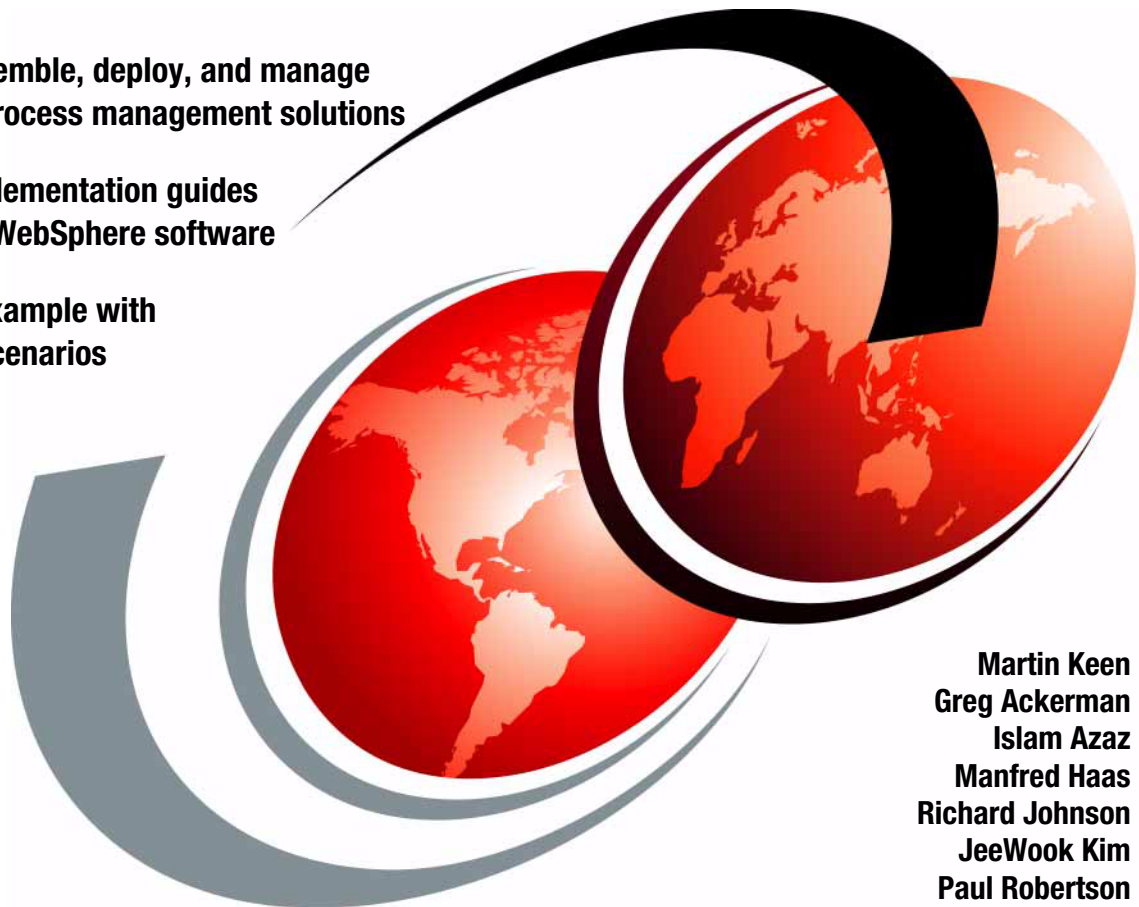


Patterns: SOA Foundation - Business Process Management Scenario

Model, assemble, deploy, and manage
business process management solutions

Follow implementation guides
using IBM WebSphere software

Learn by example with
practical scenarios



Martin Keen
Greg Ackerman
Islam Azaz
Manfred Haas
Richard Johnson
JeeWook Kim
Paul Robertson



International Technical Support Organization

**Patterns: SOA Foundation - Business Process
Management Scenario**

August 2006

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

First Edition (August 2006)

This edition applies to IBM WebSphere Business Modeler V6, IBM WebSphere Integration Developer V6, IBM WebSphere Process Server V6, and IBM WebSphere Business Monitor V6.

© 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

Notices	xi
Trademarks	xii
Preface	xiii
The team that wrote this redbook	xiii
Become a published author	xvi
Comments welcome	xvi
Part 1. SOA Foundation and Business Process Management	1
Chapter 1. Welcome to this redbook	3
1.1 An introduction to this document	4
1.2 How to navigate this book	4
1.2.1 SOA Foundation and the Business Process Management scenario ..	5
1.2.2 Applying SOA scenarios and the IBM Patterns for e-business	5
1.2.3 Business Process Management product capabilities	6
1.2.4 Business Process Management scenario implementation	7
Chapter 2. IBM SOA Foundation	9
2.1 SOA Foundation lifecycle	10
2.2 SOA Foundation Reference Architecture	13
2.3 SOA Foundation scenarios	17
2.3.1 Service Creation scenario	20
2.3.2 Service Connectivity scenario	23
2.3.3 Interaction and Collaboration Services scenario	29
2.3.4 Business Process Management scenario	30
2.3.5 Information as a Service scenario	31
Chapter 3. Business Process Management scenario	33
3.1 Business Process Management overview	34
3.2 IBM products for Business Process Management	35
3.2.1 Summary of IBM products for Business Process Management	35
3.2.2 IBM product descriptions	37
3.3 Roles of Business Process Management	39
Chapter 4. Process for applying SOA scenarios	43
4.1 Process for using SOA scenarios and patterns	44
4.1.1 SOA scenario selection table	46
4.2 Applying the process	48

Chapter 5. Patterns for e-business and Business Process Management	49
5.1 Introduction to the Patterns for e-business	50
5.1.1 Patterns for e-business layered asset model	50
5.1.2 Summary description of patterns	52
5.1.3 Patterns for e-business Web site	52
5.2 Application Integration pattern.	53
5.2.1 Serial Process	53
5.2.2 Parallel Process	59
5.3 Extended Enterprise pattern	64
5.3.1 Exposed Serial Process	65
Part 2. Business Process Management product capabilities	71
Chapter 6. Modeling capabilities of WebSphere Business Modeler.	73
6.1 Business value	74
6.2 Modeling capabilities with WebSphere Business Modeler	75
6.2.1 Overview of WebSphere Business Modeler features	75
6.2.2 Basic business modeling and modeling modes	79
6.2.3 Simulation	82
6.2.4 Business measures design	82
6.2.5 Modeling for execution and monitoring	86
6.2.6 Importing and reusing existing Web services definitions	94
Chapter 7. Assembly capabilities of WebSphere Integration Developer	99
7.1 Business value of using WebSphere Integration Developer	100
7.2 The IBM SOA programming model	101
7.3 Role of the integration developer	104
7.4 Getting to know WebSphere Integration Developer	105
7.4.1 Key component types	105
7.4.2 Integration services	115
7.4.3 Essential terms and concepts	120
Chapter 8. Deployment capabilities of WebSphere Process Server.	121
8.1 Business value	122
8.1.1 Reliable and simple operation environment	122
8.1.2 Protection of investment	123
8.1.3 Agility built into operations	123
8.2 Capabilities	124
8.2.1 Capabilities as an SCA and WS-BPEL runtime environment	124
8.2.2 Capabilities to work with external resources	126
8.2.3 Installation capabilities	127
8.2.4 Capabilities to run BPM applications.	129
8.2.5 Administration capabilities.	130
8.2.6 Capabilities to configure security to use with human tasks.	133

8.2.7 Version control capabilities	134
Chapter 9. Management capabilities of WebSphere Business Monitor .	137
9.1 Business value	138
9.1.1 Basic activities of process monitoring	138
9.1.2 Process monitoring using BPM-enabling solutions	139
9.1.3 Real-time process monitoring using IBM BPM solutions	140
9.2 WebSphere Business Monitor capabilities	141
9.2.1 Real-time business measure extraction	142
9.2.2 Business situation handling.	143
9.2.3 Performance analysis and reporting	143
9.2.4 Using historical data for continuous process improvement.	149
Part 3. Scenario implementation	151
Chapter 10. Business scenario and solution architecture	153
10.1 Business model.	154
10.1.1 Initial context	154
10.1.2 Existing business process	154
10.1.3 Business rules and business measures	155
10.1.4 Roles in the organization.	156
10.2 IT requirements	157
10.2.1 Functional requirements	157
10.2.2 Non-functional requirements	158
10.2.3 System context diagram	158
10.3 Solution architecture	159
10.3.1 Apply fit gap analysis to ITSO Speedy Rentals.	159
10.3.2 Select the SOA scenario	160
10.3.3 Reuse patterns assets to accelerate solution architecture	161
10.4 Implementation details	163
10.4.1 Model	163
10.4.2 Assemble	163
10.4.3 Deploy	165
10.4.4 Manage	166
10.4.5 Continuous improvement	166
Chapter 11. Model with WebSphere Business Modeler	167
11.1 Overview of the process	168
11.2 Modeling the current business process	169
11.3 Creating the As-Is process model	170
11.3.1 Configuring WebSphere Business Modeler	171
11.3.2 Creating the As-Is process flow	176
11.3.3 Creating the business items	181
11.3.4 Connecting tasks with associated data	184

11.3.5	Adding classifiers	189
11.3.6	Adding resources and roles	192
11.3.7	Defining the organization	193
11.3.8	Adding resources and organizations to the process	196
11.4	Simulating the As-Is current process model	198
11.4.1	Configuring simulation.	198
11.4.2	Generating a simulation snapshot and running the simulation	203
11.4.3	Analysis summary and recommendations for the To-Be model	208
11.5	Creating the To-Be process model	209
11.5.1	ITSO Speedy Rentals management team decisions	209
11.5.2	Creating the To-Be project	210
11.5.3	Refining the business items	212
11.5.4	Creating the To-Be process flow.	214
11.6	Modeling for WebSphere Process Server	219
11.6.1	WebSphere Process Server mode considerations and unsupported elements	219
11.6.2	Defining element attributes	220
11.6.3	Technical implementation specification.	226
11.6.4	Adding condition logic	232
11.7	Creating the business measures model	235
11.7.1	Creating business measures.	236
11.7.2	Creating an event for a business situation	239
11.7.3	Dimensional analysis	240
11.7.4	Stopwatches	249
11.7.5	Counters	252
11.7.6	Key performance indicators (KPIs)	256
11.7.7	Aggregate metrics.	258
11.7.8	Synchronizing the business measures model.	259
11.8	Exporting the To-Be model to WebSphere Integration Developer and WebSphere Business Monitor	260
Chapter 12. Assemble with WebSphere Integration Developer		263
12.1	Import necessary artifacts	264
12.1.1	Import artifacts from WebSphere Business Modeler.	264
12.1.2	Import service artifacts	268
12.2	Integrate and assemble the composite application	274
12.2.1	Organize artifacts for reuse between modules	275
12.2.2	Customize logic in main business process	276
12.2.3	Implement subprocess	277
12.2.4	Implement components.	285
12.2.5	Create interface maps and data maps	295
12.2.6	Assemble components into the composite application	302
12.3	Test and deploy the composite application	307

12.3.1	Setting up the test environment	307
12.3.2	Configuring security in the test environment	309
12.3.3	Unit-test components in the composite application	315
12.3.4	Export deployable artifacts for WebSphere Process Server	331
Chapter 13.	Deploy with WebSphere Process Server	333
13.1	Configuring a server with data source and Business Rules Manager	334
13.1.1	Configuring the server to use a data source	334
13.1.2	Installing the Business Rules Manager	336
13.2	Installing the business process application	337
13.2.1	Installing the application using command line	338
13.2.2	Installing the application using the administrative console	339
13.3	Running the application using BPC Explorer	341
13.3.1	Using the BPC Explorer as an end user	341
13.3.2	Using the BPC Explorer as an administrator	347
13.4	Using the Business Rules Manager	348
13.5	Administering a selector component	350
Chapter 14.	Manage with WebSphere Business Monitor	353
14.1	Overview	354
14.2	Deploying the business measures model	354
14.2.1	Schema generation	355
14.2.2	Database configuration	360
14.2.3	Replication configuration	364
14.2.4	Starting the Replication daemons	370
14.2.5	Cube configuration	379
14.2.6	Model import	385
14.2.7	Configuring the adaptive action manager	387
14.3	Testing the business measures model deployment	390
14.4	Creating the dashboards	392
14.4.1	Copying the dashboard view template portlets	392
14.4.2	Creating the dashboard pages	397
14.4.3	Creating the dashboard pages	398
14.4.4	Configuring dashboard views to show data	401
Chapter 15.	Continuous process improvement	417
15.1	Monitoring the existing process	418
15.1.1	Executive overview analysis	418
15.1.2	Aggregated analysis	418
15.1.3	Current rentals analysis	419
15.1.4	Multi-dimensional business analysis	421
15.1.5	Historical trends	426
15.1.6	Exporting values	426
15.2	Improving the process model in WebSphere Business Modeler	427

15.2.1 Adding the Confirm Updated Agreement activity	427
15.2.2 Adding new business measures	429
15.2.3 Export the project	432
15.3 Assembling the new process model	433
15.3.1 Importing new process artifacts	433
15.3.2 Customizing the new process	435
15.3.3 Implementing the new process task	436
15.3.4 Assembling the new modules	437
15.3.5 Deploying and testing the new components	439
15.3.6 Export the enterprise applications	442
15.4 Deploying the new process model to WebSphere Process Server. . .	442
15.4.1 Stopping the old process.	442
15.4.2 Installing the new process.	443
15.5 Managing the new business measures model and process model. . .	444
15.5.1 Generating the new schema from the new model.	444
15.5.2 Monitoring the new process model	448
Part 4. Appendixes	451
Appendix A. Product installation	453
Runtime topology	454
WebSphere Business Modeler Version 6.0	455
Installing WebSphere Business Modeler into WebSphere Integration Developer	455
Fixes	456
WebSphere Integration Developer Version 6.0.1	456
Fixes	456
WebSphere Process Server Version 6.0.1.	457
Fixes	473
WebSphere Business Monitor Version 6.0.	474
Fixes	475
Appendix B. WebSphere Business Monitor system administration . . .	477
Monitor Server	478
Starting the Monitor Server	478
Stopping the Monitor Server	478
Monitor Server log files	478
Monitor Server administrative console	478
Monitor Server user ID and password.	479
Dashboard Client	479
Starting the Dashboard Client	479
Stopping the Dashboard Client	479
Dashboard Client log files	479
Dashboard Client administrative console	479

Dashboard Client user ID and password	480
DB2 Alphablox	480
Starting and stopping DB2 Alphablox	480
DB2 Alphablox log files	480
DB2 Alphablox administrative console	480
DB2 Alphablox user ID and password	480
DB2 Universal Database	481
Starting DB2 Universal Database	481
Stopping DB2 Universal Database	481
DB2 Universal Database log files	481
DB2 Universal Database Control Center	481
DB2 user IDs and passwords	481
Undeploying a business measures model	481
Appendix C. Additional material	487
Locating the Web material	487
Using the Web material	487
System requirements for downloading the Web material	488
How to use the Web material	488
Abbreviations and acronyms	489
Related publications	491
IBM Redbooks	491
Online resources	491
How to get IBM Redbooks	492
Help from IBM	492
Index	493

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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


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

COPYRIGHT LICENSE:

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

Trademarks

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

AIX®	DB2 Universal Database™	Redbooks™
BetaWorks™	developerWorks®	Redbooks (logo)  ™
CICS®	Everyplace®	RUP®
Cloudscape™	IBM®	Tivoli®
Cube Views™	IMS™	WebSphere®
DataPower®	OMEGAMON®	Workplace™
DataStage®	Rational®	zSeries®
DB2®	Rational Unified Process®	

The following terms are trademarks of other companies:

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

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

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

Preface

This IBM® Redbook focuses on the Business Process Management service-oriented architecture (SOA) scenario of the IBM SOA Foundation. The SOA Foundation is an integrated set of software, best practices, and patterns, based on open standards, that is designed to provide you what you need to get started with service-oriented architecture.

This book describes modeling, assembling, deploying, and monitoring business processes using IBM software. It highlights the core products for Business Process Management:

- ▶ IBM WebSphere® Business Modeler V6
- ▶ IBM WebSphere Integration Developer V6
- ▶ IBM WebSphere Process Server V6
- ▶ IBM WebSphere Business Monitor V6

The book also provides a complete end-to-end Business Process Management scenario implementation. Step-by-step instructions are provided for building a scenario for a fictional vehicle rental company called ITSO Speedy Rentals.

