every new device.
  - a. Open the file, `createWedmJobs.xml` for editing (Example 11-4). It is located in the following directory:

`RFID_INSTALL_DIR\IBM\RFID\premises\WEDMJobUtility\xml`

*Example 11-4 createWedmJobs.xml*

---

```
<?xml version="1.0" encoding="UTF-8"?>

<dms-task>
 <server uid="<USERID>" passwd="<PASSWORD>">
 <url value="http://<HOSTNAME>/dmsserver/servlet/rpcrouter"/>
 </server>

 <!--CREATE JOBS-->
 <job action="replace" type="SYNCLDM_CMD" deviceClass="OSGi" targetScope="BOTH"
priority="3">
 <param name="REPLACE_ITEM_1_TARGET_URI"
value="./OSGi/BundleConfiguration/com.ibm.kimono.EdgeConfigAdmin/wedm.bundle.urlprefix"/>
 <param name="REPLACE_ITEM_1_DATA" value="http://<HOSTNAME>/<DIRECTORY>/"/>
 </job>
</dms-task>
```

```

</job>

<job action="install" bundleName="KIMONO_<READER TYPE>_LOADER" type="SW_DIST"
deviceClass="OSGi" targetScope="BOTH" priority="3">
 <param name="AutoStart" value="TRUE"/>
</job>

</dms-task>

```

b. Modify the variables shown in Table 11-2.

Table 11-2 Variables

Variable	Value
USERID	<i>dmsadmin</i> as defined in 11.1, "Installing WebSphere Everyplace Device Manager" on page 220
PASSWORD	<i>dmspassword</i> as defined in 11.1, "Installing WebSphere Everyplace Device Manager" on page 220
HOSTNAME	The Device Manager server host name or IP address
DIRECTORY	The directory in which the production bundles are stored, relative to the HTTP server's document root
READER TYPE <sup>1</sup>	The type of reader used in your production environment

**Note 1:** Currently, READER TYPE can only be one of the following fully supported reader type values: ALIEN, INTERMEC, MATRICS, or SAMSYS. Software for unsupported and *as is* reader types cannot be distributed from WebSphere Everyplace Device Manager unless customized production bundles have been provided.

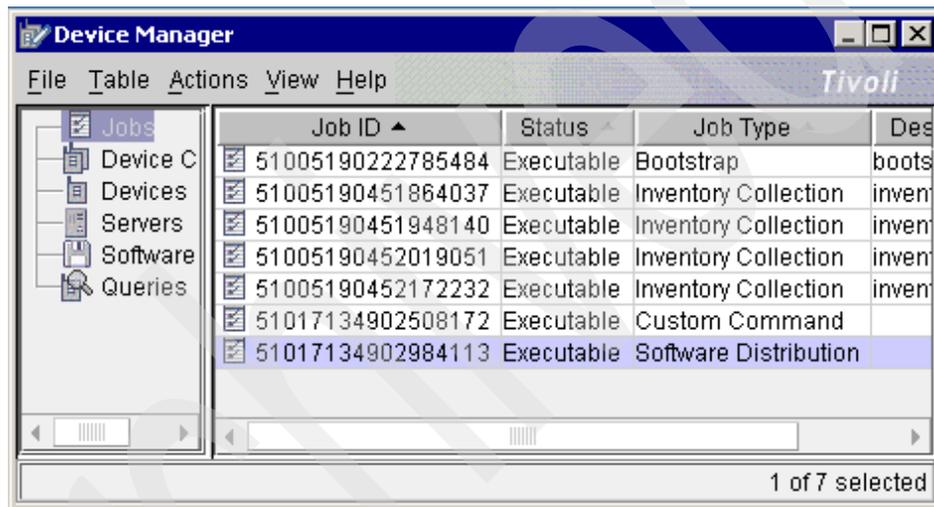
- c. Save and close the XML file.
- d. At the command, execute the following command:

```
xmlConfig.bat xml\createWedmJobs.xml
```

This creates two enrollment jobs. The first, a custom command, tells each device where its software is located. This information is then used by the Edge Controller when it loads the software distributed by the second job.

**Attention:** Because these are enrollment jobs, they are executed on each and every new device as it enrolls with the WebSphere Everyplace Device Manager server. Thus, all Edge Controllers use the same bundles directory and receive the same software. If you want to distribute different software to specific Edge Controllers, refer to the WebSphere Everyplace Device Manager Information Center for information about defining groups.

Check the Device Manager Jobs view to ensure that the Custom Command and Software Distribution jobs were created and executed (Figure 11-6). If there is any doubt, you can match the Job IDs from your console output.



Job ID	Status	Job Type	Description
51005190222785484	Executable	Bootstrap	boots
51005190451864037	Executable	Inventory Collection	invent
51005190451948140	Executable	Inventory Collection	invent
51005190452019051	Executable	Inventory Collection	invent
51005190452172232	Executable	Inventory Collection	invent
51017134902508172	Executable	Custom Command	
51017134902984113	Executable	Software Distribution	

Figure 11-6 Confirming job creation

## 11.3 Enrolling and configuring the Edge Controller

After WebSphere Everyplace Device Manager is installed, configured, and prepared to manage the Edge Controllers in your IBM WebSphere RFID solution properly, you can fully incorporate the Edge Controllers into the solution. This section shows you how to configure your Edge Controllers to enroll with WebSphere Everyplace Device Manager and connect to the Premises Server.

### 11.3.1 Configuring the OSGi Agent on the Edge Controller

In order for your Edge Controller to contact and to enroll with the WebSphere Everyplace Device Manager server, you need to configure the OSGi Agent as follows.

1. Login to your Edge Controller as root and `cd` to the `/smf` directory.
2. Ensure that SMF is NOT already running. If it is, issue a `killall j9`, to stop it.
3. Open `OSGiAgent.properties.bak` for editing. It should appear as shown in Example 11-5.

*Example 11-5 OSGiAgent.properties.bak*

---

```
AccountID = <USERID>
Addr = http://<HOSTNAME>/dmserver/SyncMLDMServletAuthRequired
ServerPW = <PASSWORD>
UserName = <USERID>
ClientPW = <PASSWORD>
DevId = OSGiViper-<MAC ADDRESS>
Mod = OSGi
PollingEnabled = true
PollingStart = 00:00
PollingEnd = 23:59
PollingInterval = 00:01
LogSize = 200
LogThreshold = 3
KeyRing = KeyRing.p12
KeyRingPassword = fred
TempFileLoc = /tmp
```

---

4. Modify the necessary property values, according to Table 11-3.

*Table 11-3 OSGi Agent properties*

Property	Variable	Value
AccountID	USERID	<i>dmsadmin</i>
Addr	HOSTNAME	Device Manager host name or IP address
ServerPW	PASSWORD	<i>dmpassword</i>
UserName	USERID	<i>dmsadmin</i>
ClientPW	PASSWORD	<i>dmpassword</i>

**Attention:** Both sets of *USERID* and *PASSWORD* should be identical. Remember that you set *dmsadmin* and *dmspassword* during the installation of WebSphere Everyplace Device Manager V5.0. Refer to 11.1.2, “Installing WebSphere Everyplace Device Manager” on page 222 for details.

5. Save the text file and exit the editor.
6. From the `smf` directory, execute the following command:

```
./initDMS.sh
```

The OSGi Agent is now configured to connect to your WebSphere Everyplace Device Manager server when SMF is started on the Edge Controller. When this occurs, the Edge Controller will enroll as a device and receive any enrollment jobs from the WebSphere Everyplace Device Manager Server. You can then manage the device through the use of the Device Manager console and XML jobs executed using the `xmlConfig.bat` command.

### 11.3.2 Starting SMF on the Edge Controller

On most Edge Controllers, SMF is set to start automatically to ensure that the Edge Controller can run independently of any outside interaction from the moment it boots up. Starting automatically is very useful in the event of a reboot or power outage, but now that SMF has been stopped manually, it must be restarted.

If SMF was running when you logged in to configure the OSGi Agent the SMF runtime on your Edge Controller is most likely set to start automatically. You can verify this by the existence of the `S99startKimono` script, which is located in the `/etc/rc3.d` directory on the Edge Controller.

If this is the case, then you can simply reboot the Edge Controller, by typing **reboot** at the prompt, and disconnect your session.

If this script does not exist and you want SMF to start automatically, you can create it as shown in Example 11-6, and then reboot the Edge Controller.

**Important:** Ensure the `start.smf` command has execute permission for the user that is running the command. In Linux, you can use the `chmod` command to add the execute permission. Refer to the `chmod` manual page for a description of how to add the execute permission.

*Example 11-6 S99startKimono*

```
#!/bin/bash
/smf/start.smf &
```

After the Edge Controller reboots, SMF starts automatically, the OSGi Agent contact enrolls with the WebSphere Everyplace Device Manager server, and any enrollment jobs are executed.

**Note:** If you do not want SMF to start automatically you can simply type `./start.smf` from inside the `/smf` directory to start SMF manually. Be aware, however, that if you are connected using Telnet or an SSH session, any processes that you have started, will be stopped when you disconnect, including SMF. So, this method is not recommended. If you logged into the Edge Controller with a serial cable, however, you can disconnect it at anytime and the processes will remain.

You can confirm that a device has enrolled on the Devices view of the Device Manager console as shown in Figure 11-7. The Device Name is displayed in the form *Type-MAC* so that of an Edge Controller displays OSGiViper-MAC.