The team that wrote this redbook

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

Martin Keen is a Senior IT Specialist at the ITSO, Raleigh Center. He writes extensively about WebSphere products, SOA, and Patterns for e-business. He also teaches IBM classes worldwide about WebSphere, SOA, and business process management. Before joining the ITSO, Martin worked in the EMEA WebSphere Lab Services team in Hursley, United Kingdom. Martin holds a bachelor's degree in Computer Studies from Southampton Institute of Higher Education.

Greg Ackerman is a Certified IT Specialist in Research Triangle Park, North Carolina. He has six years of experience advising IBM customers on building WebSphere solutions. He holds a Bachelor of Science degree in Computer Information Systems from North Carolina Wesleyan College. His areas of expertise include J2EE™, SOA, process integration, and business process management. He has written about WebSphere Process Choreography and IBM

WebSphere Portal, and has spoken extensively at, and helps lead, WebSphere User Groups.

Islam Azaz is a Staff Software Engineer in Egypt working in the IBM Software development lab in Cairo as a member of WebSphere Business Monitor development team. He has five years of experience in the software development field. He has worked at IBM for three years. His areas of expertise include J2EE application design and development, SOA, workflow, Business Performance Management, Business Activity Monitoring, and Electronic Commerce. He holds a B.Sc. degree in Pharmacy from Cairo University, Egypt, and a professional diploma in Software Development and Electronic Commerce.

Manfred Haas is a Consultant IT Specialist in Germany working in the IBM Development Lab in Boeblingen. He has more than 20 years of experience in software development, customer consulting, and in worldwide technical presales support in the field. He holds a Bachelors degree in computer science and informatics. His areas of expertise include Business Process Management and process integration based on the IBM SOA ESP software stack. He has written extensively about Business Process Management and workflow integration and he has been a speaker at several IT conferences.

Richard Johnson is a Senior IT Specialist in United Kingdom, working on the IBM BetaWorks™ team to support customer early programs. He has six years of experience working with customers and IBM products in this role and in his previous role as an IBM Software Services for WebSphere consultant. He creates and delivers consultancy, education, and implementation services in a wide range of areas across SOA, Web services, business integration, and Business Process Management. He holds a Masters degree in Chemistry from the University of Oxford, UK.

JeeWook Kim is a Senior IT Specialist in Korea. He has eight years of experience in WebSphere software and the e-business field. He holds a Bachelors degree in Computer Science from Korea University. His areas of expertise include business process management, e-business, and WebSphere software. He has performed several successful e-business projects in the areas of Internet banking, Internet shopping malls, e-marketplace, and Web application standardization. He has performed many proof-of-concept projects extensively in the areas of business process management and enterprise application integration. He has written extensively about WebSphere Process Integration.

Paul Robertson is a certified IT Architect in the UK, working in the Channel Technical Sales team supporting IBM Business Partners and Systems Integrators. He has more than 15 years of experience working on customer engagements utilizing IBM software products. He holds a degree in Electronics from University of Dundee and Ph.D. from St. Andrews. His areas of expertise

include business process management and business integration. He has written extensively on the areas of Business Process Modeling and Monitoring.



Figure 1 The IBM Redbook team (left to right): Islam, Manfred, Richard, Paul, Greg, JeeWook, and Martin

Thanks to the following people for their contributions to this project:

- ▶ Special thanks to Erica Carmel, IBM WebSphere Software Platform User Experience, USA
- ▶ Special thanks to Jonathan Adams, Distinguished Engineer, Project and Technical leader of the IBM Patterns for e-business, IBM UK
- ▶ John Ganci and Carla Sadtler, IBM ITSO Raleigh Center, USA
- ▶ Daniel Tishman, ITSO Intern, Penn State University
- ▶ David Enyeart, Ben Parees, Patrick Titzler, Gregg Snodgrass, Ken Sari, Hubert Fort, Scott Walden, Jim Thorpe, and Raymond Kong, Randolph Williams, Billy Rowe

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

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. HZ8 Building 662
P.O. Box 12195
Research Triangle Park, NC 27709-2195



Part 1

SOA Foundation and Business Process Management



Welcome to this redbook

This chapter introduces this book to you and provides guidelines for how to read it. It includes the following sections:

- ▶ An introduction to this document
- ▶ How to navigate this book
 - SOA Foundation and the Business Process Management scenario
 - Applying SOA scenarios and the IBM Patterns for e-business
 - Business Process Management product capabilities
 - Business Process Management scenario implementation

1.1 An introduction to this document

Welcome to this redbook, from the book's authors. We all gathered for six intense weeks in Raleigh, North Carolina, to put together this resource. We hope you find it a useful read.

This book is part of a series of three Redbooks that introduce scenarios from the IBM SOA Foundation. This volume is based on the Business Process Management SOA Foundation scenario. In this book you will find an introduction to the IBM SOA Foundation, and all of the scenarios within it, as well as detailed information specific to Business Process Management. This information includes a look at the IBM products that deliver Business Process Management capabilities, and an in-depth, step-by-step guide for building a working Business Process Management scenario for a fictional car rental company.

You might find the other two books in this series to be of interest:

- ▶ *Patterns: SOA Foundation - Service Creation Scenario*, SG24-7240.
Introduces the SOA Foundation in greater depth than in this book, and provides a step-by-step guide for building a working Service Creation SOA scenario. This includes use of IBM Rational® Application Developer, IBM WebSphere Application Server, and IBM Tivoli® Composite Application Manager for SOA.
- ▶ *Patterns: SOA Foundation - Service Connectivity Scenario*, SG24-7228
Provides a step-by-step guide for building a working Service Connectivity SOA scenario. This includes the use of IBM WebSphere Integration Developer, IBM WebSphere Enterprise Service Bus, and Tivoli Composite Application Manager for SOA.

1.2 How to navigate this book

This book provides guidance about how to use the core IBM products that deliver Business Process Management capabilities. It also offers step-by-step instructions for building a working Business Process Management solution using these products, for a fictional company. Additionally, it describes how to use SOA Foundation scenario selection criteria and the Patterns for e-business assets to help architect your own Business Process Management solution.

Use this section to determine the chapters that are of most interest to you.

1.2.1 SOA Foundation and the Business Process Management scenario

The IBM SOA Foundation is an integrated, open standards based set of IBM software, best practices, and patterns designed to provide what you need to get started with SOA from an architecture perspective.

We provide an overview of the SOA Foundation, and each of the SOA scenarios within it in the following chapter:

- ▶ Chapter 2, “IBM SOA Foundation” on page 9

Note: If you are not familiar with SOA concepts and terms, we recommend first reading Chapter 1 of *Patterns: SOA Foundation - Service Creation Scenario*, SG24-7240.

We then provide a more detailed introduction into Business Process Management, including a description of the products used, and the roles and resources required. This can be found in the following chapter:

- ▶ Chapter 3, “Business Process Management scenario” on page 33

1.2.2 Applying SOA scenarios and the IBM Patterns for e-business

A process for selecting SOA scenarios and related reusable patterns assets has been designed and documented in *Patterns: SOA Foundation - Service Creation Scenario*, SG24-7240. This process can be used by specialists in pre-sales and post-sales roles, and by architects in a customer / consulting engagement.

We provide an overview of this process in the following chapter:

- ▶ Chapter 4, “Process for applying SOA scenarios” on page 43

Part of the process for applying SOA scenarios involves the use of the Patterns for e-business. The Patterns for e-business approach enables architects to implement successful e-business solutions through the reuse of components and solution elements from proven successful experiences.

The Patterns for e-business that apply to Business Process Management are described in:

- ▶ Chapter 5, “Patterns for e-business and Business Process Management” on page 49

The Patterns for e-business are an optional step in helping to select an SOA Foundation scenario and product mapping.

1.2.3 Business Process Management product capabilities

This book includes detailed information about the Business Process Management capabilities offered by a core set of IBM products, with each product described in its own chapter.

Use the following chapters to gain a deeper understanding of these products:

- ▶ Chapter 6, “Modeling capabilities of WebSphere Business Modeler” on page 73
- ▶ Chapter 7, “Assembly capabilities of WebSphere Integration Developer” on page 99
- ▶ Chapter 8, “Deployment capabilities of WebSphere Process Server” on page 121
- ▶ Chapter 9, “Management capabilities of WebSphere Business Monitor” on page 137

Note: Tivoli Composite Application Manager for SOA is also a core product for the Management phase of Business Process Management. This book does not include a chapter describing the Business Process Management capabilities of Tivoli Composite Application Manager for SOA. This product is fully discussed in the chapter titled “Manage Solutions with Tivoli Composite Application Manager” of the IBM Redbook *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240.

Additional products can be used for Business Process Management, including:

- ▶ Model: IBM Rational Software Architect
- ▶ Deploy:
 - IBM WebSphere Portal
 - IBM WebSphere Adapters
- ▶ Manage:
 - IBM Tivoli Access Manager
 - IBM Tivoli Federated Identity Manager
- ▶ Governance: IBM WebSphere Service Registry and Repository

For example, WebSphere Portal could be used to provide a portal front end for users completing human tasks within a business process running in WebSphere Process Server.

Additionally, the Business Process Management SOA scenario could be combined with other SOA scenarios. For example, it could be combined with the

Service Connectivity SOA scenario, so that service calls from WebSphere Process Server are routed through WebSphere Enterprise Service Bus. For more information about the Service Connectivity SOA scenario, see:

- ▶ *Patterns: SOA Foundation - Service Connectivity Scenario*, SG24-7228

1.2.4 Business Process Management scenario implementation

The use of the core IBM products for the Business Process Management SOA scenario is demonstrated in a scenario. The scenario provides step-by-step instructions for the modeling, assembly, deployment, and management of a vehicle fulfillment system for a fictional company called ITS0 Speedy Rentals.

The fictional ITS0 Speedy Rentals organization is described in the following chapter, which also illustrates how the process for applying SOA scenarios and patterns is used:

- ▶ Chapter 10, “Business scenario and solution architecture” on page 153

The step-by-step instructions for building this solution are described in:

- ▶ Chapter 11, “Model with WebSphere Business Modeler” on page 167
- ▶ Chapter 12, “Assemble with WebSphere Integration Developer” on page 263
- ▶ Chapter 13, “Deploy with WebSphere Process Server” on page 333
- ▶ Chapter 14, “Manage with WebSphere Business Monitor” on page 353

Note: This book does not include a chapter with step-by-step instructions for Tivoli Composite Application Manager for SOA. Step-by-step instructions for this product are provided in the chapter titled “Manage Solutions with Tivoli Composite Application Manager” of the redbook *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240

Finally, we provide a chapter describing how to improve the business process for our fictional scenario using all of the products noted above, to illustrate how continuous process improvement can be achieved:

- ▶ Chapter 15, “Continuous process improvement” on page 417

A graphic featuring two overlapping globes showing the Americas, with a large, stylized 'G' shape formed by two curved lines passing behind them.

IBM SOA Foundation

IBM SOA Foundation is an integrated, open standards based set of IBM software, best practices, and patterns designed to provide what you need to get started with service-oriented architecture from an architecture perspective. The key elements of the IBM SOA Foundation are the SOA lifecycle (model, assemble, deploy, manage), reference architecture, and SOA scenarios.

The SOA Foundation scenarios (or simply SOA scenarios) are representative of common scenarios that use IBM products and solutions for SOA engagements. The SOA scenarios communicate the business value, architecture, and IBM open standards based software used in the SOA scenario.

The SOA scenarios can be used as a reference architecture (starting point) to accelerate the SOA architecture and implementation of a customer scenario. These scenarios can be implemented using an incremental SOA adoption approach, whereby a customer can incrementally add elements of other SOA scenarios to the environment to achieve business objectives.

Note: For a more detailed explanation of the SOA Foundation, refer to *IBM SOA Foundation: An Architectural Introduction and Overview V1.0*, found at:
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf>

2.1 SOA Foundation lifecycle

IBM customers have indicated that they think of SOA in terms of a lifecycle. The IBM SOA Foundation includes the following lifecycle phases (Figure 2-1):

- ▶ Model
- ▶ Assemble
- ▶ Deploy
- ▶ Manage

There are a couple of key points to consider about the SOA lifecycle:

- ▶ The SOA lifecycle phases apply to all SOA projects.
- ▶ The activities in any part of the SOA lifecycle can vary in scale and the level of tooling used depending on the stage of adoption.

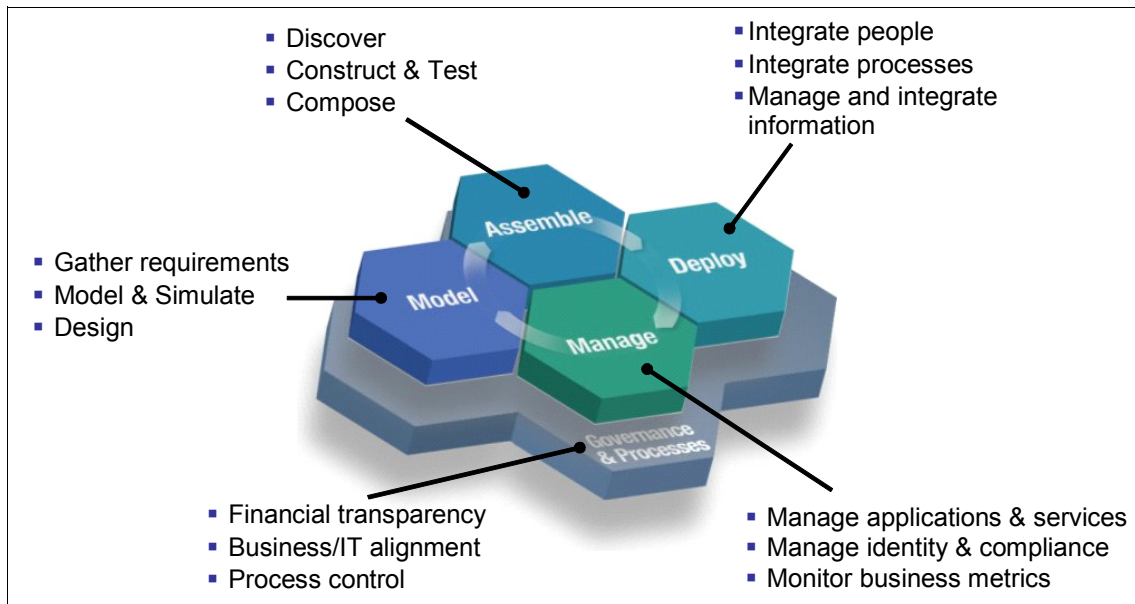


Figure 2-1 IBM SOA Foundation lifecycle

Model

Modeling is the process of capturing the business design from an understanding of business requirements and objectives. Business requirements are translated into a specification of business processes, goals, and assumptions for creating a model of the business. Many businesses do not go through a formal modeling exercise. In some cases, businesses that do perform modeling use primitive

techniques such as drawing the design in Microsoft® Visio® or using text documents.

Capturing the business design using a sophisticated approach that includes the use of specialized tooling enables you to perform “what-if” scenarios with various parameters that the business might experience. The process can then be simulated using those parameters to predict the effect that the process will have on the business and IT systems. If the achieved results do not match the business objectives, then the process definition can be refined.

The model also captures key performance indicators, such as business metrics that are important measurements of your business. For example, this could include a measure of the new accounts that you have opened in a given month. These key performance indicators are input to the assembly of the application. In addition, the indicators can be monitored in production to capture the critical data to measure whether the objectives are being met.

Assemble

The business design is used to communicate the business objectives to the IT organization that will assemble the information system artifacts that implement the design. The enterprise architect works closely with the business analyst to convert the business design into a set of business process definitions, as well as activities used to derive the required services from the activity definitions. The enterprise architect and business analyst work with the software architect to flesh out the design of the services.

During the process of resolving the design and implementation of the modeled business processes and services, a search of existing artifacts and applications should be performed in an effort to find components that meet the needs of the design. Some applications will fit perfectly, some will have to be re-factored, and some will have to be augmented to meet the requirements of the design.

These existing assets should be rendered as services for assembly into composite applications. Any new services that are required by the business design have to be created. Software developers should use the SOA programming model to create these new services.

Finally, the assemble phase includes applying a set of policies and conditions to control how your applications operate in the production runtime environment. For example, these policies and conditions include business and government regulations. In addition, the assemble phase includes critical operational characteristics such as packaging deployment artifacts, localization constraints, resource dependency, integrity control, and access protection.

Deploy

The deploy phase of the lifecycle includes a combination of creating the hosting environment for the applications and the deployment tasks of those applications. This includes resolving the application's resource dependencies, operational conditions, capacity requirements, and integrity and access constraints.

A number of concerns are relevant to construction of the hosting environment including the presence of the existing hosting infrastructure supporting applications and pre-existing services. Beyond that, you need to consider appropriate platform offerings for hosting the user interaction logic, business process flows, business services, access services, and information logic.

Manage

The manage phase includes the tasks, technology, and software used to manage and monitor the application assets such as services and business processes that are deployed to the production runtime environment.

Monitoring is a critical element of ensuring that the underlying IT systems and application are up and running to maintain the service availability requirements of the business. Monitoring also includes monitoring performance of service requests and timeliness of service responses. In addition, monitoring includes maintaining problem logs to detect failures in various services and system components, as well as localizing failures and restoring the operational state of the system.

Managing the system also involves performing routine maintenance; administering and securing applications, resources and users; and predicting future capacity growth to ensure that resources are available when the demands of the business call for it. The security domain includes such topics as authentication, single sign-on, authorization, federated identity management, and user provisioning.

The manage phase also includes managing the business model, tuning the operational environment to meet the business objectives expressed in the business design, and measuring success or failure to meet those objectives. SOA is distinguished from other styles of enterprise architecture by its correlation between the business design and the software that implements that design, and its use of policy to express the operational requirements of the business services and processes that codify the business design. The manage phase of the lifecycle is directly responsible for ensuring that those policies are being enforced and for relating issues with that enforcement back to the business design.

Governance

SOA Governance is critical to the success of any SOA project. Governance helps clients extend the planned SOA across the enterprise in a controlled manner. SOA Governance has four core objectives or challenges:

- ▶ Establish decision rights
- ▶ Define high-value business services
- ▶ Manage the lifecycle of assets
- ▶ Measure effectiveness

2.2 SOA Foundation Reference Architecture

This section describes the SOA Foundation Reference Architecture, which includes the components and middleware services used by applications in the runtime environment.

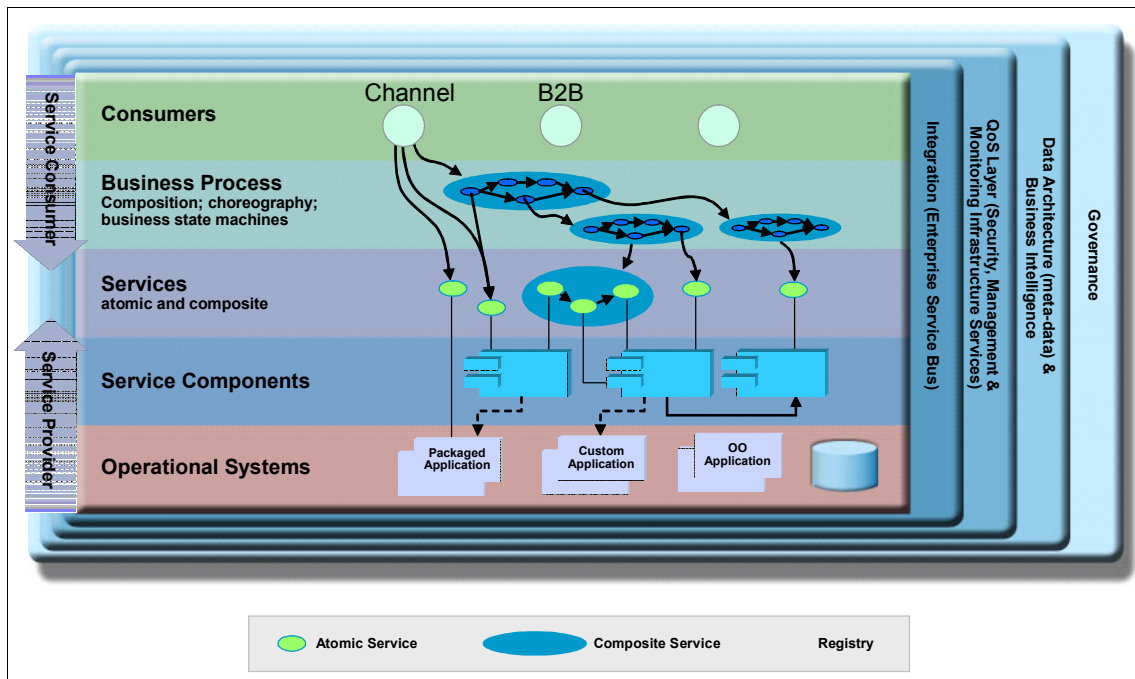


Figure 2-2 SOA Foundation Reference Architecture: Solution view

Figure 2-2 depicts the SOA Foundation Reference Architecture solution view used to decompose an SOA design. SOA puts a premium on the role of the *Enterprise Architect*, who is responsible for spanning between the business design and the information system that codifies that design.

When taking a top-down approach, the enterprise architect starts by identifying the business processes and business services used by business users. Business users are consumers of the processes and services. Business processes should be treated as compositions of other business processes and services, and thus should be decomposed into their subordinate sub-processes and services.

Note: This flow describes a top-down approach. Other variations include a bottom-up approach, and the more common meet-in-the-middle approach.

Services and business processes are then detailed into service components. Service components include a detailed set of definition metadata used to describe the service to the information system. Services can be aggregated into module assemblies. The module assemblies are used to establish related design concerns and begin the planning to determine what teams will collaborate to implement the related services to be deployed as a single unit.

The resulting set of business process definitions, services, and schema will make up the logical architecture of the application. The enterprise architect then maps that logical architecture to a physical architecture.

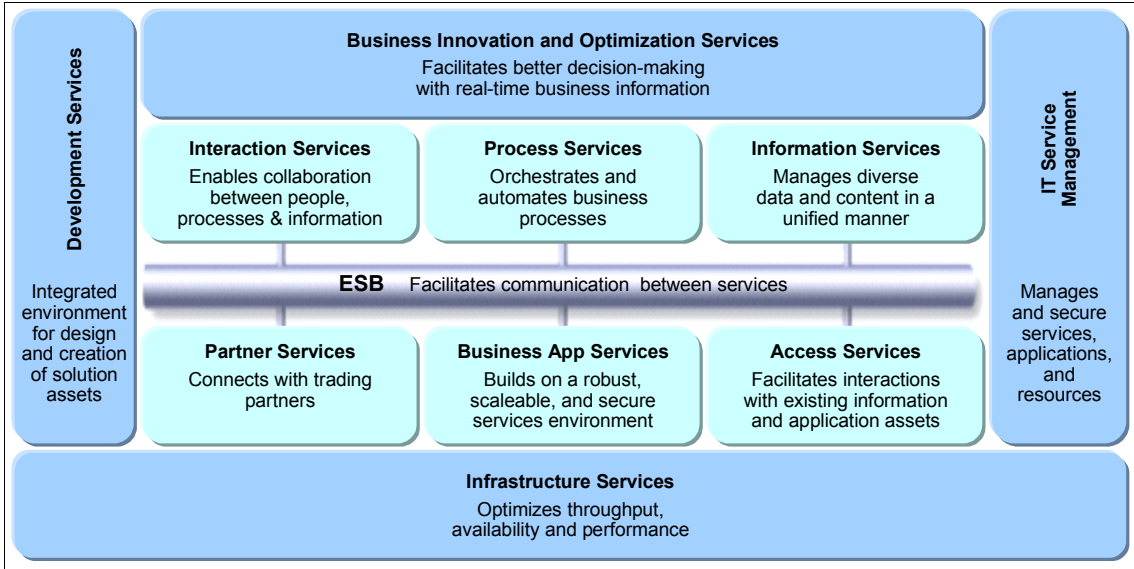


Figure 2-3 SOA Foundation Reference Architecture: Middleware Services view

We have included a summary description for each of the services found in the logical architecture displayed in Figure 2-3. The services found in the center of Figure 2-3 (Interaction, Process, Information, Partner, Business Application, Access) are the core set of services used by applications within the runtime

environment when deployed. The other services displayed in Figure 2-3 on page 14 (outer services) are used in support of the core services.

Core components of the logical architecture

This section includes a brief description of the following core components of the logical architecture:

- ▶ Interaction services
- ▶ Process services
- ▶ Business application services
- ▶ Information services
- ▶ Access services
- ▶ Partner services

Interaction services

Interaction services provide the capabilities that are required to deliver IT functions and data to users, meeting their specific preferences.

Process services

Process services provide the required control capabilities for managing the flow and interactions of multiple services in ways that implement business processes.

Business application services

Business application services are called by service consumers. Service consumers include other components in the logical architecture such as a portal or business processes.

Information services

Information services provide the capabilities necessary to federate, replicate, and transform disparate data sources.

Access services

Access services provide bridging capabilities between core applications, prepackaged applications, enterprise data stores, and the ESB to incorporate services that are delivered through existing applications into an SOA.

Partner services

Partner services provide the document, protocol, and partner management capabilities for business processes that involve interactions with outside partners and suppliers.

Supporting components of the logical architecture

This section includes a brief description of the supporting components of the SOA Foundation logical architecture used in support of the core components:

- ▶ Enterprise Service Bus
- ▶ Business Innovation and Optimization Services
- ▶ Development services
- ▶ IT service management
- ▶ Infrastructure services

Enterprise Service Bus

The Enterprise Service Bus (ESB) or simply *bus*, provides an infrastructure that removes the direct connection dependency between service consumers and providers. Consumers connect to the bus and not the provider that actually implements the service. This type of connection further decouples the consumer from the provider. A bus also implements further value-add capabilities, such as security and delivery assurance. It is preferred to implement these capabilities centrally within the bus at an infrastructure level rather than within the application. The primary driver for an ESB, however, is that it increases decoupling between service consumers and providers.

Although it is relatively straightforward to build a direct link between a consumer and provider, these links can lead to an interaction pattern that consists of building multiple point-to-point links that perform specific interactions. With a large number of interfaces this quickly leads to the build-up of a complex spaghetti of links with multiple security and transaction models. When routing control is distributed throughout the infrastructure, there is typically no consistent approach to logging, monitoring, or systems management. This type of environment is difficult to manage or maintain and inhibits change.

Note: An ESB can be thought of as an architectural pattern, with an implementation to match the deployment needs. IBM has two ESB products:

- ▶ IBM WebSphere Enterprise Service Bus
- ▶ IBM WebSphere Message Broker

In addition, several products extend the capabilities of these ESBs, including IBM WebSphere DataPower® XML Security Gateway XS40.

Business Innovation and Optimization Services

Business innovation and optimization services are primarily used to represent the tools and the metadata structures for encoding the business design, including the business policies and objectives.

Business innovation and optimization services exist in the architecture to help capture, encode, analyze, and iteratively refine the business design. The services also include tools to help simulate the business design. The results are used to predict the effect of the design, including the changes the design will have on the business.

Development services

Development services encompass the entire suite of architecture tools, development tools, visual composition tools, assembly tools, methodologies, debugging aids, instrumentation tools, asset repositories, discovery agents, and publishing mechanisms that are needed to construct an SOA-based application.

IT service management

After the application has been deployed to the runtime environment, it has to be managed along with the IT infrastructure on which it is hosted. IT service management represents the set of management tools that are used to monitor your service flows, the health of the underlying system, the utilization of resources, the identification of outages and bottlenecks, the attainment of service goals, the enforcement of administrative policies, and recovery from failures.

Infrastructure services

Infrastructure services form the core of the information technology runtime environment used for hosting SOA applications. These services provide the ability to optimize throughput, availability, performance, and management.

2.3 SOA Foundation scenarios

The SOA Foundation scenarios (or simply SOA scenarios) are representative of common scenarios of using IBM products and solutions for SOA engagements. The SOA scenarios quickly communicate the business value, architecture, and IBM open standards based software used in the SOA scenario. The SOA scenarios can be implemented as part of an incremental adoption of SOA growing from one scenario to using elements of multiple scenarios together. *Realizations* are used to provide more specific solution patterns and IBM product mappings within the SOA scenarios. (A realization is an example business case that describes a customer situation and the solution.)

The SOA scenarios can be used as a reference architecture implementation (starting point) to accelerate the SOA architecture and implementation of your customer scenario.

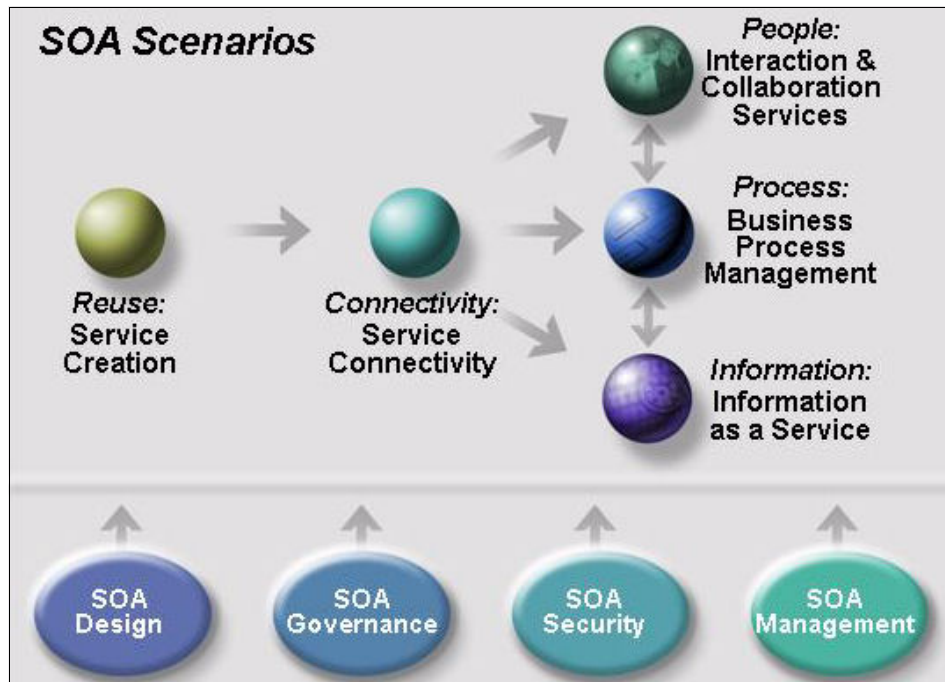


Figure 2-4 SOA scenarios and entry points

Figure 2-4 displays the SOA scenarios (Service Creation, Service Connectivity, Interaction and Collaboration Services, Business Process Management, Information as a Service), and the relationship among the scenarios.

We have included examples of how the scenarios can be used together and adopted incrementally. For example, it is common that the other scenarios will include service creation and often want connectivity. In addition, the scenarios can be used together such as a portal accessing a business process or a portal accessing an information service through an ESB from a service consumer.

SOA Design, SOA Governance, SOA Security, and SOA Management can be used in each of the SOA scenarios based on customer requirements.

SOA Design is focused on service modeling and service design. The service model is an abstraction of the IT services that are implemented in an enterprise and support the development of one or more service-oriented solutions. It is used to conceive and document the design of the software services. It is a comprehensive, composite work product encompassing all services, providers, specifications, messages, collaborations, and relationships between them.

The service design model extends the standard IBM Rational Unified Process® (RUP®) design model by adding the service components. The RUP for SOA service design includes service identification and modeling techniques derived from IBM Service-Oriented Modeling and Architecture (SOMA). The service components artifact is intended for use in describing the realization of a service specification. A service component might provide the realization for one or more services by the realization of multiple service specifications. The set of model elements on the inside of the component represents the concrete realization of the structural, behavioral, and policy contract described by these service specifications.

SOA Governance is critical to the success of any SOA project. Governance helps clients extend the planned SOA across the enterprise in a controlled manner. SOA Governance has four core objectives or challenges:

- ▶ Establish decision rights
- ▶ Define high-value business services
- ▶ Manage the lifecycle of assets
- ▶ Measure effectiveness

SOA Security encompasses the need for a security model that enables secure business transactions across and between enterprises. The security domain includes such topics as Web Services Security, authentication, single sign-on, authorization, identity propagation, federated identity management, and user provisioning.

SOA Management includes solutions for managing and monitoring composite applications. We highlight three key points regarding SOA management. First, it is important to understand the relationship of service consumers and providers for composite applications. SOA brings the benefits of application reuse. The importance of monitoring and managing the availability and performance of the application functionality exposed as a service increases when reused by service consumers that depend on this functionality.

Second, composite applications span the various layers of the architecture. Composite applications drive a mind shift in the management approach. There is a need for monitoring and management tooling that cover the end-to-end view of the composite application, as well as provide detailed information about performance and availability metrics for the individual components.