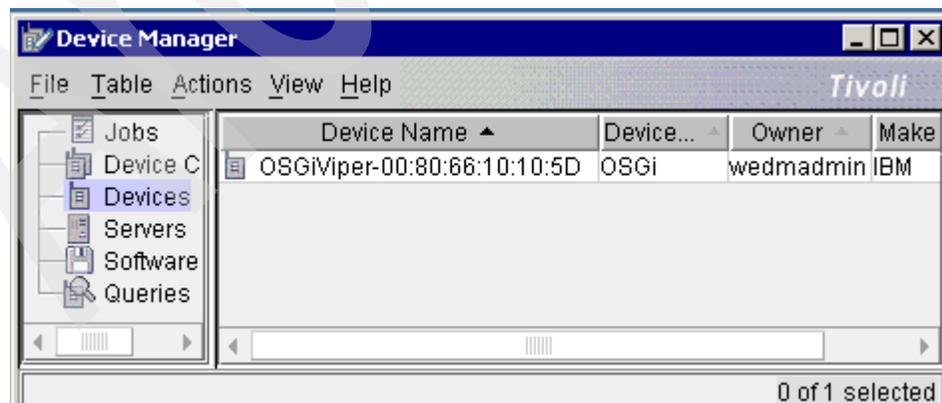


Figure 11-7 Viewing enrolled devices

You can also confirm that the Edge Controller received its enrollment jobs by right-clicking the device, selecting **View Job Progress**, and clicking **OK**.

**Tip:** The Edge Controller should enroll with WebSphere Everyplace Device Manager within two minutes of restarting. This time frame is defined by the `PollingInterval` value in the `OSGiAgent.properties.bak` file in Example 11-5 on page 234. The Edge Controller should also receive the enrollment jobs within a minute of enrolling, if not simultaneously. If this does not occur, you might have a problem either in the OSGi Agent configuration, the WebSphere Application Server security settings, or the creation of the enrollment jobs themselves.

### 11.3.3 Configuring the Edge Controller parameters

After the Edge Controller has enrolled with the Device Manager, it receives its software through the enrollment jobs, but it still needs to set its Edge ID and Premises Server IP address parameters in order to pull the proper configuration from the correct Premises Server. The Edge Controller receives these parameters through a final, device specific, job from the Device Manager.

Follow these steps to define and execute the job, incorporating your Edge Controller in to your IBM WebSphere RFID solution:

1. Create a job to define the Edge ID and Premises IP parameters:
  - a. Open the file, `updateParameters.xml` for editing (Example 11-7). It is located in the following directory:

```
RFID_INSTALL_DIR\IBM\RFID\premises\WEDMJobUtility\xml
```

*Example 11-7 updateParameters.xml*

---

```
<?xml version="1.0" encoding="UTF-8"?>
<dms-task>
 <server uid="<USERID>" passwd="<PASSWORD>">
 <url value="http://<HOSTNAME>/dmsserver/servlet/rpcrouter"/>
 </server>

 <job action="replace" type="SYNCLDM_CMD" deviceClass="OSGi"
deviceName="OSGiViper-<MAC_ADDRESS>" <!--MUST BE EXISTING DEVICE-->
 <param name="1#REPLACE_ITEM_1_TARGET_URI"
value="./OSGi/BundleConfiguration/com.ibm.kimono.EdgeConfigAdmin/edge.id"/>
 <param name="1#REPLACE_ITEM_1_DATA" value="<EDGEID>"/>
 <param name="1#REPLACE_CMD_NUMBER" value="1"/>
 <param name="2#REPLACE_ITEM_1_TARGET_URI"
value="./OSGi/BundleConfiguration/com.ibm.kimono.EdgeConfigAdmin/premises.ip"/>
 <param name="2#REPLACE_ITEM_1_DATA" value="<PREMISES_IP>:<PORT_NUMBER>"/>
 <param name="2#REPLACE_CMD_NUMBER" value="2"/>
 </job>
</dms-task>
```

---

- b. Modify the variables shown in Table 11-2.

Table 11-4 Variables

Variable	Value
USERID	<i>dmsadmin</i> as defined in 11.1, "Installing WebSphere Everyplace Device Manager" on page 220
PASSWORD	<i>dmpassword</i> as defined in 11.1, "Installing WebSphere Everyplace Device Manager" on page 220
HOSTNAME	The Device Manager server host name or IP address
MAC_ADDRESS	The MAC address of the Edge Controller to which these parameters will be assigned. This can be obtained from the second half of the Device Name of the Edge Controller as displayed in the Device Manager Console.
EDGEID	The ID of a Controller defined on the Premises Server Administrative Console. The Edge Controller retrieves the configuration for this Controller when it connects to the Premises Server. See 7.3, "Defining your RFID network topology" on page 150 for details.
PREMISES_IP	The IP address of the Premises Server.
PORT_NUMBER	The port of the Premises Server application. By default, this value is 9080.

- c. Save and close the XML file.
- d. At the command prompt, execute the following command:

```
xmlConfig.bat xml\updateParameters.xml
```

This command creates a job that will be executed only on the Edge Controller specified by the MAC address in the XML. This job updates two parameters, the Edge ID and the Premises IP, which the Edge Controller uses to connect to the Premises Server and obtain its configuration.

This is the last step in setting up an Edge Controller, because when the configuration from the Premises Server is loaded, the Edge Controller can correctly communicate with its devices and the Premises Server. This completes the incorporation of the Edge Controller into the IBM WebSphere RFID solution.

## 11.4 Verifying the Edge Controller setup

After completing all the steps thus far, you should be able to run the Dock Door Receiving Scenario. Following the steps described in this chapter is the easiest way to verify that the Edge Controller has been installed and configured correctly. See Chapter 8, “Running the Dock Door Receiving scenario” on page 181 for detailed information about how to complete this process.

If the Dock Door Receiving Scenario fails, there are a number of verification points that you can check to ensure that there were no errors in the installation and configuration of the Edge Controller:

- ▶ SMF is started on the Edge Controller.

Login to the Edge Controller as root and execute a ps command. If you do not see any J9 processes then SMF is not started. Start SMF manually or ensure that SMF is set to start automatically, and reboot the device.

- ▶ The Edge Controller has enrolled with the Device Manager.

Check the Devices view on the Device Manager Console to verify that the Edge Controller in question is in the list of enrolled devices. If it is not, you should verify your WebSphere Everyplace Device Manager Server security settings and the contents of the /smf/OSGiAgent.properties.bak file on the Edge Controller to ensure that the Edge Controller can contact the WebSphere Everyplace Device Manager Server and properly enroll.

- ▶ Software and enrollment jobs were successfully created and executed.

Example 11-8 shows a sample of the command prompt output from a group of valid software creation and configuration jobs.

### *Example 11-8 Creating jobs with xmlConfig.bat*

---

```
WEDMJobUtility>xmlConfig.bat xml\createWedmSoftware.xml
server element found: uid=wedmadmin passwd=w112work
url=http://bc1srv3.itso.ral.ibm.com/dmservlet/servlet/rpcrouter

software element found: action=install
 created software with swid=51017133501022233

No job nodes to configure

WEDMJobUtility>xmlConfig.bat xml\createWedmJobs.xml
server element found: uid=wedmadmin passwd=w112work
url=http://bc1srv3.itso.ral.ibm.com/dmservlet/servlet/rpcrouter

No software nodes to configure

```

```

job element found: action=replace type=SYNCMLDM_CMD deviceClass=OSGi targetScope=BOTH
priority=3 parameters= {REPLACE_ITEM_1_DATA=http://bc1srv3.itso.ral.ibm.com/bundles/,
REPLACE_ITEM_1_TARGET_URI=./OSGi/BundleConfiguration/com.ibm.kimono.EdgeConfigAdmin/wedm.bundle
.urlprefix}
 custom job created, id = 51017134902508172
job element found: action=install type=SW_DIST deviceClass=OSGi targetScope=BOTH priority=3
parameters= {AutoStart=TRUE}
creating from bundle: KIMONO_INTERMEC_LOADER
swid:51017133501022233
 software distribution job created, id = 51017134902984113,
software=KIMONO_INTERMEC_LOADER

WEDMJobUtility>xmlConfig.bat xml\updateParameters.xml
server element found: uid=wedmadmin passwd=will2work
url=http://bc1srv3.itso.ral.ibm.com/dmsserver/servlet/rpcrouter

No software nodes to configure

job element found: action=replace type=SYNCMLDM_CMD deviceClass=OSGi targetScope= priority=0
parameters= {1#REPLACE_CMD_NUMBER=1, 1#REPLACE_ITEM_1_TARGET_URI=./OSGi/BundleC
onfiguration/com.ibm.kimono.EdgeConfigAdmin/edge.id, 2#REPLACE_ITEM_1_DATA=9.42.171.98:9080,
2#REPLACE_ITEM_1_TARGET_URI=./OSGi/BundleConfiguration/com.ibm.kimono.EdgeConfigAd
min/premises.ip, 1#REPLACE_ITEM_1_DATA=ArcomB, 2#REPLACE_CMD_NUMBER=2}
 custom job created, id = 51017162944332713

```

You can check on the status of these jobs in the Jobs view of the Device Manager console, as shown in Figure 11-8.

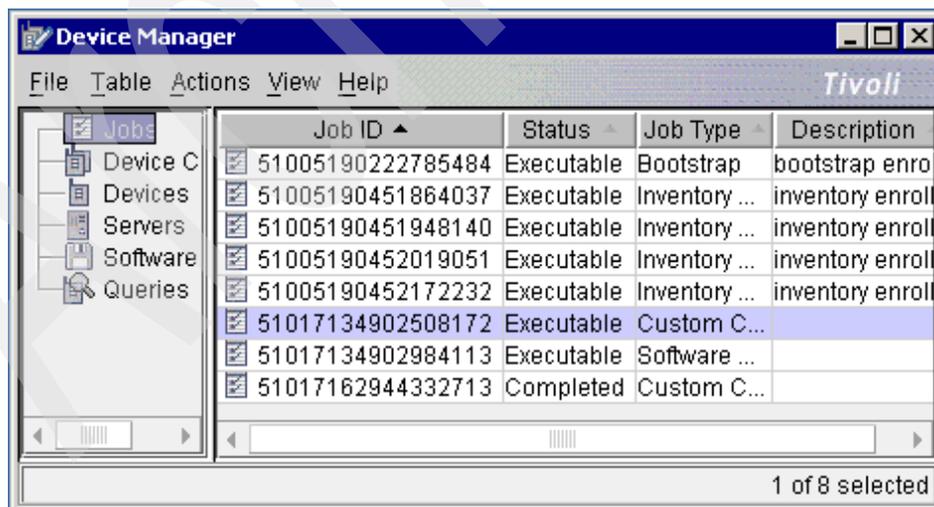


Figure 11-8 Viewing jobs and job status

The first two jobs with no description are your enrollment jobs. They will always have a status of Executable, because they will be run on any device that enrolls. The last job contains the parameters for that specific Edge Controller and should have a status of Completed.