Third, SOA management is used to ensure management non-functional requirements of the IT architecture are aligned with the business objectives. Monitoring and management of the composite application should include specific metrics to ensure that Service Level Agreements (SLAs) are meeting the business objectives.

We have provided a summary for each of the following SOA scenarios:

- ▶ Service Creation scenario
- ▶ Service Connectivity scenario
- ▶ Interaction and Collaboration Services scenario
- ▶ Business Process Management scenario
- ▶ Information as a Service scenario

2.3.1 Service Creation scenario

The Service Creation scenario (that is, service provider and service consumer) is used to demonstrate exposing application functionality of an existing application or new business logic as a service. The services may then be consumed by other services or client applications within an enterprise and between enterprises. The key driver for this scenario is the reuse of existing or new application functionality as services.

Note: For a more detailed explanation of the Service Creation scenario, refer to *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240.

There are many possible examples to illustrate the Service Creation scenario. We have included a summary of the following four common realizations of the Service Creation scenario:

- ▶ Directly expose existing applications as services
- ▶ Indirectly expose existing applications via service components
- ▶ Create an EJB Web service from WSDL
- ▶ Consume services from third-party service providers

One of the goals of enterprise transformation is to enable access to Enterprise Information System (EIS) applications as services, with the objective of leveraging the investment of existing business applications and systems. By adopting an SOA approach, the EIS application functionality can be turned into reusable services that can be consumed by a new set of client applications and users.

The Service Creation scenario includes a two methods of exposing EIS applications as services, known as direct and indirect exposure. We will use CICS® as an example of an existing EIS application for both the direct and indirect exposure realization examples.

Directly expose existing applications as services

In this realization, we directly expose existing applications as services. The existing application can be a wide range of application types such as an EIS (for example, CICS or IMS™), J2EE application, SAP, and so on. A key distinction for

this realization is that the service interface to be exposed is defined by the existing application. This approach to creating a service is known as bottom-up.

We use CICS as an example of an existing EIS application. In this example, we access applications hosted by CICS Transaction Server *directly* as Web services.

The core IBM products used for this example realization are as follows:

- ▶ Assemble: WebSphere Developer for zSeries® V6.0.1
- ▶ Deploy:
 - IBM CICS Transaction Server V3.1
 - IBM WebSphere MQ V6

Additionally, the following IBM products may be considered, depending on specific customer requirements:

- ▶ Manage: IBM Tivoli OMEGAMON® XE for CICS V3.10
- ▶ Governance: WebSphere Service Registry and Repository

WebSphere Service Registry and Repository: IBM currently intends to make available in the second half of 2006 a WebSphere service registry and repository capability that will enable customers to securely register business services for finding, publishing, and notifying changes to SOA infrastructure components such as enterprise service bus and process servers. Customers will also be able to house the metadata about business services in managing the lifecycle of a service in SOA. The new capabilities will also include a model for governance to provide guidance and oversight for a SOA project.

All statements regarding IBM plans, directions, and intent are subject to change or withdrawal without notice.

Indirectly expose existing applications via service components

For this realization, we expose existing application functionality *indirectly* via service components. Here, business alignment is achieved by defining the service interface (WSDL), independently of existing assets. This approach to creating a service is known as top-down. In practice, the implementation of the service may be a combination of top-down and meet-in-the-middle.

We use CICS as an example of an existing EIS. In this realization example, we expose existing COMMAREA applications hosted by CICS Transaction Server indirectly by creating middle-tier Web services to access CICS. The middle-tier Web service wraps a session Enterprise JavaBean (EJB™) that uses the CICS ECI resource adapter to communicate with the CICS Transaction Gateway (CTG) to access CICS Transaction Server. The CICS ECI resource adapter is a JCA adapter for WebSphere Application Server packaged with the CTG.

The core IBM products that are used for this example realization are as follows:

- ▶ Assemble: Rational Application Developer V6.0.1
- ▶ Deploy:
 - WebSphere Application Server V6 for zSeries or distributed platform depending on the selected runtime topology
 - IBM CICS Transaction Gateway V6.1
(CICS ECI resource adapter for WebSphere is included with the CICS TG)
 - IBM CICS V2.x

Additionally, the following IBM products may be considered, depending on specific customer requirements:

- ▶ Model: Rational Software Architect V6
- ▶ Manage: Tivoli OMEGAMON XE for CICS V3.1
- ▶ Governance: WebSphere Service Registry and Repository

Create an EJB Web service from WSDL

In this realization, we create a new session EJB Web service from an existing service WSDL. This realization is also known as *create from scratch*. This realization uses a top-down approach to creating a Web Service.

The core IBM products used for this realization are as follows:

- ▶ Assemble: Rational Application Developer V6
- ▶ Deploy: IBM WebSphere Application Server Network Deployment V6
- ▶ Manage: Tivoli Composite Application Manager for SOA V6

Additionally, the following IBM products may be considered, depending on specific customer requirements:

- ▶ Model: Rational Software Architect V6
- ▶ Deploy: WebSphere Service Registry and Repository
- ▶ Manage: Tivoli Composite Application Manager for WebSphere V6
- ▶ Governance: WebSphere Service Registry and Repository

Consume services from third-party service providers

In this realization, client applications consume services from third-party service providers. This realization represents the view of a consumer that is using one or more third-party services. The service consumer sees only the service interfaces. The endpoints and any security or transport constraints are imposed by the service provider.

For example, a Web services client application might invoke an address verification Web service from a third-party service provider.

This realization assumes that the WSDL is WS-I compliant and compatible with JAX-RPC. The service provider is outside the enterprise firewall, and security is implemented between the consumer and provider using mutual SSL authentication.

The core IBM products used for this realization are as follows:

- ▶ Assemble: Rational Application Developer V6
- ▶ Deploy: WebSphere Application Server V6

Additionally, the following IBM products may be considered, depending on specific customer requirements:

- ▶ Manage:
 - Tivoli Composite Application Manager for SOA V6
 - IBM Tivoli Composite Application Manager for WebSphere V6
- ▶ Governance: WebSphere Service Registry and Repository

2.3.2 Service Connectivity scenario

The Service Connectivity scenario is used to demonstrate the integration of service providers and consumers, allowing for the reuse of existing and new services across multiple channels. This scenario is appropriate for an enterprise that has a set of core services or systems that are to be made available as services to a variety of internal and external clients. Flexibility to make changes to service providers and make changes to service clients independent of each other is a requirement.

The focus of this scenario is on the underlying connectivity used to support business-centric SOA. An enterprise service bus provides decoupling between clients and providers, providing the flexibility to implement applications more quickly. In circumstances where services are provided to or consumed from a third party, an ESB gateway can be used in conjunction with the ESB to add security measures. An ESB gateway alone might be sufficient if all of your service interactions are with third parties and you have the basic requirements to mediate between service consumers and providers.

Note: *Patterns: SOA Foundation - Service Connectivity Scenario*, SG24-7228 offers a more detailed description of the Service Connectivity scenario.

Implementations of this scenario have the following features:

- ▶ Enables changes to the implementation of a service without affecting clients.
- ▶ Registers services to a service registry.

- ▶ Uses an enterprise service bus as the integration point between service providers and service consumers.
- ▶ Enables clients to access a service with a different interface and protocol than what the service consumer supports.
- ▶ Uses an ESB gateway to isolate and protect services.
- ▶ Enables management and monitoring of services to ensure Service Level Agreements.
- ▶ Provides security and credential mapping (where needed) to ensure proper use of the services.

Specific connectivity and integration requirements for an enterprise will ultimately drive product selection of the ESB and supporting products. The choice of runtime products might include one or more of the following products:

- ▶ WebSphere Message Broker
- ▶ WebSphere Enterprise Service Bus
- ▶ IBM WebSphere DataPower SOA Appliances
- ▶ Web Services Gateway (IBM WebSphere Application Server Network Deployment V6)
- ▶ WebSphere Adapters
- ▶ WebSphere Service Registry and Repository (available 2H06)

The choice of SOA lifecycle products depends largely on the runtime products selected. The following products can be used to support the runtime environment:

- ▶ WebSphere Message Broker Toolkit
- ▶ Rational Application Developer
- ▶ WebSphere Integration Developer
- ▶ Tivoli Composite Application Manager for SOA
- ▶ Tivoli Composite Application Manager for WebSphere
- ▶ IBM Tivoli Composite Application Manager for Response Time Tracking V6.0
- ▶ IBM Tivoli OMEGAMON for Messaging
- ▶ Tivoli Access Manager
- ▶ Tivoli Federated Identity Manager

Realizations have been developed to help you understand how the scenario can be used and how products are selected. We have included a summary of the following common realizations of the Service Connectivity scenario:

- ▶ Gateway: For use when interactions with third parties are present and mediation requirements are basic.
- ▶ Local integration: For use with standards-based interactions that require routing capabilities.
- ▶ Web services access to Enterprise Information Systems: For use when requiring access to EIS systems.
- ▶ Expose existing systems to heterogeneous clients: For use in a diverse, non-standards based environment.

Gateway

An ESB gateway can be used alone or in conjunction with an ESB to provide controlled and secure service interaction between internal or external domain boundaries. In this realization, its primary function is to provide secure access to resources when interacting with third parties. An ESB gateway can also provide basic functionality such as protocol switching and message switching to enable interaction between service consumers and service providers.

This realization assumes that the customer has adopted standards-based technology, has an existing infrastructure, and has the following business requirements:

- ▶ Standards-based requestors/providers will use SOAP/HTTP for transport.
- ▶ Dynamically add new providers and requestors at runtime.
- ▶ Support a defined, high response time with a moderate load.
- ▶ SOA security for interaction with requestors and providers. Security may have to be adapted between the requestors and providers.
- ▶ Requests and responses must be logged to a file.

The following technical requirements have been identified:

- ▶ Many services that are deployed will require the same mediation flow. An ESB gateway will minimize administration and streamline the process for making new services available.
- ▶ Services must be monitored for performance and usage.
- ▶ Monitoring for all components must be integrated into existing management infrastructure.

The IBM products used for this realization are as follows:

- ▶ Deploy: DataPower XML Security Gateway XS40

The customer chooses the DataPower XS40 model for the runtime. The XS40 is designed specifically to provide XML acceleration and SOA security and can provide the required basic mediation functions. Because requirements for mediating the interaction between consumers and providers are met, the XS40 as an ESB gateway is sufficient and no ESB product is required.

- ▶ Assemble: DataPower Toolkit
- ▶ Manage: Tivoli Composite Application Manager for SOA

Tivoli Composite Application Manager for SOA will be used to monitor Web services flowing through the DataPower appliance.

- ▶ Manage: Tivoli Access Manager

The XS40 can be integrated with Tivoli Access Manager to secure applications.

Local integration

A local integration solution provides multi-channel access for clients to an existing service with a range of connectivity options for standards-based clients and services, message routing with or without the use of external data, message transformation and augmentation, protocol switching, and security.

With this type of connectivity, a client can request a secure service without knowledge of its location. Transparent to the client, requests can be routed to the service that can best handle the request. Also transparent to the client is the message format and transport protocol required to access the provider. The response could be immediate or delayed.

This realization assumes that the customer has adopted standards-based technology, has an existing WebSphere Application Server infrastructure, and has the following business requirements:

- ▶ Integration of multiple client channels to service providers
- ▶ Routing of client requests to the appropriate service provider
- ▶ Intranet environment that does not require WS-Security or other complex security considerations
- ▶ Able to support moderate volume of requests

The following technical requirements have been identified:

- ▶ Message data from clients must be examined in order to determine the service provider to route the request to.

- ▶ Clients and service providers will use JMS, SOAP/JMS, or SOAP/HTTP.
- ▶ Data transformation will be required. This should be done with XSLT.

The core IBM products used for this realization are as follows:

- ▶ **Deploy: WebSphere Enterprise Service Bus**
WebSphere ESB provides the transport flexibility to support the transports required by the customer. WebSphere ESB also has the mediation capabilities that are required to perform the message routing and transformation.
- ▶ **Assemble: WebSphere Integration Developer**
WebSphere Integration Developer is the development and assembly tool for building WebSphere ESB mediations.
- ▶ **Manage: Tivoli Composite Application Manager for SOA**
Tivoli Composite Application Manager for SOA will be used to monitor Web services requests as they arrive at WebSphere ESB.

Web services access to Enterprise Information Systems

An ESB can be used to provide access to EIS systems through the use of adapters. Mediations in the ESB are used to adapt the client request to a form understood by the adapter, and then to adapt the response to the client's format.

This realization assumes that the customer has adopted standards-based technology, has an existing WebSphere Application Server infrastructure, and has the following business requirements:

- ▶ Can provide Web service access to functionality in an Enterprise Information System such as SAP R/3, PeopleSoft, or Oracle Financials
- ▶ An intranet environment that does not require WS-Security or other complex security considerations
- ▶ The integration is based on message exchange/data replication scenarios; no business process or data synchronization between clients and EIS systems
- ▶ Can support a moderate volume of requests

The following technical requirements have been identified:

- ▶ The targeted integration is point-to-point, although multiple EISs can be exposed as Web services at the same time.
- ▶ Data transformation will be required. This should be done with XSLT.
- ▶ Log the messages as they flow through the hub; want to log asynchronously to a file.

The IBM products used for this realization are as follows:

- ▶ **Deploy: WebSphere Enterprise Service Bus and WebSphere Adapters**
WebSphere ESB supports the SOAP/HTTP transport required by the customer. WebSphere Adapters provide the EIS adapters required. WebSphere ESB also provides the mediation capability required to do XSLT transformation on the data and includes a logging function to log messages as they flow through the mediation.
- ▶ **Assemble: WebSphere Integration Developer**
WebSphere Integration Developer is the development and assembly tool for building WebSphere ESB mediations. It includes the enterprise discovery capabilities needed to incorporate the WebSphere Adapters into the mediation applications.
- ▶ **Manage: IBM Tivoli Composite Application Manager for SOA**
Tivoli Composite Application Manager for SOA will be used to monitor Web services requests as they arrive at WebSphere ESB.

Expose existing systems to heterogeneous clients

An integration solution that includes a range of diverse business applications must provide connectivity for a wide range of service consumers and service providers as well as advanced options for message mediation, including message augmentation, message routing, and the ability to decompose messages into multiple requests and to recompose the responses.

This type of connectivity would provide the most advanced options for integrating dissimilar and widespread service consumers and service providers. Clients can request a secure service that may be provided by one or more service providers with the service composition occurring within the ESB. Services and clients also have a wide range of connectivity options. Connectivity to legacy applications as well as standards-based applications are managed by the ESB.

This realization assumes the customer has extensive legacy systems as well as some newer Web services based systems and has these business requirements:

- ▶ Providers use a variety of heterogeneous protocols.
- ▶ Any provider must be accessible via basic Web services that will be used by a variety of clients.
- ▶ Can support a moderate volume of requests.
- ▶ Intranet environment does not require SOA security or other complex security considerations.
- ▶ Global transactions across multiple heterogeneous transaction managers for some providers.

The following technical requirements have been identified:

- ▶ The ESB must support communication protocol conversion.
- ▶ The ESB must support flexible data model conversion, with acceptable performance and adequate tooling.
- ▶ Enterprise-class persistent messaging backbone.
- ▶ Global transactions management.
- ▶ The ESB must “adapt” the service definitions between the requestors and providers.

The IBM products used for this realization are as follows:

- ▶ Deploy: WebSphere Message Broker, WebSphere MQ, and WebSphere Adapters

WebSphere Message Broker is selected to provide the ESB capabilities, including mediation support. WebSphere MQ will be used to provide an enterprise-class persistent messaging backbone. This combination will support the wide variety of transport protocols and conversions required for the integration solution. WebSphere Adapters provide connectivity to legacy systems.

- ▶ Assemble: Message Brokers Toolkit

The Message Brokers Toolkit is the development tool for building mediation message flows in WebSphere Message Broker and provides the runtime configuration and management tools.

2.3.3 Interaction and Collaboration Services scenario

The Interaction and Collaboration Services scenario features single sign-on and a role-based portal used to consolidate access to information and application within the enterprise and between enterprises.

The key drivers for this scenario are to improve people productivity and consumability of applications and content. The content can be personalized in the aggregated portal page based on the user role.