You can right-click any job, select **View Job Progress**, and click **OK** to see if, when, and on which device that job was actually submitted and completed, as shown in Figure 11-9. The value **OK** denotes successful submission of a job.

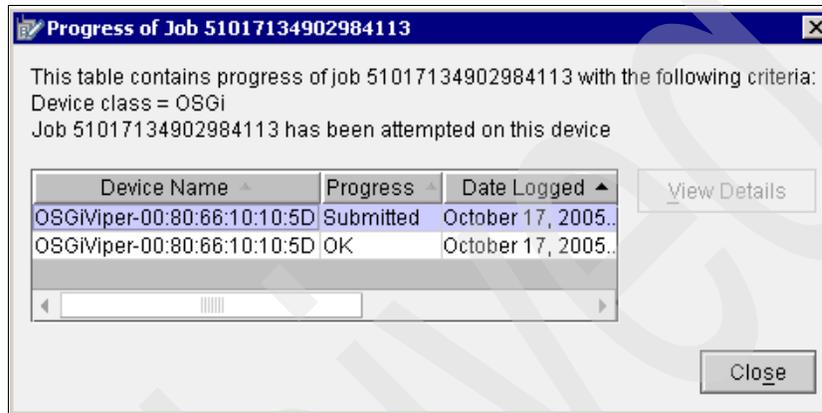


Figure 11-9 Monitoring job progress

You can also do the same for a Device in the Devices view, in order to see the progress of all the jobs submitted to that specific device, as shown in Figure 11-10 on page 241.

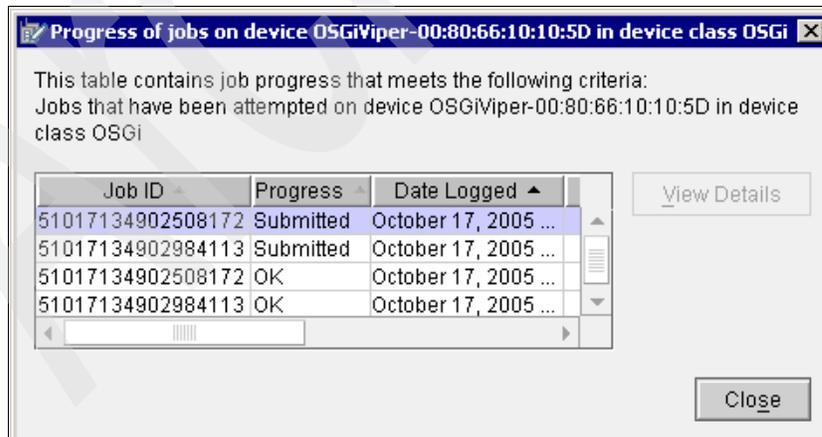


Figure 11-10 Device specific job progress

- ▶ Software has been installed on the Edge Controller.  
If SMF is started on the Edge Controller, but there are only eleven J9 processes running, then it is most likely that no software has been installed. Verify that your Edge Controller is enrolled and that the software distribution job has been executed on the device in question.
- ▶ The Edge Controller has received the correct parameters.  
Check the status of the final Custom Command job. Also verify that the parameters assigned in the XML job defined in Example 11-7 on page 237 match those of the actual solution topology. The *Premises\_IP* must be correct and a Controller with an ID matching that of the *Edge\_ID* must exist in the Premises Server Administrative Console.
- ▶ Verify that the Premises Server is accessible from the Edge Controller.  
Login to the Edge Controller and either ping or Telnet to the Premises Server.
- ▶ The Edge Controller has pulled a valid configuration from the following URL:  
`http://PremisesIP:9080/event_admin_web/premises.s1?action=getconfig&edge=<EdgeID>`



# Part 4

## Appendixes

Archived

Archived

## Supported device matrix

This appendix provides the matrix of the Edge Controllers, RFID readers, and RFID printers that are supported by IBM WebSphere RFID Premises Server V1.0.2 and IBM WebSphere RFID PVS Starter Kit V1.0.2.

To get an up-to-date list of the supported Edge Controller hardware devices and RFID readers for IBM WebSphere RFID Premises Server V1.0.2, visit the following:

[http://www.ibm.com/software/pervasive/ws\\_rfid\\_premises\\_server/technical\\_details](http://www.ibm.com/software/pervasive/ws_rfid_premises_server/technical_details)

Visit the WebSphere RFID Premises Server Support URL to get the up-to-date IBM WebSphere RFID PVS Starter Kit V1.0.2 installation guide, which provides the list of supported RFID readers and RFID printers:

[http://www.ibm.com/software/pervasive/ws\\_rfid\\_premises\\_server/support](http://www.ibm.com/software/pervasive/ws_rfid_premises_server/support)

## A.1 IBM WebSphere RFID V1.0.2 matrix

This section presents the Edge Controllers and RFID readers that are supported by IBM WebSphere RFID Premises Server V1.0.2.

**Note:** The devices that we used during this book are in **bold** font.

Table 11-5 provides the list of supported Edge Controllers.

Table 11-5 IBM WebSphere RFID V1.0.2 supported Edge Controllers

Manufacturer	Model
<b>Arcom</b>	<b>Rugged RFID Edge Controller Industrial Enclosure</b> <a href="http://www.arcom.com/ibm/rfid-edge-controller-IC.htm">http://www.arcom.com/ibm/rfid-edge-controller-IC.htm</a>

Table 11-6 provides the list of supported RFID readers.

Table 11-6 IBM WebSphere RFID V1.0.2 supported RFID readers

Manufacturer	Model
<b>Alien</b>	<b>ALR9780</b>
Asyst	ATR 9100 technology preview
Feig	ISC.LRU1000 technology preview
<b>Intermec</b>	<b>IF5</b>
Intermec	ITRF
Samsys	MO9320 2.8EPC technology preview
Samsys	MP9320 2.7EPC
Symbol	Matrics AR400
Symbol	Matrics RDR-001 technology preview
Symbol	Matrics XR400 technology preview
Tagsys	medio L100 technology preview

## A.2 IBM WebSphere RFID PVS Starter Kit V1.0.2 matrix

This section presents the Edge Controllers and RFID readers that are supported by IBM WebSphere RFID PVS Starter Kit V1.0.2.

**Note:** The devices that we used during this book are in **bold** font.

Table 11-7 provides the list of supported Edge Controllers.

Table 11-7 Starter Kit supported Edge Controllers

Manufacturer	Model
<b>Arcom</b>	<b>Rugged RFID Edge Controller Industrial Enclosure</b> <a href="http://www.arcom.com/ibm/rfid-edge-controller-IC.htm">http://www.arcom.com/ibm/rfid-edge-controller-IC.htm</a>

Table 11-8 provides the list of supported RFID readers.

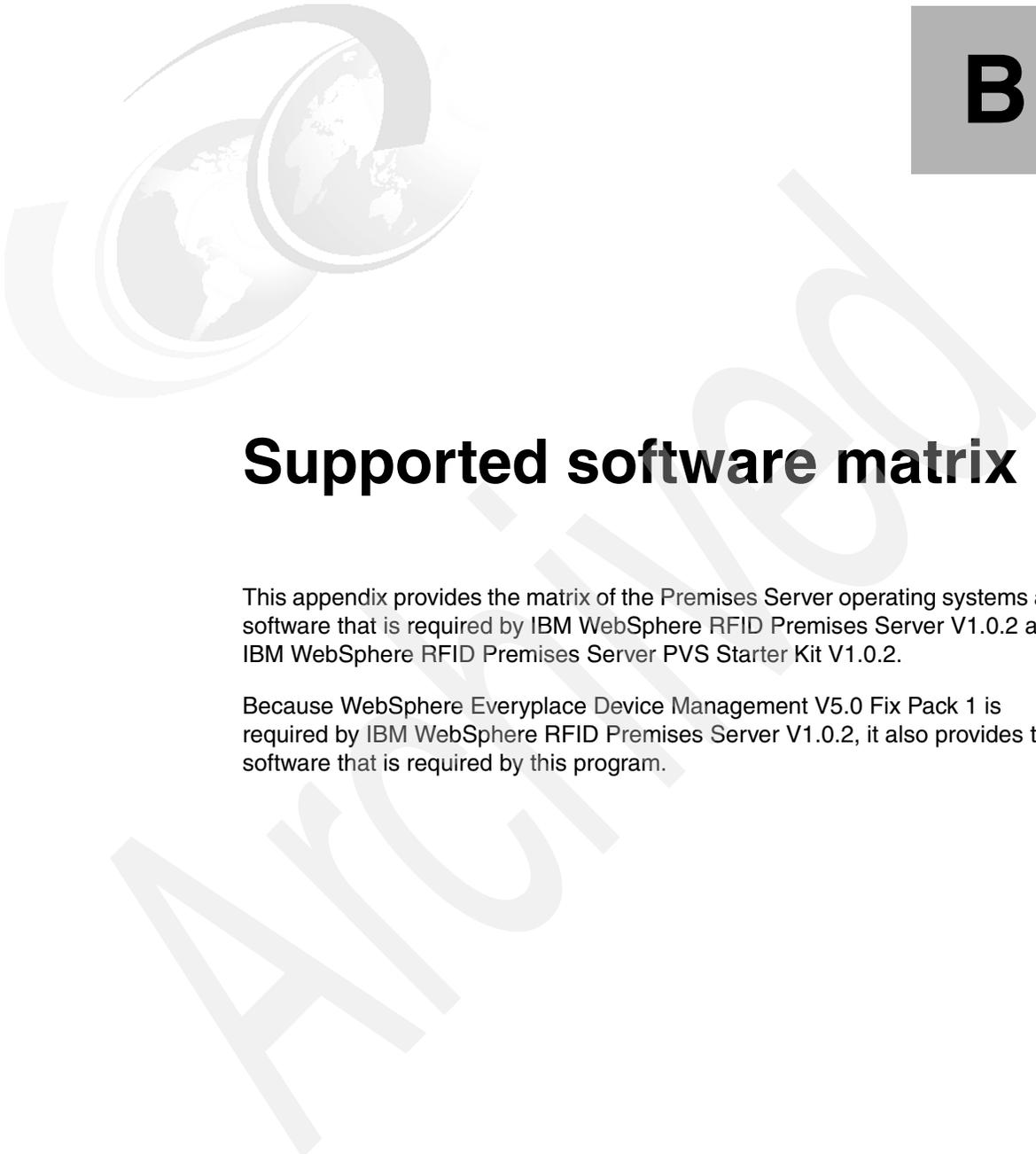
Table 11-8 Starter Kit supported RFID readers