The core IBM products used for the Interaction and Collaboration Services scenario are as follows:

- ▶ Assemble:
 - Rational Application Developer V6
 - Bowstreet Portlet Factory
- ▶ Deploy: WebSphere Portal V5.1
- ▶ Manage: Tivoli Composite Application Manager for SOA V6

Additionally, the following IBM products may be considered by a customer depending on specific requirements:

- ▶ Assemble: WebSphere Integration Developer
- ▶ Deploy:
 - WebSphere Process Server
 - IBM WebSphere Everyplace® Deployment Server
- ▶ Manage:
 - Tivoli Access Manager
 - Tivoli Federated Identity Manager

2.3.4 Business Process Management scenario

Business Process Management leads to business innovation and optimization by implementing business strategy through modeling, developing, deploying, and managing business processes throughout the entire lifecycle. Business Process Management acts as an enabler for businesses in defining and implementing strategic business goals and then measuring and managing a company's financial and operational performance against these goals.

Note: See Chapter 3, “Business Process Management scenario” on page 33 for a more detailed description of the Business Process Management scenario.

Business Process Management combines business processes, information, and IT resources, aligning the organization's core assets, people, information, technology, and processes to create a single integrated view, with real-time intelligence, of both its business measurements and IT system performance.

The IBM process integration portfolio provides capabilities required for delivery of comprehensive enterprise-wide Business Process Management strategies and solutions. It offers a holistic approach to transform and manage a business by aligning strategic and operational objectives with business activities and supporting IT services. The IBM Business Process Management solution includes development tools that are used to implement custom artifacts that leverage the infrastructure capabilities, and business performance management tools, which are used to monitor and manage the runtime implementations at both the IT and business process levels.

Core IBM products used for the Business Process Management scenario are:

- ▶ Model: WebSphere Business Modeler V6
- ▶ Assemble: WebSphere Integration Developer V6
- ▶ Deploy: WebSphere Process Server V6

- ▶ Manage:
 - WebSphere Business Monitor V6
 - Tivoli Composite Application Manager for SOA V6

The additional IBM products used for this scenario, depending on customer requirements, are as follows:

- ▶ Model: Rational Software Architect V6
- ▶ Deploy:
 - WebSphere Portal
 - WebSphere Adapters
- ▶ Manage:
 - Tivoli Access Manager
 - Tivoli Federated Identity Manager
- ▶ Governance: WebSphere Service Registry and Repository

2.3.5 Information as a Service scenario

The Information as a Service scenario is used to demonstrate unified and trusted information available as services from multiple data sources. This scenario includes content management, e-forms, security, business intelligence, information integration through just-in-time integration, and ETL (extract, transform, and load).

The key drivers for this scenario are to facilitate better decision making and better information sharing between business operations.

Core IBM products used in the Information as a Service scenario are as follows:

- ▶ Model:
 - WebSphere Business Modeler
 - Rational Data Architect
- ▶ Assemble:
 - WebSphere Integration Developer
 - IBM WebSphere DataStage® Designer
- ▶ Deploy:
 - IBM WebSphere Information Server
 - WebSphere DataStage Integration Suite
- ▶ Manage:
 - WebSphere Business Monitor
 - Tivoli Composite Application Manager for SOA

Additionally, the following IBM products may be considered by a customer depending on specific requirements:

- ▶ Deploy:
 - WebSphere Portal
 - WebSphere Adapters
 - IBM Workplace™ Forms View / Server
- ▶ Manage:
 - Tivoli Access Manager
 - Tivoli Federated Identity Manager



Business Process Management scenario

This chapter introduces the Business Process Management SOA scenario from the SOA Foundation. In this chapter, you will find more information about the IBM products that have been identified as core products in implementing Business Process Management.

This chapter contains the following sections:

- ▶ Business Process Management overview
- ▶ IBM products for Business Process Management
- ▶ Roles of Business Process Management

3.1 Business Process Management overview

Business Process Management combines business processes, information, and IT resources, aligning an organization's core assets—people, information, technology, and processes—to create a single integrated view, with real-time intelligence, of both its business measurements and IT system performance.

Business Process Management allows for an enterprise to be flexible and responsive to the ever-changing On Demand Business through the optimization and automation of the business processes to:

- ▶ Identify and eliminate redundancies and bottlenecks
- ▶ Decouple business integration logic from its underlying implementation code
- ▶ Increase portability and decrease maintenance cost by being based on industry standards
- ▶ Automate process implementation, eliminate manual deployment tasks
- ▶ Immediately execute new business rules and processes
- ▶ Visualize actual process performance against key performance indicators
- ▶ Pinpoint future process improvements

A robust business process management solution will provide powerful tools for the business and IT sides to meet their challenges:

- ▶ The *business executive* needs:
 - Revenue growth with cost containment
 - Responsiveness to business conditions and ability to pursue new market opportunities
 - Improving internal skills, capabilities, and leadership as first step toward growth
- ▶ The challenges for the *IT executive* are:
 - Aligning IT and business goals to grow revenue and contain costs
 - Building responsiveness and agility into the organization through IT
 - Enabling people and teams to be more effective through IT

3.2 IBM products for Business Process Management

This section introduces the core products of the Business Process Management scenario and the role they play, and provides a more detailed description of these products.

The core IBM products for Business Process Management are as follows:

- ▶ Model: WebSphere Business Modeler V6.0
- ▶ Assemble: WebSphere Integration Developer V6.0
- ▶ Deploy: WebSphere Process Server V6.0
- ▶ Manage:
 - WebSphere Business Monitor V6.0
 - Tivoli Composite Application Manager for SOA V6.0

3.2.1 Summary of IBM products for Business Process Management

The core IBM products for Business Process Management are used as follows:

▶ Model

WebSphere Business Modeler is an Eclipse-based toolset to model business processes. It is designed for business analysts to model, simulate, and optimize business processes before they hand over the model to IT for implementation refinements.

▶ Assemble

WebSphere Integration Developer is an Eclipse-based integration toolset for assembly of composite applications. This tooling is designed for IT developers and IT architects. It links directly with WebSphere Business Modeler for seamless interaction between different roles and organizations.

▶ Run

WebSphere Process Server is a runtime environment for flexible deployment of business processes. It makes plug-and-play of components a reality. WebSphere Process Server offers the secure, robust, and scalable environment needed to deploy mission-critical business processes.

▶ Manage

WebSphere Business Monitor offers real-time visibility into process performance, enabling process intervention and continuous improvement. It allows for visualization of key performance indicators, so that the health of the business can be monitored and arising problems can be pinpointed, allowing for immediate resolution. It includes support for monitoring processes running in WebSphere Process Server.

Tivoli Composite Application Manager for SOA gives you real-time visibility into Web services interactions. It can be used to monitor Web services calls into and out of WebSphere Process Server.

Figure 3-1 shows the role these products play in the Business Process Management scenario.

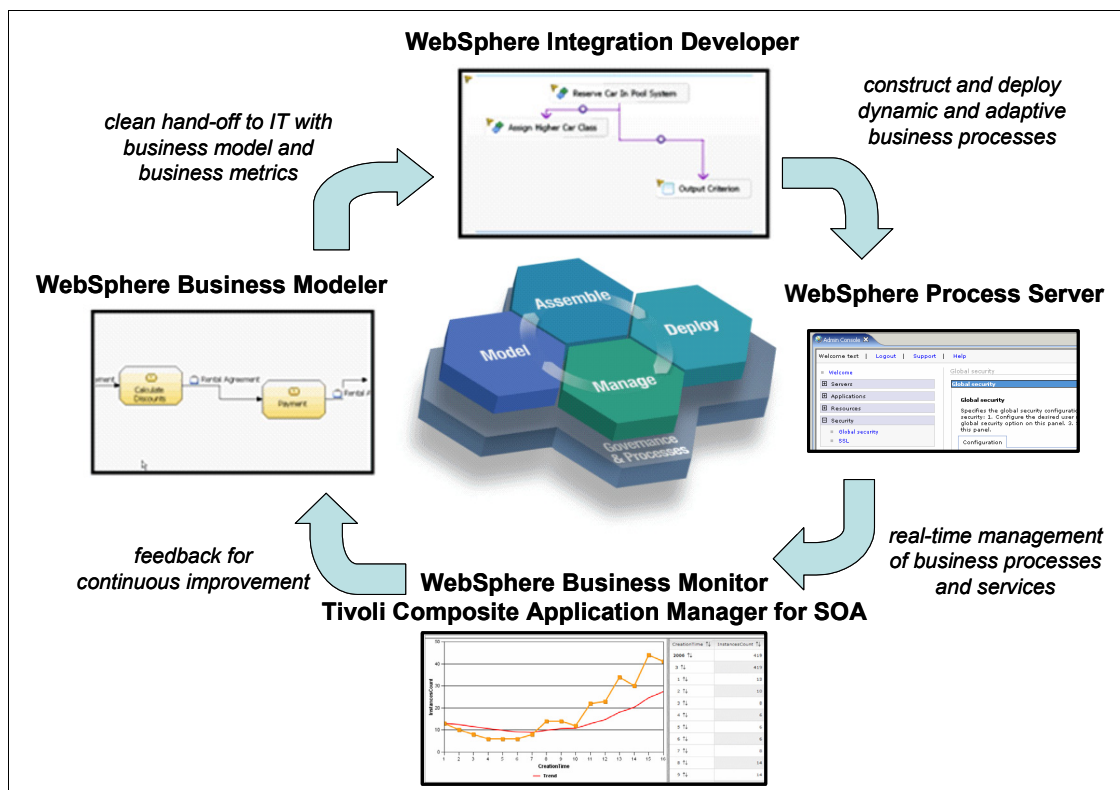


Figure 3-1 Business Process Management scenario lifecycle

The Business Process Management scenario can be combined with other SOA Foundation scenarios to extend functionality. Two examples of this are:

- ▶ Service calls from WebSphere Process Server could be sent to an Enterprise Service Bus instead of invoking these services directly. This would make use of the Service Connectivity scenario, described in *Patterns: SOA Foundation - Service Connectivity Scenario*, SG24-7228.
- ▶ The user interface for completing human tasks in WebSphere Process Server could be implemented as portals using WebSphere Portal Server. This would make use of the Interaction and Collaboration Services scenario.

3.2.2 IBM product descriptions

This section provides more information about each of the core IBM products for Business Process Management.

WebSphere Business Modeler

IBM WebSphere Business Modeler is targeted at business analysts to help capture business design. WebSphere Business Modeler can be used for documentation and compliance purposes, providing a visual and textual representation of processes, information, organization, resources, classifiers, and business measurements that can be shared across an organization. You can import static diagrams that you have created previously in Microsoft Visio.

WebSphere Business Modeler includes a simulation tool that enables analysis of processes and testing of how well processes will hold up under different operating assumptions. You can use this analysis to refine and optimize business design. WebSphere Business Modeler is built on the Eclipse tool framework, which makes it easy to share information about your business design with other parts of your organization and tools. In particular, you can export a design into WebSphere Integration Developer and Rational Software Architect so that application developers can use it as a blueprint for designing process flows for automating business design.

Find more information about WebSphere Business Modeler at:

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

WebSphere Integration Developer

IBM WebSphere Integration Developer is an Eclipse-based tool designed to help create business process flows, state machines, and business rules.

WebSphere Integration Developer V6.0 simplifies integration with its Service Component Architecture (SCA). SCA uses Business Process Execution Language for assembling business process tasks into workflows, which can then be deployed to IBM WebSphere Process Server.

WebSphere Integration Developer can directly import business models from the IBM Business Modeler. You can then use a wiring editor for assembling service components, for importing service interface definitions, and for setting binding policies to build SOA-enabled applications.

Find more information about WebSphere Integration Developer at:

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

WebSphere Process Server

IBM WebSphere Process Server is the primary hosting environment for business processing. Built on WebSphere Enterprise Service Bus, WebSphere Process Server includes support for both Web Services Business Process Execution Language (WS-BPEL) based process flows and business state machines. WebSphere Process Server also supports the integration of business rules in process and service selection. The Process Server is the first product in the IBM suite to offer direct support for the Service Component Architecture SOA programming model.

WebSphere Process Server integrates with WebSphere Portal to deliver business process management through a portal — there is support for human tasks in a business process. Human tasks are defined as activities within the process definition that are carried out by users. This includes built-in support for task assignment, pick lists, scheduling, and escalation policies in case a task is not processed in a timely fashion.

Find more information about IBM WebSphere Process Server at:

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

WebSphere Business Monitor

IBM WebSphere Business Monitor enables you to monitor business processes in real time, providing a visual display of business process status.

WebSphere Business Monitor complements WebSphere Business Modeler. It helps in creating dashboards for visualizing the performance of a business, based on the key performance indicators that are identified in a business design. You can use this to track time, cost, and resources used in processes. WebSphere Business Monitor provides tools that enable you to set situational triggers and notifications of potential bottlenecks or workload imbalances.

Ultimately WebSphere Business Monitor helps you better understand how your business design achieves your business objectives, and provides guidance about how to refine and optimize that business design in case goals are not being met.

Find more information about IBM WebSphere Business Monitor at:

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

Tivoli Composite Application Manager for SOA

IBM Tivoli Composite Application Manager is designed specifically to enable IT service management by understanding the unique semantics and loosely coupled characteristics of SOA-based services. Tivoli Composite Application Manager has three editions that are relevant directly to the SOA Foundation: IBM Tivoli Composite Application Manager for WebSphere, IBM Tivoli Composite

Application Manager for SOA, and IBM Tivoli Composite Application Manager for Response Time Tracking. These cover the gamut from application server monitoring and resource consumption, deep-dive diagnostics and correlation as service invocations cascade across multiple systems, and service-level response times and problem isolation.

Tivoli Composite Application Manager for SOA utilizes the foundation services of IBM Tivoli Monitoring services.

Find more information about IBM Tivoli Composite Application Manager at:

<http://www.ibm.com/software/tivoli/solutions/application-management/>

3.3 Roles of Business Process Management

Figure 3-2 highlights the participants and necessary steps to develop, deploy, and continuously improve an IBM Business Process Management solution over time.

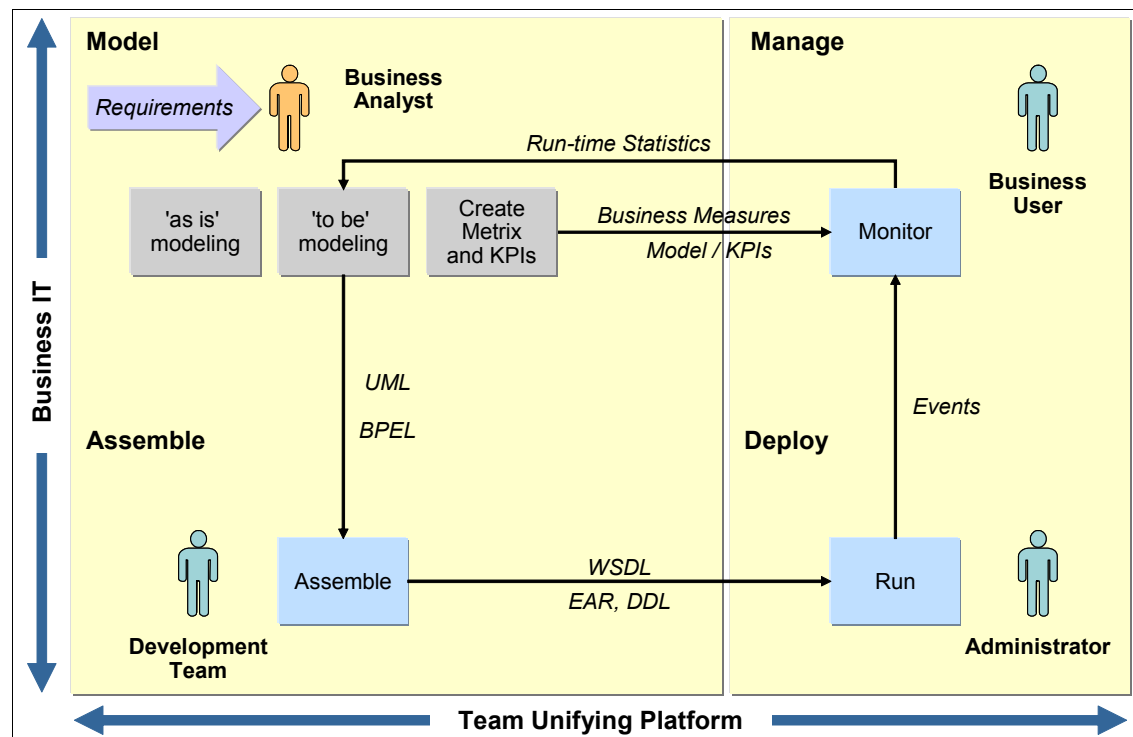


Figure 3-2 Roles-based approach

Figure 3-2 on page 39 shows these roles:

- ▶ Business analyst
 - Typically the first step is to document and capture a high-level abstraction of the currently implemented business process in WebSphere Business Modeler. This model is referred to as the “As-Is” model and may be a totally manual process without any IT automation.
 - The next step is to develop a “To-Be” model by analyzing, simulating, and optimizing the As-Is situation. Simulation allows the assessment of the performance of a process, just as in a real-life work environment. It generates statistics and reports to pinpoint potential areas of improvement. The organization should be in agreement about implementation of the optimized To-Be process model.
- ▶ Process specialist
 - An IT-aware business process specialist or architect will finalize the To-Be model to hand over to IT and will describe the business performance aspects of a business model as a business measures model (observation model). This model is used by WebSphere Business Monitor to create custom dashboards for business users. When the business measures model is exported, the process model is exported as well.
- ▶ IT development and IT architects
 - The process model is handed over to IT development for refinement and IT integration within WebSphere Integration Developer. IT architects typically define the component structure and the reuse of assets for the new overall composite application solution.
 - Development teams build and assemble services and components required for the execution of a defined process. WebSphere Integration Developer contains an integrated test environment to test components and the assembled application.
- ▶ WebSphere administrators and IT production
 - Development hands over the tested composite application with its artifacts to WebSphere Process Server administrators. They deploy the process to the runtime environment and the business measures model to the WebSphere Business Monitor.
- ▶ Business users
 - WebSphere Business Monitor provides role-based custom dashboards for the interests of different business users for observing the status and business goals of a business process in real time. In addition, users can examine and analyze historical information and perform trending analysis.