Manufacturer	Model
<b>Alien</b>	<b>ALR9780</b>
Asyst	ATR 9100 technology preview
Feig	ISC.LRU1000 technology preview
Intermec	IF5U
Intermec	ITRF
Samsys	MO9320 2.7EPC
Samsys	MP9320 2.8EPC technology preview
Symbol	Matrics AR400
Symbol	Matrics RDR-001 technology preview
Symbol	Matrics XR400 technology preview
Tagsys	medio L100 technology preview

Table 11-9 provides the list of supported RFID printers.

*Table 11-9 Starter Kit supported RFID printers*

<b>Manufacturer</b>	<b>Model</b>
IBM	6700 RFID printer technology preview
Loftware	Loftware Print Server
<b>Printronix</b>	<b>SL5000e</b>
Printronix	SL5000r
<b>Zebra</b>	<b>R110Xi III Plus</b>
Zebra	R170Xi (adapter only)



## Supported software matrix

This appendix provides the matrix of the Premises Server operating systems and software that is required by IBM WebSphere RFID Premises Server V1.0.2 and IBM WebSphere RFID Premises Server PVS Starter Kit V1.0.2.

Because WebSphere Everyplace Device Management V5.0 Fix Pack 1 is required by IBM WebSphere RFID Premises Server V1.0.2, it also provides the software that is required by this program.

## B.1 IBM WebSphere RFID Premises Server V1.0.2 matrix

This section presents the software programs that are supported by IBM WebSphere RFID Premises Server V1.0.2.

Table B-1 provides the list of software that is required by the Premises Server.

*Table B-1 IBM WebSphere RFID Premises Server V1.0.2 required software*

Software type	Software version
Operating System	One of the following operating systems: <ul style="list-style-type: none"><li>• Windows 2000 Server Service Pack 4</li><li>• Windows Server 2003 Service Pack 1</li></ul>
Database Server	One of the following database servers: <ul style="list-style-type: none"><li>• DB2 Workgroup Server V8.1.4</li><li>• Oracle 9i</li></ul>
Application Server	WebSphere Application Server V5.1 Fix Pack 1
Application Messaging	WebSphere MQ V5.3.0.8 CSD08

**Note:** We used Windows Server 2003 Service Pack 1 and database program DB2 Workgroup Server V8.1.4 for this book.

Table B-2 provides the list of software that is required for WebSphere Everyplace Device Manager V5.0 Fix Pack 1. You can find more information in the WebSphere Everyplace Device Manager Information Center.

*Table B-2 Everyplace Device Manager V5.0 FP 1 required software*

Software type	Software version
Operating System	One of the following operating systems: <ul style="list-style-type: none"><li>• IBM AIX V5.1</li><li>• AIX V5L</li><li>• IBM AIX V5.2</li><li>• Sun Solaris V8</li><li>• Sun Solaris V9</li><li>• Microsoft Windows 2000 Server</li><li>• Microsoft Windows 2000 Advanced Server</li></ul>
Web server	IBM HTTP Server V1.3.26

Software type	Software version
Database Server	One of the following database servers: <ul style="list-style-type: none"> <li>• IBM DB2 Universal Database Enterprise Edition V8.1 with Fix Pack 1</li> <li>• IBM DB2 Universal Database Enterprise Edition V7.2 with Fix Pack 9</li> <li>• Oracle9i RDBMS database V9.2</li> <li>• Oracle8i RDBMS database V8.1.7</li> </ul>
Application Server	IBM WebSphere Application Server, Base Edition (Full Version) V5.0.2

**Note:** The hardware requirements for WebSphere Everyplace Device Manager are as follows:

- ▶ Hard disk: 700 MB
- ▶ RAM: 1 GB
- ▶ Processor:
  - For Intel-compatible processors running Windows : 600 MHz
  - For IBM RS/6000® and SPARC processors: 450 MHz

## B.2 IBM WebSphere RFID Premises Server PVS Starter Kit V1.0.2 matrix

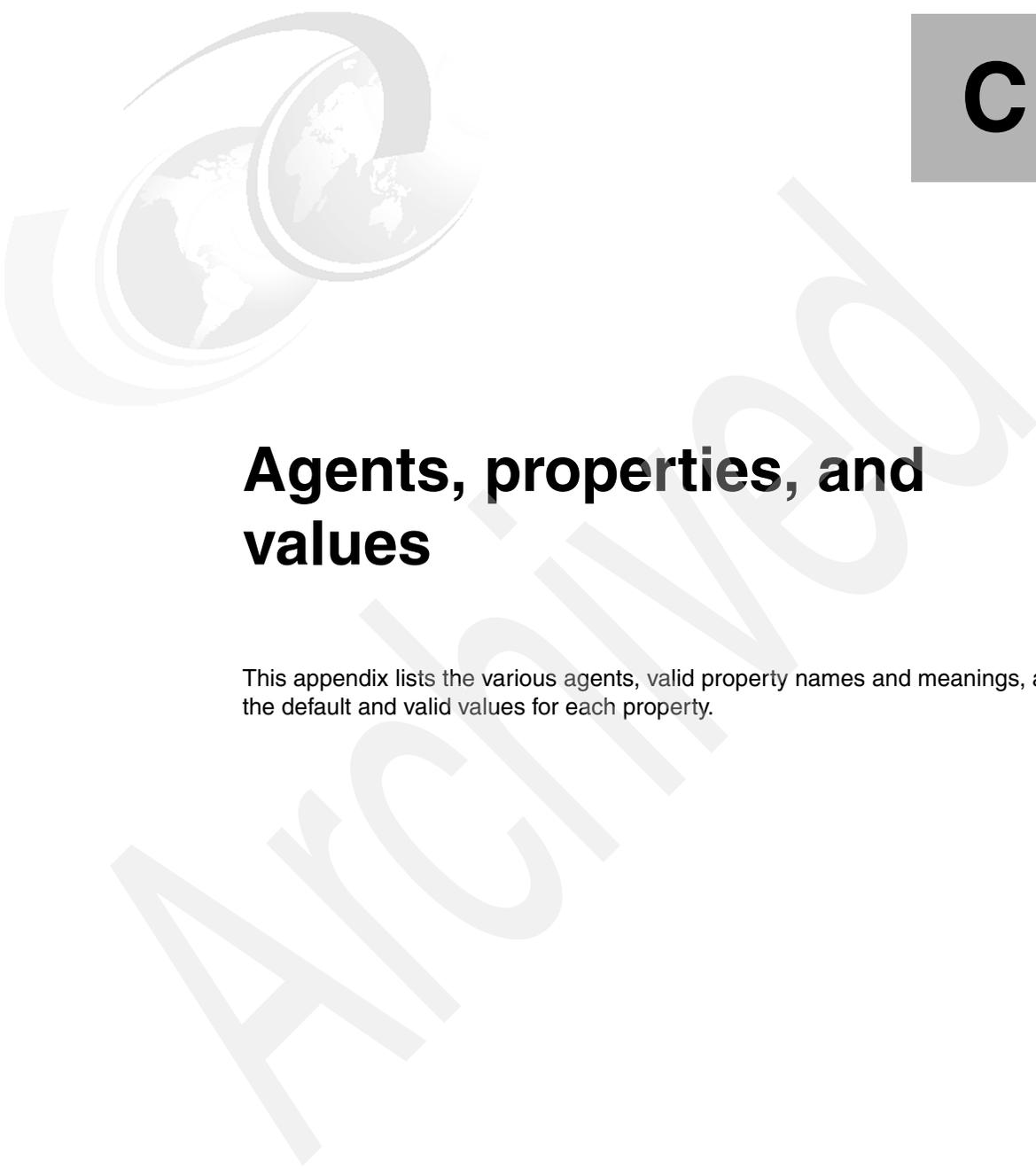
This section presents the Edge Controllers and RFID readers that are supported by IBM WebSphere RFID Premises Server PVS Starter Kit V1.0.2.

Table B-3 provides the list of software required by the Premises Server.

*Table B-3 Premises Server PVS Starter Kit required software*

Software type	Software version
Operating System	One of the following operating systems: <ul style="list-style-type: none"> <li>• Windows 2000 Server Service Pack 4</li> <li>• Windows Server 2003 Service Pack 1</li> </ul>
Database Server	One of the following database servers: <ul style="list-style-type: none"> <li>• DB2 Workgroup Server V8.1.4</li> <li>• Oracle</li> </ul>
Application Server	WebSphere Application Server V5.1 Fix Pack 1 with e-Fix PQ93022
Application Messaging	WebSphere MQ V5.3.0.8 CSD08

Archived



## Agents, properties, and values

This appendix lists the various agents, valid property names and meanings, and the default and valid values for each property.