- ▶ Business analysts
 - Business analysts focus on further process improvements and new business opportunities. WebSphere Business Monitor can help them to examine and analyze historical information, and even perform trending analysis on runtime statistics. Runtime statistics can be imported back into WebSphere Business Modeler to run simulations based on real runtime average values.



Process for applying SOA scenarios

A process for selecting SOA scenarios and related reusable patterns assets has been documented in *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240. This process can be used by specialists in pre-sales and post-sales roles, and by architects in a customer-consulting engagement.

In this process, generic use cases have been designed to be representative of SOA capabilities that are possible with the software used in SOA scenarios. The process includes a selection table that maps the generic use cases to the appropriate SOA scenario. When the SOA scenario is identified, the user can reference the related reusable patterns assets to help accelerate the solution design.

This chapter offers an overview of the SOA scenario selection process and discusses the generic use cases that lead to the Business Process Management scenario.

This chapter contains the following topics:

- ▶ Process for using SOA scenarios and patterns
- ▶ Applying the process

4.1 Process for using SOA scenarios and patterns

This section gives an overview of the process for using SOA scenarios and patterns. This process is discussed in detail in *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240.

Figure 4-1 depicts the high-level process for applying the SOA scenarios and related patterns.

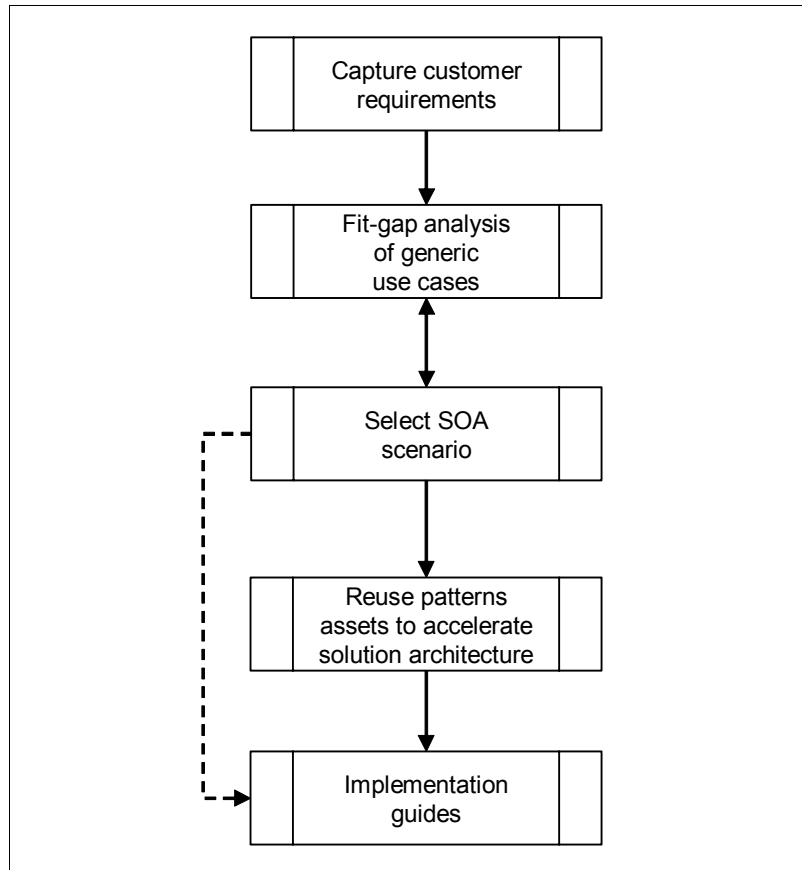


Figure 4-1 Process for using SOA scenarios and patterns

The following steps briefly describe each phase of the high-level process depicted in Figure 4-1 on page 44:

1. Capture the customer requirements

Collection of the customer requirements is a standard activity for development methodologies (for example, RUP or IBM Method). This includes an initial

context, a vision of where the customer wants to be, and use cases that represent the refined customer requirements.

2. Fit-gap analysis of generic use cases

The business analyst or architect evaluates which of the generic use cases are representative of the defined customer requirements (use cases).

Note: The upward-pointing arrow in Figure 4-1 on page 44 between *Select the SOA scenario* and *Fit-gap analysis* is a special case where the user already owns the software representative of an SOA scenario and wants to know how to leverage the SOA capabilities of the software. As is the case in this book, the user knows the SOA scenario and will look at what generic-use cases are possible in the SOA scenario selection table. (See Table 4-1 on page 46.)

3. Select the SOA scenario

The business analyst or architect can use the SOA scenario selection table (see Table 4-1 on page 46), which lists the generic uses cases and the SOA scenario that enables the use case.

After you have select the SOA scenario, you have two options of where to go next:

- Reuse patterns assets to accelerate the solution architecture

This option is appropriate if you are interested in reusable assets for the architecture and design of the solution. This option also leads to the implementation guides.

or

- Select the implementation guide (model, assemble, deploy, manage)

This option is appropriate if you are interested in a working example that demonstrates how to implement the scenario (model, assemble, deploy, manage).

4. Reuse patterns assets to accelerate the solution architecture

We have mapped the reusable assets found in the Patterns for e-business for each of the SOA scenarios. The reusable assets are used to accelerate the solution architecture.

5. Select the implementation guide (model, assemble, deploy, manage)

After you have identified the SOA scenario (and, optionally, the corresponding patterns), you can go to the appropriate implementation guide. The implementation guides provide detailed examples demonstrating how to implement selected use cases for the SOA offering. This includes

implementation details for each phase of the lifecycle using the solution offering software (model, assemble, deploy, manage).

4.1.1 SOA scenario selection table

A set of generic use cases has been developed to be used as selection criteria to determine the appropriate SOA scenario. Table 4-1 lists the generic use cases and corresponding SOA scenario that fulfill the use case requirement.

Table 4-1 SOA scenario selection criteria

Generic use cases selection criteria	Service Creation	Service Connectivity	Interaction and Collaboration Services	Business Process Management	Information as a Service
U1: Reuse existing or create new application logic as a service within the enterprise	X				
U2: Reuse existing or create new application logic as a service beyond the enterprise	X				
U3: Point-to-point integration of enterprise applications using services	X				
U4: Point-to-point integration of intra-enterprise applications using services	X				
U5: Enable users to invoke services simply	X				
U6: Enable loose coupling of service consumers and providers using static routing		X			
U7: Enable loose coupling of service consumers and providers using dynamic routing based on standards-based protocols		X			
U8: Enable loose coupling of service consumers and providers using advanced dynamic routing and diverse protocols		X			
U9: Improve an existing business process flow through business process and policy modeling and simulation				X	
U10: Implement a new business process flow.				X	
U11: Analyze existing business process flow using monitoring				X	

Generic use cases selection criteria	Service Creation	Service Connectivity	Interaction and Collaboration Services	Business Process Management	Information as a Service
U12: Allow single sign-on access to different services			X		
U13: Personalize information based on user profile			X		
U14: Allow users to create and manage content			X		
U15: Allow users to access services through client devices	X		X		
U16: Allow users to perform information inquiries					X
U17: Populate information					X
U18: Allow users seamless access to diverse data sources					X

Because this book focuses on the Business Process Management scenario, next we look specifically at the use cases that lead you to this scenario. If these use cases do not apply to your situation, consider reading *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240 to explore the remaining use cases.

Generic use cases for the Business Process Management scenario

The following generic use cases have been identified as leading to the Business Process Management scenario:

- U9: Improve an existing business process flow through business process and policy modeling and simulation

A customer who has an existing business process flow consisting of automated and manual tasks wishes to improve the efficiency of this business process. In order to do this, the customer fully documents the business process, then runs simulated workloads through the process to identify bottlenecks, inefficiencies, and associated costs. This information is used to create a new, more efficient refined business process.

- U10: Implement a new business process flow

A customer wants to create a new business process flow implementation. This implementation either is based on a completely new business process or

is a refinement of an existing business process that has previously been modeled and simulated.

The business process implementation is designed by a business analyst, and an implementation is created by an enterprise architect and a development team. The business process implementation can include automated activities exposed as services, and manual activities. These activities may reside in or outside of the enterprise.

- U11: Analyze existing business process flow using monitoring

A customer with an existing business process flow implementation in place wishes to monitor the execution of the process instances for this business process. The data captured for monitoring can use key performance indicators, dimension analysis, and situation events to analyze the process instance data.

When the monitored data has been analyzed, it can be used to drive continual business improvement, resulting in a refined business process model and business process implementation.

4.2 Applying the process

This book includes the following chapters to help you apply the process for selecting an SOA scenario:

- Chapter 5, “Patterns for e-business and Business Process Management” on page 49

The process for selecting an SOA scenario has an optional step to reuse patterns assets to accelerate the solution architecture. If you follow this step, then you will find this chapter of interest. This chapter introduces the Patterns for e-business, then discusses in detail the patterns assets that are relevant to the Business Process Management scenario. This includes IBM product mappings.

- Chapter 10, “Business scenario and solution architecture” on page 153

This chapter introduces the scenario used in this book: ITSO Speedy Rentals. In this chapter, 10.3, “Solution architecture” on page 159 shows how to apply the process for selecting an SOA scenario to ITSO Speedy Rentals. By using this process, we determine that the Business Process Management scenario is most applicable to ITSO Speedy Rentals, and we can identify the relevant Patterns for e-business assets.



Patterns for e-business and Business Process Management

This chapter introduces the IBM Patterns for e-business and describes the patterns from the IBM Patterns asset catalog that apply to the Business Process Management scenario.

We have organized the sections by the Business or Integration pattern. The corresponding Application patterns, Runtime patterns, and Product mappings are included in the Business and Integration pattern sections.

This chapter contains the following sections:

- ▶ Introduction to the Patterns for e-business
- ▶ Application Integration pattern
- ▶ Extended Enterprise pattern

Note: For more detailed information about the IBM Patterns for e-business, including the latest product mappings, see:

<http://www.ibm.com/developerworks/patterns>

5.1 Introduction to the Patterns for e-business

This section provides an introduction to the Patterns for e-business.

The role of the IT architect is to evaluate business problems and build solutions to solve them. The architect begins by gathering input about the problem, then develops an outline of the desired solution and considers any special requirements that should be factored into that solution. The architect then takes this input and designs the solution, which can include one or more computer applications that address the business problems by supplying the necessary business functions.

To improve the process over time, the experience of the IT architects should be captured and reused in such a way that future engagements can be made simpler and faster. This can be done by capturing knowledge gained from each engagement and using it to build a repository of assets. IT architects can then build future solutions based on these proven assets. Reusing proven assets saves time, money, and effort and helps ensure delivery of a solid, properly architected solution.

The IBM Patterns for e-business help facilitate this reuse of assets. Their purpose is to capture and publish e-business artifacts that have been used, tested, and proven to be successful. The captured information is assumed to fit the majority, or 80/20, situation.

5.1.1 Patterns for e-business layered asset model

The Patterns for e-business approach enables architects to implement successful e-business solutions through the reuse of components and solution elements from proven successful experiences. The Patterns approach is based on a set of layered assets that can be exploited by any existing development methodology. These layered assets are structured in a way that each level of detail builds on the last and include:

- ▶ *Customer requirements* ultimately define the selected solution. We have developed a series of generic use cases that are representative of common SOA scenarios. The customer requirements are analyzed and when appropriate mapped to the generic use cases. The generic use cases are used as selection criteria to identify the appropriate SOA scenario and reusable patterns assets.
- ▶ *Business patterns* identify the interaction between users, businesses, and data.
- ▶ *Integration patterns* tie multiple Business patterns together when a solution cannot be provided based on a single Business pattern.

- ▶ *Composite patterns* represent commonly occurring combinations of Business patterns and Integration patterns.
- ▶ *Application patterns* provide a conceptual layout that describes how the application components and data within a Business pattern or Integration pattern interact.
- ▶ *Runtime patterns* define the logical middleware structure that supports and Application pattern. Runtime patterns depict the major middleware nodes, their roles, and the interfaces between these nodes.
- ▶ *Product mappings* identify proven and tested software implementations for each Runtime pattern.
- ▶ *Best-practice guidelines* discuss design, development, deployment, and management of e-business applications.

Figure 5-1 shows these assets and their relationships to each other.

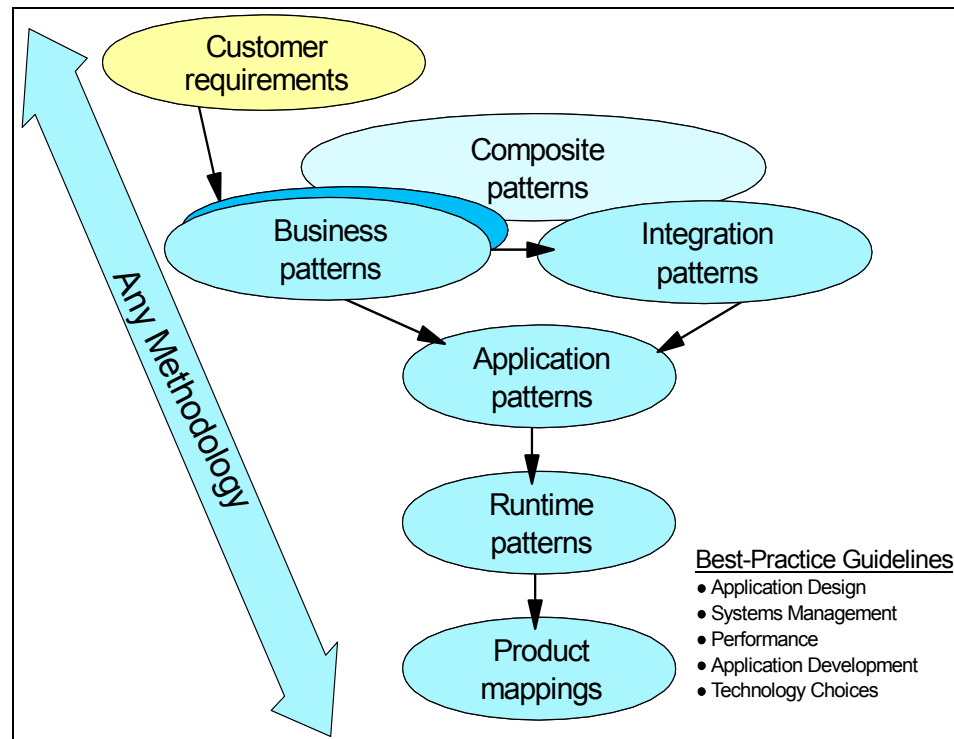


Figure 5-1 The Patterns for e-business layered asset model

5.1.2 Summary description of patterns

Table 5-1 provides a brief description of the Business, Integration, and Composite patterns.

Table 5-1 Summary description for the Business, Integration and Composite patterns

Pattern	Description
Self-Service	The Self-Service business pattern captures the essence of direct interactions between interested parties and a business. Interested parties include customers, business partners, stakeholders, employees, and all other individuals with whom the business intends to interact.
Collaboration	The Collaboration business pattern enables interaction and collaboration among users. This pattern can be observed in solutions that support small or extended teams who need to work together in order to achieve a joint goal.
Extended Enterprise	The Extended Enterprise business pattern covers interactions between applications from different enterprises. This pattern can be observed in solutions that implement programmatic interfaces to connect inter-enterprise applications
Information Aggregation	The Information Aggregation business pattern exists in e-business solutions that enable users to access and manipulate data that is aggregated from multiple sources. This Business pattern captures the process of taking large volumes of data, text, images, video, and so on, and using various user-controlled tools to extract useful information from them.
Application Integration	The Application Integration pattern serves to integrate multiple Business patterns and to integrate applications and data within an individual Business pattern.
Access Integration	The Access Integration pattern gives users a single, consistent, and seamless access mechanism to various applications that would otherwise require the use of several different access mechanisms.
Portal Composite	The Portal composite pattern leverages various mechanisms (such as content management, user interface formatting and display, data aggregation) to bring together the appropriate information and existing systems to serve the goals of the business.

5.1.3 Patterns for e-business Web site

The layers of patterns, along with their associated links and guidelines, enable the architect to start with a problem and a vision for the solution, then find a pattern that fits that vision. In order to navigate from the top down from one level to another, a decision matrix will be provided to assist the architect in making the right decision.

Then, by drilling down using the patterns process, the architect can further define the additional functional pieces that the application needs to succeed. Finally, the architect can build the application using coding techniques that are outlined in the associated guidelines.

The Patterns Web site provides an easy way to navigate through the layered Patterns assets to determine the most appropriate assets for a particular engagement. For easy reference, see the Patterns for e-business Web site at:

<http://www.ibm.com/developerWorks/patterns/>

5.2 Application Integration pattern

Integration patterns may be used to combine different Business patterns that are involved in one solution, or to support more advanced features inside one Business pattern.

Application Integration patterns are a type of Integration pattern that refer to the integration of data and applications within a Business pattern or integration between patterns.

This section introduces the Serial Process and Parallel Process Application Integration patterns.

Note: For more detailed information, refer to:

- ▶ Application Integration pattern on the Patterns for e-business Web site
<http://www.ibm.com/developerworks/patterns/application/index.html>
- ▶ The IBM Redbook *Patterns: Building Serial and Parallel Processes for IBM WebSphere Process Server V6*, SG24-7205

5.2.1 Serial Process

The Serial Process pattern enables the orchestration of multiple services to form a coherent business process. The Serial Process is navigated sequentially. This section describes the Application and Runtime patterns of the Serial Process, shows a sample product mapping, and discusses the variation of the Serial Process called Serial Workflow, which adds human intervention to the pattern.

Application pattern

The Serial Process application pattern, shown in Figure 5-2 on page 55, supports the sequential execution of business services hosted by a number of target applications. The source application initiates a serial business process.

The primary business driver for this pattern is composing a set of business services into an end-to-end business process. From an IT perspective, the key driver is improving the flexibility and responsiveness of IT by externalizing the process flow logic from individual applications.

The Serial Process application pattern is divided into three logical tiers:

- ▶ The Source Application tier represents one or more applications that are interested in interacting with the target applications.
- ▶ The Serial Process Rules tier supports most of the services provided by the broker tier in the Broker application pattern, including routing of requests, protocol conversion, message broadcasting, and message decomposition and recomposition. It also supports the separation of business process flow logic from individual application logic.

The process logic is governed by the serial process rules that define execution rules for each target application, together with control flow and data flow rules. It also may include any necessary adapter rules. The combination of these process execution rules are stored in read-only databases.

This externalization of process flow logic is essential for the implementation of a flexible and responsive IT environment that can respond quickly to changing business needs. It also makes it possible to compose new end-to-end processes by combining different business services provided by different applications. Finally, this tier uses a work-in-progress (WIP) database to store the intermediate results from the execution of different process steps.

- ▶ The Target Application tier represents new, modified, or unmodified existing applications that implement the necessary business services.

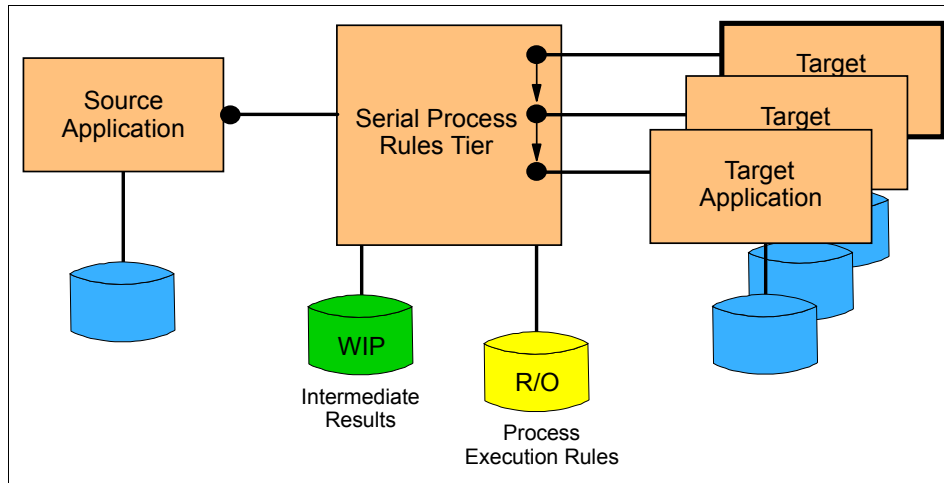


Figure 5-2 Serial Process application pattern

The Serial Workflow variation of the Serial Process application pattern, shown in Figure 5-3 on page 56, extends the basic Serial Process by including interaction with people as steps in the business process.

All of the business and IT drivers for the Serial Process application pattern apply to this variation as well. In addition, the need exists to support human interaction and intervention within the process flow. This is presented by the Human Interaction Resolution tier.

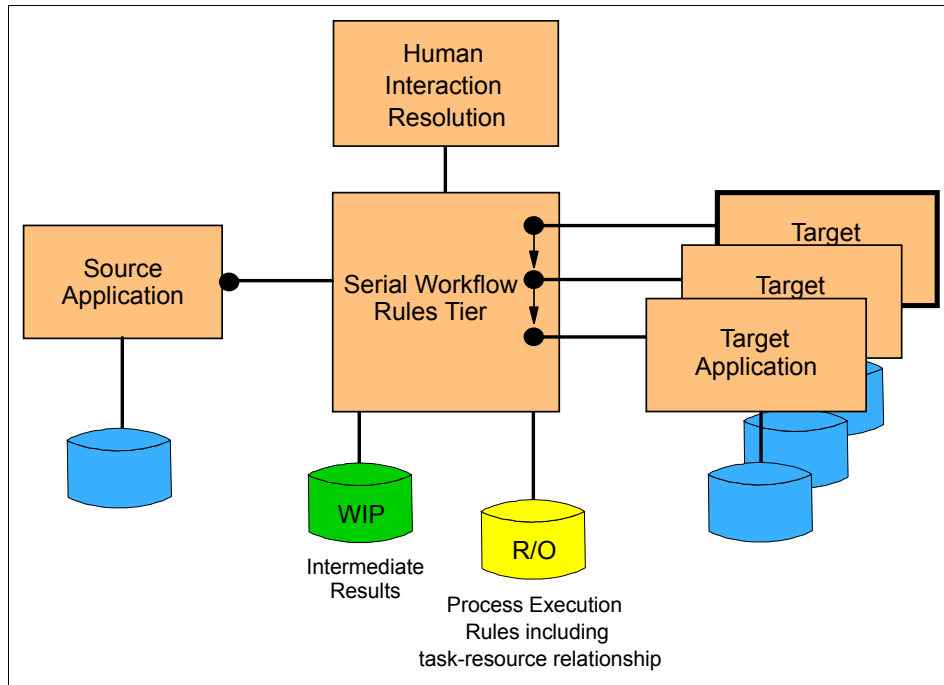


Figure 5-3 Serial Workflow variation

Runtime pattern

The Runtime pattern shown in Figure 5-4 on page 57 represents a basic topology for the implementation of the Serial Process application pattern. It can be further enhanced by clustering the key nodes to improve availability characteristics. It contains the following nodes:

- Process Manager node

The Process Manager node contains the process flow execution engine. It provides the capability for model-driven business process automation. It also enables tracking by leveraging the process execution rules stored in the associated database.

- App Server/Services node

The App Server/Services nodes represent applications running in a managed application server environment. They can be providers with the ability to provide a service, or consumers, who have the ability to request a service.

- Directory and Security Services node

The Directory and Security services node supplies authentication and authorization services. It also holds the user ID and password and related

privileges. This node typically leverages Lightweight Directory Access Protocol (LDAP) based directories.

► Repository node

The Repository node provides a persistent data storage and retrieval service in support of the execution of the process flow. It holds the Process Execution Rules and the Intermediate Results from the execution of certain activities in the context of an end-to-end process flow. The implementation of this node may involve several persistent data technologies (such as a database management system (DBMS) and flat file) for the different data types. In some cases, non-persistent storage is used to store the Intermediate Results.

► Connector node

A connector node is used to provide connectivity between:

- The App Server/Services node and the Process Manager

This enables a source application to communicate with a Process Manager. Process Managers typically support multiple connector types.

- The Process Manager and an App Server/Services node

This enables a Process Manager to communicate with a target application. Process Managers typically offer multiple connector types.

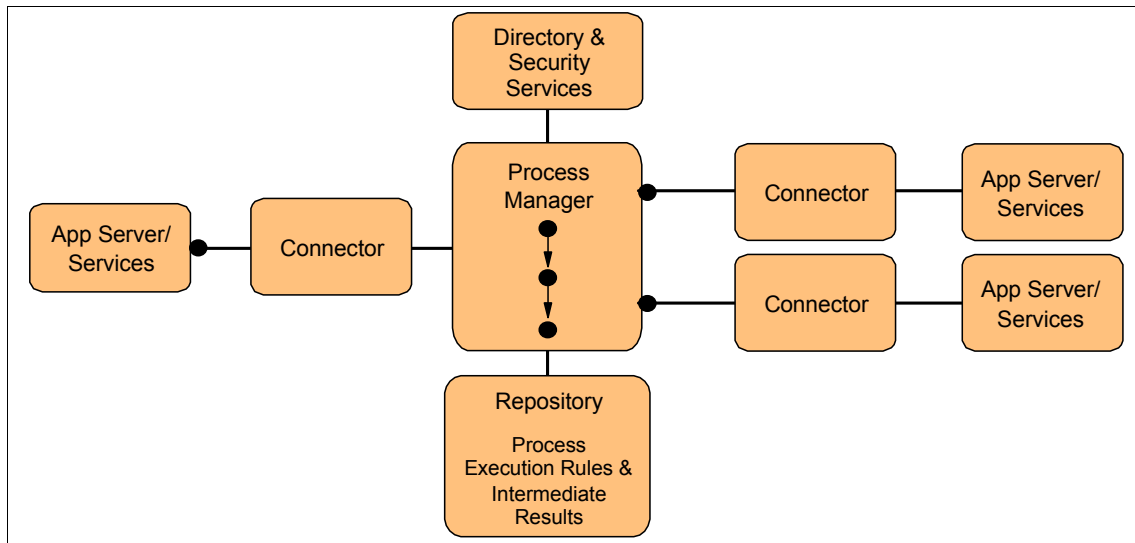


Figure 5-4 Serial Process runtime pattern: Generic profile

The Serial Workflow runtime variation is similar to the Serial Process runtime pattern except that we need to deploy nodes to handle human interaction. The Runtime pattern shown in Figure 5-5 on page 58 represents the basic topology

for the implementation of the Serial Workflow variation. It is similar to the Serial Process Runtime pattern, but has the addition of the Staff Worklist Adapter, which is accessed by the Workflow Manager.

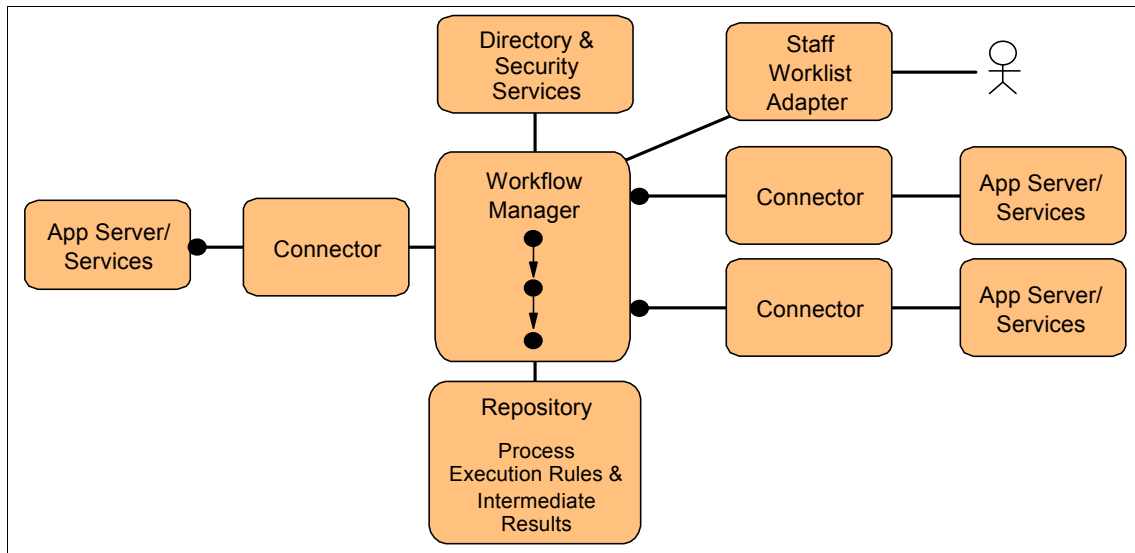


Figure 5-5 Serial Workflow runtime pattern

Product mapping

Figure 5-6 on page 59 shows a product mapping for the Serial Process. In this product mapping, the source application running on WebSphere Process Server uses Web services to invoke the automated process instance implemented by the Process Manager node. Because instructions for finding the service are in the Web Services Description Language (WSDL) files managed by WebSphere Process Server, it is also responsible for Directory and Security Services. The repository uses the IBM Cloudscape™ database provided with WebSphere Process Server. We do not recommend using this database in a product environment.

The Process Manager uses the Service Component Architecture to invoke Web services provided by the target applications running on WebSphere Process Server. We could have chosen WebSphere Application Server to implement the target application services, or for that matter, any application server that provides Web services, such as Microsoft .NET.

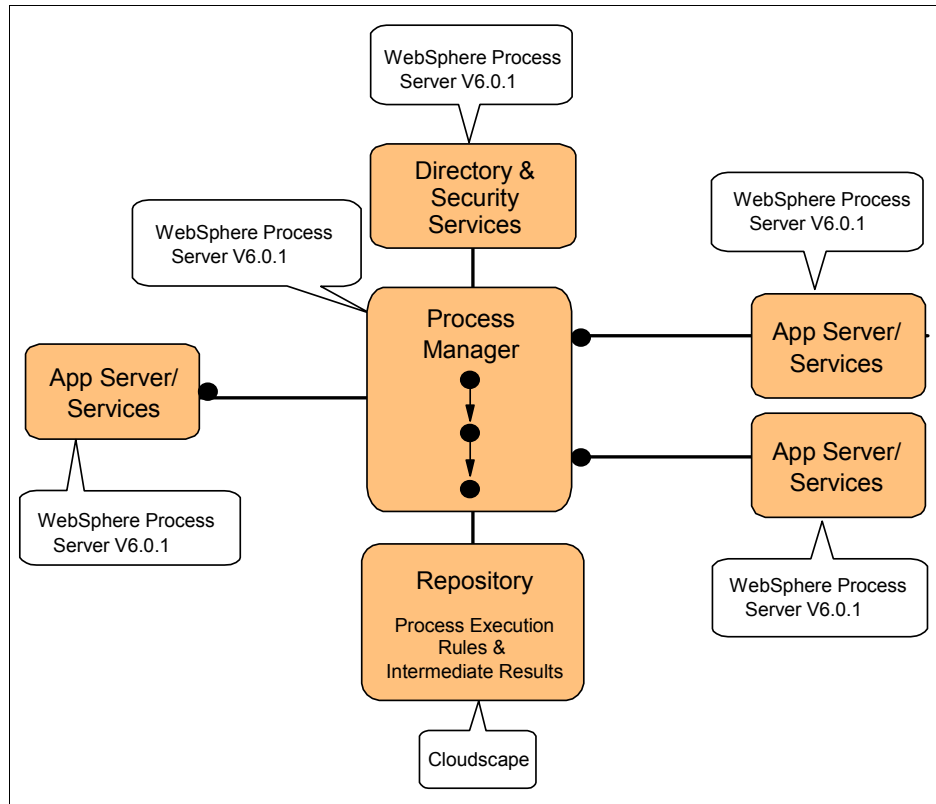


Figure 5-6 Serial Process: WebSphere Process Server product mapping

A product mapping for the Serial Workflow variation is similar to Figure 5-6 with the addition of a Staff Worklist Adapter node, which would be implemented by one of the user registries supported by WebSphere Process Server.