## C.1 Reader agents

The tables in this section list the properties, default values, and valid values for the reader agents.

### C.1.1 AlienReaderAgent

This agent controls the behavior of Alien RFID tag readers (Table C-1).

Table C-1 AlienReaderAgent properties

Property	Meaning, Default Values, Valid Values	Default Value	Valid Values
green	Output pin on the Digital I/O board to which the green light is attached	2	0-3
red	Output pin on the Digital I/O board to which the red light is attached	0	0-3
amber	Output pin on the Digital I/O board to which the amber light is attached	1	0-3
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	10000	positive integer
alien.pollingreadrate	Rate at which to poll the reader, in milliseconds	666	positive integer
alien.pollingppiorate	Rate at which to poll the reader's digital I/O, in milliseconds	250	positive integer
motion	Input pin on the Digital I/O board to which the motion sensor is attached	0	0, 1
transport.connection	Device Kit transport type for connection to the reader Default: com.ibm.esc.tcpip.connection.TcpipConnection Valid value: com.ibm.esc.tcpip.connection.TcpipConnection		
inputpins	Indicates the types of inputs connected to the Digital I/O board	switch, motion	switch, motion*
switch	Input pin on the Digital I/O board to which the switch is wired	1	0, 1
io.type	Whether the reader or the Edge Controller controls the I/O devices	reader	reader, arcom
beep	Output pin on the Digital I/O board to which the noise signal is attached	3	0-3

\* The valid value for inputpins for this reader agent is the complete string switch,motion. It is not an option to include one or the other.

## C.1.2 IntermecReaderAgent

This agent controls the behavior of Intermec RFID tag readers (Table C-2).

Table C-2 IntermecReaderAgent properties

Property	Meaning	Default Value	Valid Values
green	Output pin on the Digital I/O board to which the green light is attached	2	0-3
red	Output pin on the Digital I/O board to which the red light is attached	0	0-3
amber	Output pin on the Digital I/O board to which the amber light is attached	1	0-3
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	10000	positive integer
intermec.pollingreadrate	Rate at which to poll the reader, in milliseconds	666	positive integer
intermec.pollinggpiorate	Rate at which to poll the reader's digital I/O, in milliseconds	250	positive integer
motion	Input pin on the Digital I/O board to which the motion sensor is attached	0	0, 1
transport.connection	Device Kit transport type for connection to the reader Default: com.ibm.esc.tcpip.connection.TcpipConnection Valid value: com.ibm.esc.tcpip.connection.TcpipConnection		
inputpins	Indicates the types of inputs connected to the Digital I/O board	switch, motion	switch, motion*
switch	Input pin on the Digital I/O board to which the switch is wired	1	0-3
io.type	Whether the reader or the Edge Controller controls the I/O devices	reader	reader, arcom
beep	Output pin on the Digital I/O board to which the noise signal is attached	3	0-3

\* The valid value for inputpins for this reader agent is the complete string switch,motion. It is not an option to include one or the other.

### C.1.3 MatricsReaderAgent

This agent controls the behavior of Matrics RFID tag readers (Table C-3).

Table C-3 *MatricsReaderAgent* properties

Property	Meaning	Default Value	Valid Values
green	This property is required but ignored because I/O is not supported on this reader	2	positive integer
red	This property is required but ignored because I/O is not supported on this reader	0	positive integer
amber	This property is required but ignored because I/O is not supported on this reader	1	positive integer
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	10000	positive integer
matrics.pollingreadrate	Rate at which to poll the reader, in milliseconds	666	positive integer
motion	This property is required but ignored because I/O is not supported on this reader	0	positive integer
transport.connection	Device Kit transport type for connection to the reader Default: <code>com.ibm.esc.tcpip.connection.TcpipConnection</code> Valid value: <code>com.ibm.esc.tcpip.connection.TcpipConnection</code>		
inputpins	This property is required but ignored because I/O is not supported on this reader	switch, motion	switch, motion
switch	This property is required but ignored because I/O is not supported on this reader	1	positive integer
io.type	Must be arcom because I/O is not supported on this reader	arcom	arcom
beep	This property is required but ignored because I/O is not supported on this reader	3	positive integer

## C.1.4 SamsysReaderAgent

This agent controls the behavior of the Samsys RFID tag readers (Table C-4).

Table C-4 *SamsysReaderAgent* properties

Property	Meaning	Default Value	Valid Values
green	Output pin on the Digital I/O board to which the green light is attached	2	0-3
red	Output pin on the Digital I/O board to which the red light is attached	0	0-3
amber	Output pin on the Digital I/O board to which the amber light is attached	1	0-3
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	10000	positive integer
motion	Input pin on the Digital I/O board to which the motion sensor is attached	0	0-3
transport.connection	Device Kit transport type for connection to the reader Default: com.ibm.esc.tcpip.connection.TcpipConnection Valid value: com.ibm.esc.tcpip.connection.TcpipConnection		
inputpins	Indicates the types of inputs connected to the Digital I/O board	switch, motion	switch, motion*
switch	input pin on the Digital I/O board to which the switch is wired	1	0, 1
io.type	Whether the reader or the Edge Controller controls the I/O devices	reader	reader, arcom
beep	Output pin on the Digital I/O board to which the noise signal is attached	3	0, 1

\* The valid value for inputpins for this reader agent is the complete string switch,motion. It is not an option to include one or the other.

## C.1.5 TagSysReaderAgent

This agent controls the behavior of the TagSys RFID tag readers (Table C-5).

**Attention:** The TagSys reader is not supported and is offered *as-is*.

Table C-5 TagSysReaderAgent properties

Property	Meaning	Default Value	Valid Values
red	Output pin on the Digital I/O board to which the red light is attached	4	1-4
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	30000	positive integer
inputpins	Indicates the types of inputs connected to the Digital I/O board	switch, motion	switch, motion*
transport.connection	Device Kit transport type for connection to the reader Default: com.ibm.esc.tcpip.connection.TcpipConnection Valid value: com.ibm.esc.tcpip.connection.TcpipConnection, or com.ibm.esc.com.ibm.esc.serial.connection.service.SerialConnectionService		
io.type	Whether the reader or the Edge Controller controls the I/O devices	reader	reader, arcom
green	Output pin on the Digital I/O board to which the green light is attached	3	1-4
amber	Output pin on the Digital I/O board to which the amber light is attached	99	positive integer > 4
tagsys.tagtype	Specifies the format of the tag type. Valid values: 0 = C210 1 = C220 2 = C240 3 = C270 with anti-collision 4 = C270 with unselected read 5 = C270 with EAS read 6 = ISO 15693 STD 7 = ISO 15693 C370 8 = EPC	6	--
motion	Input pin on the Digital I/O board to which the motion sensor is attached	2	1-4

tagsys.pollingpiorate	Rate at which to poll the reader's digital I/O, in milliseconds	250	positive integer
switch	Input pin on the Digital I/O board to which the switch is wired	1	1-4
tagsys.pollingreadrate	Rate at which to poll the reader, in milliseconds	666	positive integer
beep	Output pin on the Digital I/O board to which the noise signal is attached	99	positive integer > 4
transport.comport	The port number for serial connection (required if transport.connection is set to serial). Valid value is determined by the hardware.	1	--
transport.budrate	The baud rate serial connection (required if transport.connection is set to serial). Valid value is determined by the hardware.	38400	--

\* The valid value for inputpins for this reader agent is the complete string switch,motion. It is not an option to include one or the other.

## C.1.6 SymbolReaderAgent

This agent controls the behavior of the Symbol RFID tag readers (Table C-6).

Table C-6 *SymbolReaderAgent* properties

Property	Meaning	Default Value	Valid Values
green	This property is required but ignored because I/O is not supported on this reader	0	positive integer
red	This property is required but ignored because I/O is not supported on this reader	3	positive integer
amber	This property is required but ignored because I/O is not supported on this reader	99	positive integer
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	20000	positive integer
matrics.pollingreadrate	Rate at which to poll the reader, in milliseconds	666	positive integer
motion	This property is required but ignored because I/O is not supported on this reader	1	positive integer
transport.connection	Device Kit transport type for connection to the reader Default: <code>com.ibm.esc.tcpip.connection.TcpipConnection</code> Valid value: <code>com.ibm.esc.tcpip.connection.TcpipConnection</code>		
inputpins	This property is required but ignored because I/O is not supported on this reader	switch, motion	switch, motion
switch	This property is required but ignored because I/O is not supported on this reader	0	positive integer
io.type	Must be arcom because I/O is not supported on this reader	arcom	arcom
beep	This property is required but ignored because I/O is not supported on this reader	99	positive integer

## C.1.7 FeigUHFReaderAgent

This agent controls the behavior of the FeigUHF RFID tag readers (Table C-7).

**Attention:** The FeigUHF reader is not supported and is offered *as-is*.

Table C-7 FeigUHFReaderAgent properties

Property	Meaning	Default Value	Valid Values
green	Output pin on the Digital I/O board to which the green light is attached	5	5,7
red	Output pin on the Digital I/O board to which the red light is attached	7	5,7
amber	This property is required but ignored	-1	integer <0 or >7
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	10000	positive integer
motion	Input pin on the Digital I/O board to which the motion sensor is attached	0	0
transport.connection	Device Kit transport type for connection to the reader Default: com.ibm.esc.tcpip.connection.TcpipConnection Valid value: com.ibm.esc.tcpip.connection.TcpipConnection		
inputpins	Indicates the types of inputs connected to the Digital I/O board	motion	motion
pollingreadrate	Rate at which to poll the reader, in milliseconds	666	positive integer
pollingppiorate	Rate at which to poll the reader's digital I/O, in milliseconds	250	positive integer
io.type	Whether the reader or the Edge Controller controls the I/O devices	reader	reader, arcom
beep	This property is required but ignored	-1	integer <0 or >7

## C.2 Printer agents

The tables in this section list the properties, default values, and valid values for the printer agents.

**Attention:** The printer agents are only included with the PVS Starter Kit Technology Preview.

### C.2.1 ZebraPrinterAgent

This agent controls the behavior of the Zebra RFID tag printers (Table C-8).

Table C-8 ZebraPrinterAgent properties

Property	Meaning	Default Value	Valid Values
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	10000	positive integer
transport.connection	Device Kit transport type for connection to the reader Default: com.ibm.esc.tcpip.connection.TcpipConnection Valid value: com.ibm.esc.tcpip.connection.TcpipConnection, or com.ibm.esc.com.ibm.esc.serial.connection.service.SerialConnectionService		
transport.comport	Port number for a serial connection, as determined by the hardware	1	--
transport.remoteport	Port number for a TCP/IP connection, as determined by the hardware	9100	--

## C.2.2 PrintronixPrinterAgent

This agent controls the behavior of the Printronix RFID tag printers. Printronix printers have an XML-based interface, Printronix Extensible Markup Language (PXML) that provides the ability to control and monitor directly print jobs and printer status (Table C-9).

Table C-9 PrintronixPrinterAgent properties

Property	Meaning	Default Value	Valid Values
heartbeat.period.ms	Amount of time between heartbeats from the Edge Controller to the reader, in milliseconds	10000	positive integer
pxml.transport.remoteport	Port number for PXML interface status communication, as determined by the hardware	3100	--
transport.connection	Device Kit transport type for connection to the reader Default: com.ibm.esc.tcpip.connection.TcpipConnection Valid value: com.ibm.esc.tcpip.connection.TcpipConnection, or com.ibm.esc.com.ibm.esc.serial.connection.service.SerialConnectionService		
transport.comport	Port number for a serial connection, as determined by the hardware	1	--
transport.remoteport	Port number for a TCP/IP connection, as determined by the hardware	9100	--

## C.3 Controller agents

The tables in this section list the properties, default values, and valid values for the controller agents.

### C.3.1 FilterAgent

This agent controls the tag filtering on the Edge Controller. It specifies which filters should be applied to each tag to determine whether to send the tag to the Premises Server for processing (Table C-10).

Table C-10 *FilterAgent properties*

Property	Meaning	Default Value	Valid Values
duplicates.declay.limit.sec	If you are using DecayingDuplicates, this is the length of time (in seconds) since a tag was last seen that it should stay in the duplicates list else this is ignored	5	positive integer
duplicates.declay.cleanup.sec	If you are using DecayingDuplicates, this is the length of time (in seconds) between purges of old entries on the duplicates list, else this is ignored	2	positive integer
filters	Comma-delimited list of filters applied to all incoming tags Default: Duplicates,CaseTags Valid values: Duplicates, CaseTags, DecayingDuplicates		

Here are the filter definitions:

- ▶ Duplicates  
Filters out tags which have already been read and forwards new tags. It caches tag IDs until the filter agent receives a message on the reset topic to flush the duplicates cache.
- ▶ CaseTags  
Filters out tags which match the case tag property and forwards tags that do not.
- ▶ DecayingDuplicates  
Filters out tags which have already been read, unless the cached tag has expired since the last time it was read.
  - If a new tag is read, it forwards the tag and adds the tag ID to the cache with the current timestamp value.

- If a duplicate tag is read, it compares the age of the timestamp of the last read to the `duplicates.decay.limit.sec` property in the filter agent. If the age of the cached tag is older than the decay time period, then the filter forwards that tag and updates the timestamp of the tag in the cache. If the age of a duplicate tag is within the decay time period, then the tag is not forwarded.

The `DecayingDuplicates` filter also purges expired tags from the duplicate cache at the rate specified by the `duplicates.decay.cleanup.sec` property. The `DecayingDuplicates` cache is flushed entirely when the filter agent receives a message on the `reset` topic to reset each filter.

**Note:** Use one or the other (but not both) of the `Duplicates` and `DecayingDuplicates` filters, depending on the desired behavior.

### C.3.2 ArcomIoDkReaderAgent

This agent controls the general purpose I/O card on Arcom Viper Edge Controller (Table C-11).

Table C-11 *ArcomIoDkReaderAgent* properties

Property	Meaning	Default Value	Valid Values
green	Output pin on the Digital I/O board for the green light	2	0-7
device.filename	Filename for the Digital I/O board	0	0
amber	Output pin on the Digital I/O board to which the amber light is attached	1	0-7
red	Output pin on the Digital I/O board to which the red light is attached	0	0-7
motion	Input pin on the Digital I/O board to which the motion sensor is attached	0	0, 1
inputpins	Indicates the types of inputs connected to the Digital I/O board	switch, motion	switch, motion*
device.path	Path to the device filename for the Digital I/O board Default: <code>/dev/arcom/aim104/relay8</code> Valid values: <code>/dev/arcom/aim104/relay8</code>		
transport.monitor.period.ms	Polling rate for monitoring the Digital I/O devices	250	positive integer

switch	Input pin on the Digital I/O board to which the switch is wire	1	0, 1
beep	Output pin on the Digital I/O board to which the noise signal is attached	3	0-7

\* The valid value for inputpins for this agent is the complete string switch,motion. It is not an option to include one or the other.

### C.3.3 ControllerAgent

The ControllerAgent coordinates behavior between the readers, switches, and motion sensors. This agent currently has no properties.

### C.3.4 PrinterControllerAgent

This agent currently has no properties.

**Attention:** The printer controller agent is included only with the PVS Starter Kit Technology Preview.

## C.4 Other device agents

The tables in this section list the properties, default values, and valid values for the other device agents.

### C.4.1 LightTreeAgent

This agent controls the behavior of the light tree (Table C-12).

Table C-12 *LightTreeAgent* properties

Property	Meaning	Default Value	Valid Values
duration.ms.beep	Amount of time to signal when a beep request is received, in milliseconds	500	positive integer
ignore.green.while.red	Whether any green light indicators should be discarded if the light tree is currently red	false	true, false
duration.ms.green	Amount of time to signal when a green light request is received, in milliseconds	2000	positive integer
duration.ms.red	Amount of time to signal when a red light request is received, in milliseconds	2000	positive integer

## C.4.2 MotionSensorAgent

This agent controls the behavior of the motion sensors (Table C-13).

Table C-13 *MotionSensorAgent* properties

Property	Meaning	Default Value	Valid Values
delayafterquiet	Amount of time to elapse after the motion detector no longer detects activity and it turns off, in milliseconds	2000	positive integer

## C.4.3 SwitchAgent

This agent controls the behavior of the on/off switches at the locations. It currently has no properties.

## C.5 Other agents

The tables in this section list the properties, default values, and valid values for the other agents.

### C.5.1 DutyCycleAgent

This agent monitors the cycle times for the length of time the tag readers are active (Table C-14).

Table C-14 *DutyCycleAgent* properties

Property	Meaning	Default Value	Valid Values
check.interval.ms	Amount of time between checks of the duty cycle times, in milliseconds	1000	positive integer
check.periodically	Whether the agent should check the duty cycle periodically, or only when requested	false	true, false
sampling.peiod.ms	If check.periodically is set to true, the time interval at which you want the agent to check if the duty cycle time has been exceeded, in milliseconds	6000	positive integer
limit.percent	Percentage of time which should cause the monitor to trigger	100	0-100

## C.5.2 SelfTestAgent

The agent controls the input and output behavior at a location when it is placed in self-test mode (Table C-15).

Table C-15 *SelfTestAgent* properties

Property	Meaning	Default Value	Valid Values
input-test-length	Length of time for the input test, after inputs are started, in milliseconds	30000	positive integer
output-delay	Length of time between cycles during the output test, in milliseconds	1000	positive integer

# Glossary

**encoding type.** An algorithm that is applied to a product GID when it is converted to EPC format. This conversion is done in the PVS Administrative Console as part of the PVS configuration. Supported encoding types include: GID96, SGTIN64, SGTIN96, SSCC64, SSCC96, GRAI64, GRAI96, GIAI64, and GIAI96.

**EAN.** European Article Numbering; the European equivalent of a UPC number.

**EPC Electronic product code.** a serialized global identifier (GID) comprised of a 96-bit code that gives each product its own specific identifying number. EPC is the standard for RFID tag labels.

**GID.** Global identifier; an identifier for a product or item, such as a GTIN, EAN13, UCC12.

**GTIN.** Global trade item number; a unique 14-digit identification of products worldwide within the European Article Numbering (EAN) and Uniform Code Council (UCC) systems.

**RFID** Radio frequency identifier; the technology that uses devices that are attached to objects that transmit data to an RFID receiver.

**RFID tag labels.** Sticky labels that are applied to containers of items to facilitate tracking these devices through the supply chain. RFID tag labels include a microchip that stores the encoded EPC data to identify an individual product, and a tag antenna that enables the microchip to transmit ID information to a reader. The labels also have visible item- and customer-specific data, and can optionally have a UPC bar code.

**serialized GID.** Equivalent to an electronic product code (EPC); a serialized global identifier that has been formatted by a specific encoding type to conform to the RFID standard.

**UCC.** Uniform Code Council; a group that is involved with development and maintenance of retail standards.

**UPC.** Universal Product Code; a unique product identification number that is the standard bar code symbol that is used for retail packaging in the United States.

Archived

# Abbreviations and acronyms

<b>ACP</b>	Adaptor Configuration Profile	<b>ESWE</b>	Extension Services for WebSphere Everyplace
<b>AIM</b>	Application Integration Middleware	<b>FIFO</b>	First In First Out
<b>AIX</b>	Application Interactive eXecutive	<b>FRAM</b>	Ferroelectric Random Access Memory
<b>ALE</b>	Application Level Events	<b>GB</b>	Gigabyte
<b>AMI</b>	Application Messaging Interface	<b>GPL</b>	Graphics Programming Language
<b>API</b>	Application Programming Interface	<b>HF</b>	High Frequency
<b>ASN</b>	Advance Shipping Notice	<b>HTTP</b>	Hypertext Transfer Protocol
<b>BCS</b>	Business Consulting Services	<b>I/O</b>	Input/Output
<b>CD</b>	Compact Disk	<b>IT</b>	Information Technology
<b>CDC</b>	Connected Device Configuration	<b>IBM</b>	International Business Machines Corporation
<b>CPU</b>	Central Processing Unit	<b>ID</b>	Identification
<b>CSD</b>	Corrective Service Diskette	<b>IDE</b>	Integrated Development Environment
<b>CSV</b>	Comma Separated Values	<b>IIS</b>	Internet Information Server
<b>DB</b>	Database	<b>IP</b>	Internet Protocol
<b>DC</b>	Direct Current	<b>ISM</b>	Industrial, Scientific, and Medical (radio frequency band)
<b>DDL</b>	Data Definition Language	<b>ISO</b>	International Standards Organization
<b>DNS</b>	Domain Name Server	<b>ITSO</b>	International Technical Support Organization
<b>DOS</b>	Disk Operating System	<b>J2ME</b>	Java2 Micro Edition
<b>EAR</b>	Enterprise Archive	<b>JAAS</b>	Java Authentication and Authorization Service
<b>EDT</b>	Eastern Daylight Time	<b>JDBC</b>	Java Database Connectivity
<b>EJB</b>	Enterprise Java Bean	<b>JMS</b>	Java Message Service
<b>EMEA</b>	Europe-Middle East-Africa	<b>JNDI</b>	Java Naming and Directory Interface
<b>EPC</b>	Electronic Product Code	<b>JVM</b>	Java Virtual Machine
<b>EPCDS</b>	Electronic Product Code Discovery Service	<b>LAN</b>	Local Area Network
<b>EPCIS</b>	Electronic Product Code Information Service		
<b>ERP</b>	Enterprise Resource Planning		

<b>LED</b>	Light Emitting Diode	<b>SAW</b>	Surface Acoustic Wave
<b>LF</b>	Low Frequency	<b>SMF</b>	Service Management Framework
<b>MAC</b>	Media Access Control	<b>SOAP</b>	Simple Object Access Protocol
<b>MB</b>	Megabyte	<b>SQL</b>	Structured Query Language
<b>MBAF</b>	MicroBroker Application Framework	<b>SSL</b>	Secure Socket Layer
<b>MDB</b>	Message Driven Bean	<b>TCP/IP</b>	Transport Control Protocol/Internet Protocol
<b>MHZ</b>	Megahertz	<b>TEC</b>	Tivoli Enterprise Console
<b>MQ</b>	Message Queue	<b>UDDI</b>	Universal Description, Discovery, and Integration
<b>MQI</b>	Message Queue Interface	<b>UHF</b>	Ultra High Frequency
<b>MQSC</b>	MQSeries Commands	<b>UPC</b>	Universal Product Code
<b>MQTT</b>	MQ Telemetry Transport	<b>URL</b>	Uniform Resource Locator
<b>NTIA</b>	National Telecommunication and Information Administration	<b>US</b>	United States
<b>ONS</b>	Object Naming Service	<b>USA</b>	United States of America
<b>OS</b>	Operating System	<b>WAN</b>	Wide Area Network
<b>OSM</b>	Office of Spectrum Management	<b>WCTME</b>	Workplace Client Technology Micro Edition
<b>OTA</b>	Over The Air	<b>WMQTT</b>	WebSphere Message Queue Telemetry Transport
<b>PDA</b>	Personal Digital Assistant	<b>WORM</b>	Write Once Read-only Memory
<b>PLC</b>	Programmable Logic Controllers	<b>XML</b>	Extensible Markup Language
<b>PVS</b>	Print, Verify, Ship	<b>XSL</b>	Extensible Stylesheet Language
<b>PXML</b>	Printronic Extensible Markup Language	<b>ZPL</b>	Zebra Print Language
<b>RAM</b>	Random Access Memory		
<b>RF</b>	Radio Frequency		
<b>RFID</b>	Radio Frequency Identification		
<b>RM/IIOP</b>	Remote Method Invocation/Internet Inter-ORB Protocol		
<b>RO</b>	Read/Only		
<b>ROI</b>	Return On Investment		
<b>RTP</b>	Research Triangle Park, North Carolina		
<b>RW</b>	Read/Write		

# Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this book.

## IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks” on page 274. Note that some of the documents referenced here might be available in softcopy only.

- ▶ *IBM WebSphere V5.0 Security WebSphere Handbook Series*, SG24-6573

## Other publications

These publications are also relevant as further information sources:

- ▶ *RFID Sourcebook*, Print ISBN-13: 978-0-13-185137-5
- ▶ *RFID Essentials*, Print ISBN-13: 978-0-59-600944-1
- ▶ *RFID Field Guide: Deploying Radio Frequency Identification Systems*, Print ISBN-13: 978-0-13-185355-3

## Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ National Telecommunications and Information Administration (NTIA) Office of Spectrum Management (OSM) United States Frequency Allocation Chart  
<http://www.ntia.doc.gov/osmhome/allochrt.pdf>
- ▶ Visit the following URL to get the latest up-to-date list of the supported Edge Controller hardware devices and RFID readers for IBM WebSphere RFID Premises Server V1.0.2:  
[http://www.ibm.com/software/pervasive/ws\\_rfid\\_premises\\_server/technical\\_details](http://www.ibm.com/software/pervasive/ws_rfid_premises_server/technical_details)

- ▶ Visit the WebSphere RFID Premises Server Support URL to get the latest up-to-date IBM WebSphere RFID PVS Starter Kit V1.0.2 installation guide, which provides the list of supported RFID readers and RFID printers:

[http://www.ibm.com/software/pervasive/ws\\_rfid\\_premises\\_server/support](http://www.ibm.com/software/pervasive/ws_rfid_premises_server/support)

## 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

- active tag 12, 20–22
  - micro processors 21
  - several types 21
- Adaptor Configuration Profile (ACP) 215
- application programming
  - interface 59
- Arcom Viper
  - Edge Controller 48, 107, 206, 265
  - Industrial Compact Enclosure 103, 107
  - platform 206

## B

- back-end application 56
- back-end system 8, 12, 80, 182, 184, 190, 196, 199–200, 205–207
  - other enterprise applications 55
  - RFID processes 47
- business logic 37, 40–41, 43, 46, 49, 56, 64–67, 71, 79
- Business Process
  - Integration 36–37, 53, 76
  - Integration Domain 37
  - Transformation 38

## C

- Connection Server Micro Edition 57, 60, 64, 87–90, 97, 150
- controller definition 149, 159, 162, 164, 189

## D

- DB2 Universal Database 59, 112, 114, 220, 251
- Department of Defense (DOD) 43
- Device Kit
  - Extension 94
  - transport type 254–258, 260–263
- Device Manager 103, 109, 143, 219–224, 226–240, 250–251
  - Devices view 236
  - host name 234
  - Job 233
  - OGSi Device Class 228

- server 224, 227–229, 235, 239

- Software view 231

- Digital I/O board 254–255, 257–259, 261, 265

- device filename 265

- input pin 254–255, 257, 259, 266

- Output pin 254–255, 257–258, 261, 265

- Dock Door

- Message selector 71

- dock door 12, 16, 18, 27, 41, 46, 49, 51, 57–58, 67,

- 69, 71, 78, 97–98, 139, 148, 171, 173, 176–179,

- 181–182, 184–185, 188, 195, 199–201, 205, 239

- appropriate reader interrogation zone 12

## E

- Edge Controller 41, 45–48, 50, 54–56, 60, 64–65, 67, 69, 75–77, 79–80, 87–88, 91–93, 97–99, 102–107, 109–110, 132, 138–140, 142–143, 148–152, 161–165, 169–171, 174, 177, 182–185, 188, 190, 192, 194–195, 199, 203–206, 210–211, 213, 216, 219–221, 226–227, 229, 231–238, 241–242, 245–247, 251, 254–258, 260–263, 265
  - alert threshold 164, 210–211
  - custom programming 106
  - filtered tag data 184, 205
  - MicroBroker agents 91
  - OSGi Agent 234–235
  - reference implementation 98
  - software components 103
  - Starting SMF 235
- edge controller
  - MQTT message 70
- Edge Domain 37, 40, 47, 53–54, 74, 76, 103, 105
  - Software updates 37
- edge domain
  - computing power 106
- Electronic Product Code
  - Information Service 38
- Electronic Product Code (EPC) 38–39
- embedded antenna 15
- Enterprise Java Bean (EJB) 67
- Event Handler 70–71, 138–139, 150, 169, 177–179
  - MDB 70–71
  - MDB message selector 70

Event template 66, 149, 169, 171, 174–177, 180, 184  
task definitions associate tasks 169  
Extension Services for WebSphere Everyplace (ES-WE) 83

## F

Ferroelectric Random Access Memory (FRAM) 22  
First In First Out (FIFO) 210

## I

I/O device 20, 48, 97, 103–104, 108, 149–150, 182, 254–255, 257–258, 261, 265  
adapter 161  
API 104  
IBM HTTP Server 117, 125, 222, 226, 229–230, 232, 250  
IBM RFID solution xxi, 1, 10, 13, 18, 26, 30–31, 35–36, 39–43, 45–47, 49–50, 53–55, 57–58, 65, 73–76, 88, 96, 98–99, 101–102, 107, 109–111, 143, 147–150, 169, 178, 182–183, 195, 203–204, 206, 210, 213–214, 217, 219–222, 227, 229, 233, 237–238  
Architecture 45  
Domain 36, 40  
overall multi-tiered design 74  
IBM RFID solution Domain  
Model 36  
IBM Tivoli  
Comprehensive Network Address Translator 62  
Configuration Manager 57, 61, 109  
Enterprise 42, 57, 61–62, 109, 213–215, 217  
Monitoring 42, 57, 62–63, 109  
Monitoring technology 62  
Risk Manager 62  
IBM Tivoli Monitoring  
technology 62  
Web Site 62  
IBM WebSphere Application Server 41, 57–58, 64–65, 67, 102, 109, 112–113, 117, 119, 122, 124–125, 130, 134, 143–144, 172, 178, 189, 208–209, 214–215, 220–221, 224, 226, 237, 250–251  
family 59  
IBM WebSphere RFID Premises Server 40, 47, 53, 55–56, 58, 64, 66, 79, 112–113, 245, 249, 251  
IBM WebSphere RFID Premises Server V1.0.2

RFID readers 77–78, 103–105, 108, 150, 245–247  
IBM WebSphere RFID solution 46  
Industrial, Scientific, and Medical (ISM) 23  
Input pin 254–255, 257–258, 261, 265  
integration point 77  
Intermec Technologies 6, 16, 20, 23–25, 30  
Intermec Butterfly RFID tag insert 20

## J

Java Message Service (JMS) 55, 59, 65, 67–68, 71, 84, 95, 122, 144, 170–172, 174–175  
Java Messaging  
Service 60, 67, 95  
System 60, 65, 68, 170–172, 175  
Java Virtual Machine  
IBM implementation 82  
Java Virtual Machine (JVM) 60–61, 82–83, 125

## L

location contact 151, 153  
Low-Frequency (LF) 9

## M

Message Driven Beans (MDB) 67, 70–71  
message flow 63, 68–69, 173, 184, 205–207  
MicroBroker agent 89, 91–94, 97  
base device kit 91  
MicroBroker Bridge 60, 64–65, 69–70, 88, 91, 93  
MicroBroker Bus 88–89, 91, 93–94  
subscribing agents 93  
motion sensor 12, 97–98, 149–150, 183, 191, 195, 199–200, 254–255, 257–258, 261, 265–267

## N

National Telecommunications and Information Administration (NTIA) 31  
navigation frame 149, 152, 157, 161–162, 166–167, 170, 175, 178, 198  
Network topology 105, 110, 142, 147, 150–151, 155–156, 164–165, 174, 183–185

## O

Object Naming Service (ONS) 38  
Office of Spectrum Management (OSM) 31  
Open Management Alliance (OMA) 84  
Open Services Gateway initiative (OSGI) 41, 82

Output Channel  
  Unique identifier 172  
  various types 172  
Output channel 67, 70, 150, 169, 171–175, 177,  
184  
Output pin 188, 254–255, 257–259, 261, 265

## P

param name 231, 237  
Point-of-Sales (POS) 54  
preconfigured parameter 172  
Premises Server 1, 40–41, 44–45, 47–50, 53–60,  
64–65, 67–68, 70, 74, 76–79, 88, 91, 93, 97–99,  
102–103, 105–107, 109–113, 124–127, 130–131,  
134–144, 147–153, 155, 157, 160–161, 163–166,  
169–171, 174–175, 177–178, 181–184, 188–190,  
192, 194, 196, 198–200, 203–204, 206–207,  
209–210, 213–217, 222, 233, 237–238, 242,  
245–246, 249–251, 264  
  defined WebSphere MQ queues 131  
  Defining Edge Controller configuration 93  
  Edge Controller 56, 76, 106, 140, 237, 264  
  Edge Logfile Adapters 216  
  IP address 106, 110, 112, 148, 198, 238  
  new events 214, 216  
  RFID-specific world 41  
  Services Management Framework runtime 207  
premises server  
  adjacent domains 67  
Print, Verify, and Ship (PVS) 1, 49, 58, 209, 245,  
247, 249, 251, 262, 266

## R

Radio Frequency (RF) 3–4, 9, 13–15, 21, 23,  
29–32, 37, 150  
reader agent 103–104, 106, 151, 161–162, 184,  
186, 199–200, 254, 257, 259  
reader definition 150, 157–158, 161, 186  
  location association 156  
reader type 106, 142, 158, 160, 184, 211, 230–232  
RFID application  
  DDL file 116  
  provider 44  
RFID Device  
  Infrastructure 40–41, 44–45, 48, 55, 60, 73–81,  
87–88, 90–91, 93, 95–97  
  Kit 90–91  
RFID Event

  Server 57–58, 64–66, 68  
RFID event  
  collection 48  
  data 48, 56  
  handling 49  
  information 46, 55–56, 58, 65  
  message 48, 58, 65  
  processing 64–65  
RFID Premises Server  
  CD 11 113  
  product bundle 57  
  Software V1.0.2 57  
  V1.0.2 56, 249–250  
RFID printer 245, 248  
RFID reader 26, 44, 47, 54, 74–79, 86, 92, 97, 99,  
103–104, 106, 108, 110, 142, 149–151, 156,  
182–183, 204–206, 211, 245–247, 251  
  API 103  
  final output 205  
  identifier 110  
  location identifier 110  
  output 205  
  physical location 149  
  simulator 206  
RFID Site Survey 30  
RFID solution 35–36, 39–43, 45–46, 48, 50–51,  
54–55, 58, 66, 73, 76, 98, 101, 111, 143, 147–148,  
182, 203–204, 206, 210, 213, 220  
  deployment 42  
  development 42  
  event information 51, 65  
  individual parts 204  
  middleware development 46  
  other key components 217  
RFID system 10–12, 30–31, 37, 39, 103, 182,  
200–201, 204  
  edge domain 37, 103  
  fundamental building blocks 200  
RFID Tag 8, 15, 20, 24, 40, 66, 150, 167, 182–183,  
188, 190–191, 257, 263  
RFID tag 6, 10, 15–16, 20, 24–25, 66, 97, 182, 190,  
199, 254–256, 258, 260–262  
RFID technology 1, 3–4, 7, 10, 26, 30, 43  
  possible uses 4  
  potential adopters 26  
  Utilization 4

## S

Secure Sockets Layer (SSL) 60  
Service Management Framework 57, 60, 82–84, 86–87, 103, 109, 113, 132  
    key features 83  
    Runtime 83  
Service Management Framework (SMF) 60–61, 64–65, 69–70, 82–83, 86, 127, 132–134, 136–138, 143, 165, 190–192, 194, 207, 209, 234–236, 239, 242  
Simple Object Access Protocol (SOAP) 60  
Starter Kit 1, 49, 51, 57–58, 67, 97, 245, 247, 249, 251, 262, 266  
Surface Acoustic Wave (SAW) 23, 198

## T

tag data 22, 76, 78–79, 97, 147, 149–150, 166–167, 171, 183–184, 190, 192, 196, 198–200, 204–207  
technology preview 1, 49, 58, 246–247, 262, 266  
Tivoli Enterprise 61–62, 213–214, 216–217  
Tivoli Monitoring 57, 62–63, 109  
transport.connection 187–188, 254–259, 261–263

## U

Ultra-High Frequency (UHF) 9, 20, 22  
Universal Description, Discovery, and Integration (UDDI) 59  
Universal Product Code (UPC) 7–8

## V

valid value 162, 253–258, 260–267

## W

Web Infrastructure  
    Client V5.1.2 109  
    V5.1.2 IBM Tivoli Monitoring 57, 62, 109  
Web service 55, 59–60, 65, 95  
WebSphere Application Server 57–58, 103, 109, 112, 117, 120, 125, 130, 134, 143–145, 208–210, 214–215, 221, 224–226, 251  
    IBM infrastructure components 64  
    J2EE application 65  
WebSphere Connection Server Micro Edition  
    important components 88  
    inner workings 89  
WebSphere Everyplace Device Manager 143,

219–224, 226–229, 231–233, 235, 239, 250  
WebSphere MQ 57–59, 63–65, 67–68, 70, 87–88, 112–113, 117, 120–124, 131, 139–140, 143–144, 170–172, 250–251  
    channel 67, 173  
    component 65  
    environment 65, 122  
    Installation 122  
    installation directory 122  
    installation Java directory 123  
    message 88  
    protocol 65  
    queue 65, 131, 173  
    queue manager 173  
    Telemetry Transport 88  
    tray icon 117  
WebSphere RFID  
    Device Infrastructure 41, 49, 55, 60, 73–80, 91, 94–96  
    Device Infrastructure functionality 74  
    Event Server 58  
    infrastructure 46, 55, 60, 74  
    middleware xx, 40, 55  
    Premises Server 47, 53, 56, 58, 65, 80, 112  
    Premises Server CD set 58  
    Premises Server V1.02 58  
    product documentation 96  
    Solution 40, 54, 58, 65, 76, 147  
    Solution middleware 43  
    system 80, 99  
    V1.0.2 Information Center 112  
WebSphere RFID Device Infrastructure  
    functionality 76  
Workplace Client Technology, Micro Edition (WCT-ME) 80



## IBM WebSphere RFID Handbook: A Solution Guide

(0.5" spine)  
0.475" <-> 0.875"  
250 <-> 459 pages







# IBM WebSphere RFID Handbook A Solution Guide



**Redbooks**

## **Get started with IBM WebSphere RFID Premises Server V1.0.2**

This IBM Redbooks publication explains the key products and components that are included in the WebSphere RFID solution, with a focus on WebSphere RFID Premises Server and Device Infrastructure. You will learn how to install and to configure the Premises Server, to connect and to communicate with your edge devices, and to implement the Dock Door Receiving Starter Kit.

## **Understand WebSphere RFID Device InfraStructure**

This book starts with a broad picture of RFID technology and explains how the WebSphere RFID solution fits into that overall technology. It details the features and functions of the Premises Server and includes information about the PVS Starter Kit technology preview.

## **Set up the WebSphere RFID Solution**

It also gives step-by-step instructions for installing the WebSphere RFID Premises Server and related software. You will learn how to use the Administrative Console to configure the Premises Server, Edge Controllers, RFID readers, and the components that enable them to communicate.

This book also explains what the redbook team did to configure the WebSphere RFID system and to run the Dock Door Receiving sample scenario. It also offers suggestions for extending this reference implementation to meet the needs of your enterprise.

## **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)