5.2.2 Parallel Process

The Parallel Process pattern differs from the Serial Process in that it defines multiple simultaneous paths of execution. As with the Serial Process, the Parallel Process pattern also has one variation: the Parallel Workflow variation, which adds human intervention.

Application pattern

The Parallel Process application pattern, shown in Figure 5-7, adds concurrency to the Serial Process application pattern. The process orchestration may include paths that lead to parallel invocation of target application services.

All business and IT drivers listed under the Serial Process application pattern apply to this Application pattern as well. The additional driver is the need to reduce cycle time by executing parts of the process in parallel.

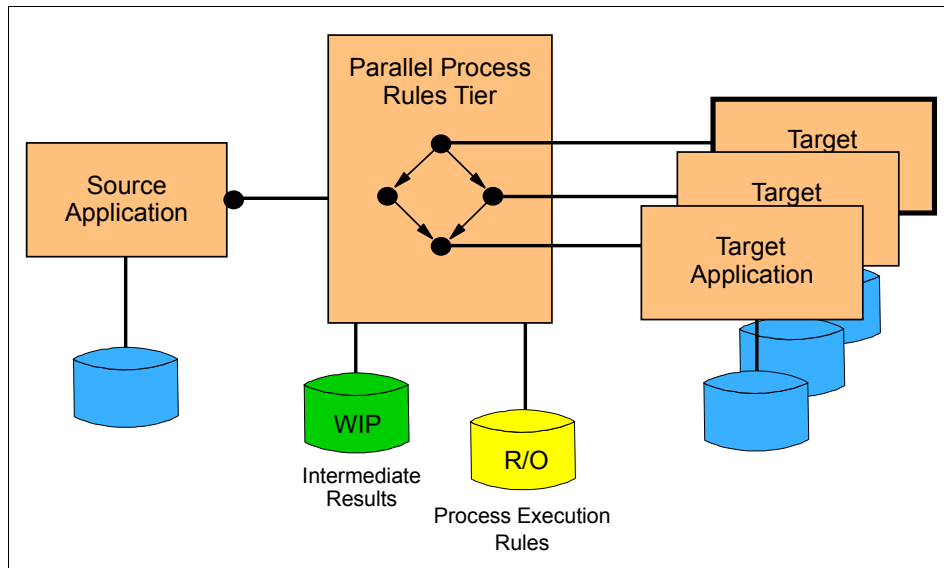


Figure 5-7 Parallel Process application pattern

The Parallel Workflow variation of the Parallel Process application pattern, shown in Figure 5-8, extends the basic parallel process orchestration capability by supporting human interaction for completing certain process steps.

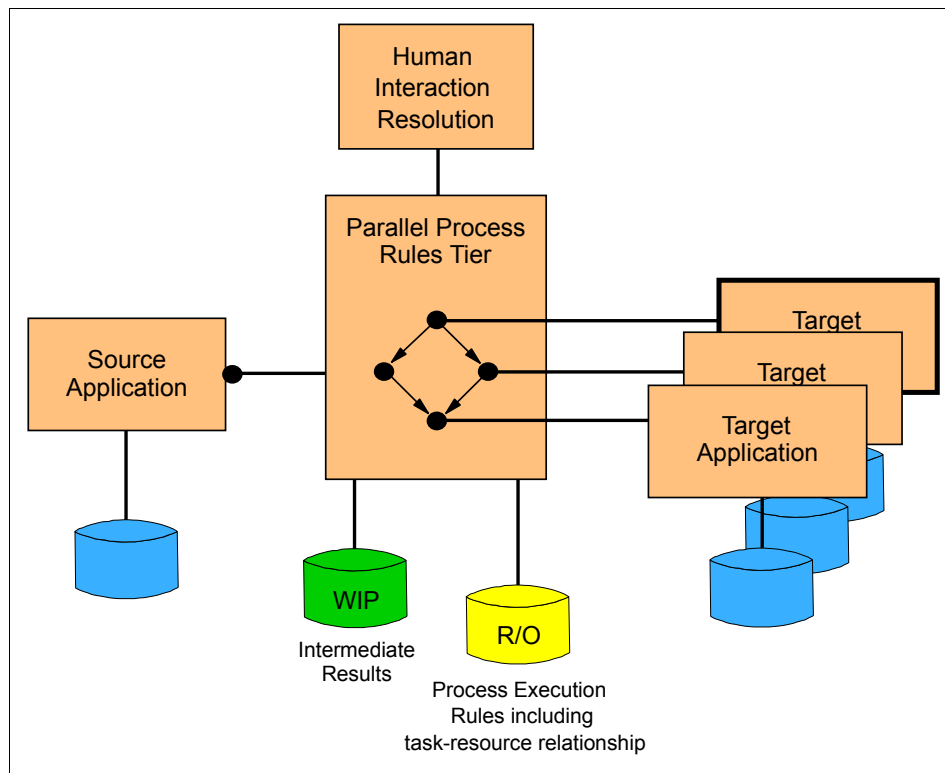


Figure 5-8 Parallel Workflow variation

Runtime pattern

The Runtime pattern shown in Figure 5-9 is similar to the Runtime pattern for the Serial Process, except for the parallel capabilities of the Process Manager node. See “Runtime pattern” on page 56 for descriptions of each node in this pattern.

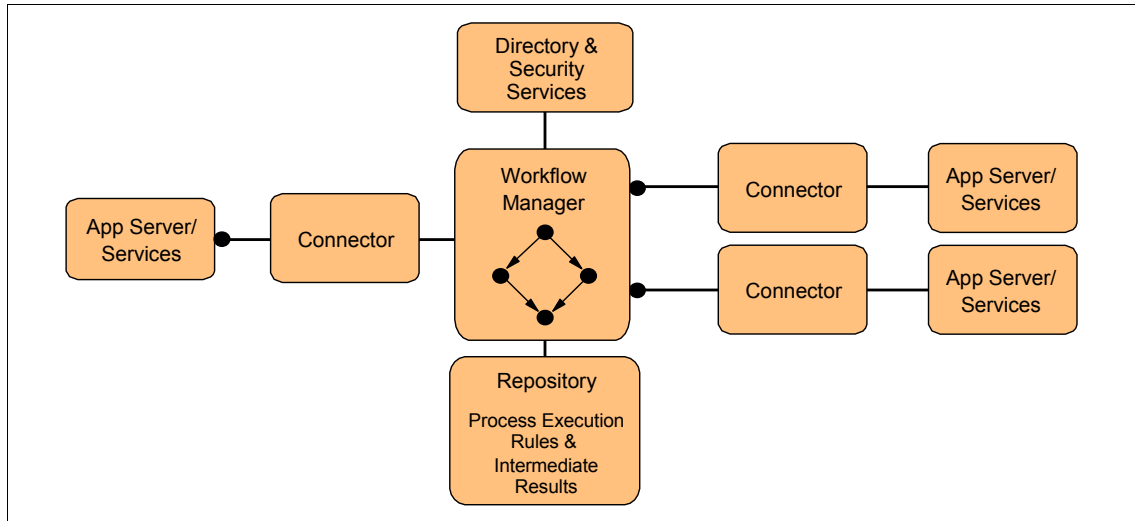


Figure 5-9 Parallel Process runtime pattern

The Parallel Workflow variation (Figure 5-10) adds a Staff Worklist Adapter, which is used for communication with human tasks.

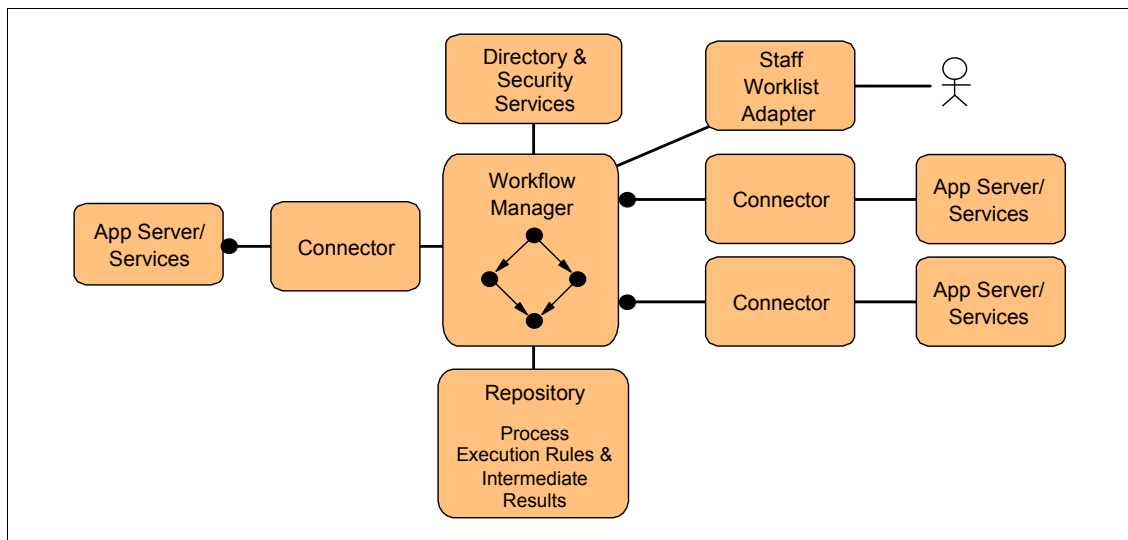


Figure 5-10 Parallel Workflow runtime pattern

Product mapping

Figure 5-11 shows a product mapping for the Parallel Process pattern. Unlike the Serial Process pattern, this product mapping uses IBM DB2® Universal Database™ as the Repository node. WebSphere Process Server will not truly execute multiple threads of execution in parallel if a Cloudscape database is used.

One of the App Server/Services nodes is implemented by DB2 Universal Database. Connectivity is through a Java™ Database Connectivity (JDBC™) adapter, which in this case is part of WebSphere Process Server.

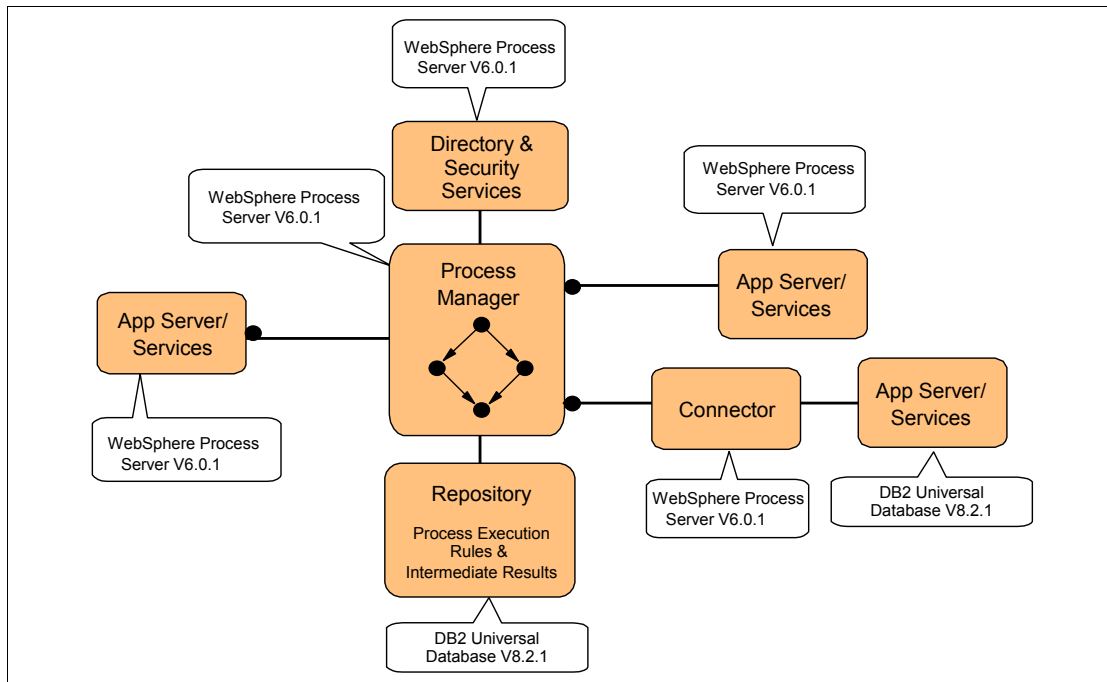


Figure 5-11 Parallel Process: WebSphere Process Server product mapping

Figure 5-12 shows a Parallel Workflow variation product mapping. In it, the Workflow Manager invokes processes that require human interaction. The human interactions in the process flow are resolved and handled by the Staff Worklist Adapter, which in this case is the Business Process Explorer included in WebSphere Process Server.

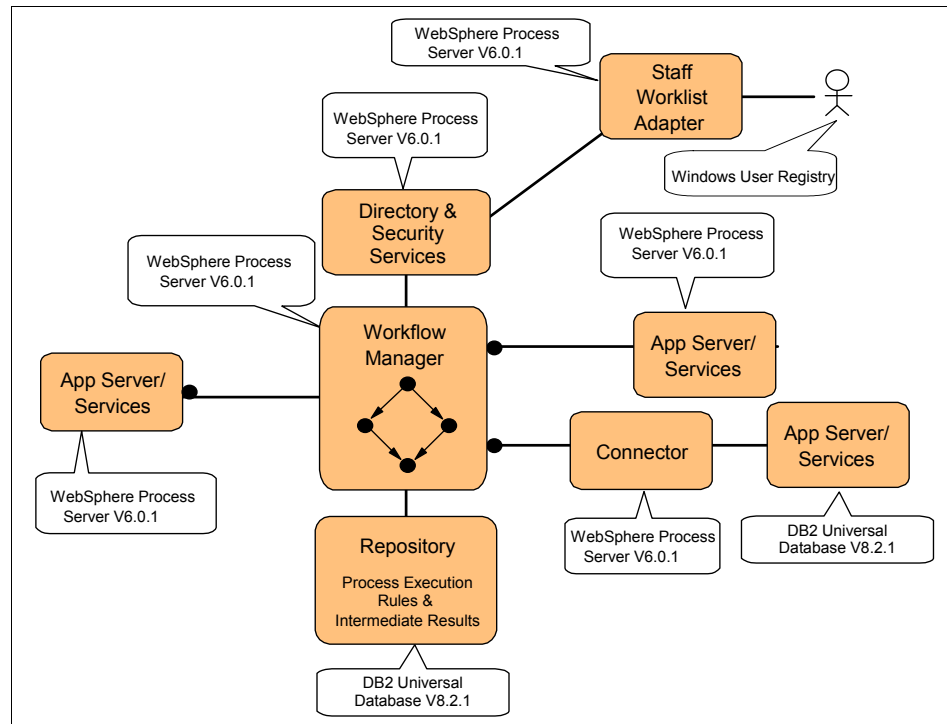


Figure 5-12 Parallel Workflow variation: WebSphere Process Server product mapping

Another good option for the Staff Worklist Adapter is the Process Portal component of WebSphere Portal. WebSphere Process Server includes plug-ins for different user directories to store the users and roles that are necessary for human interaction. We selected the Windows® User Registry. Another popular choice, and one more likely to be used in production, is a Lightweight Directory Access Protocol (LDAP) directory such as IBM Tivoli Directory Server.

5.3 Extended Enterprise pattern

The Application Integration patterns automate business functionality within an enterprise. When the same requirements have to be fulfilled between separate

enterprises, you can use the Extended Enterprise business pattern. In this section we introduce the Exposed Serial Process Extended Enterprise pattern.

Note: For more detailed information, refer to the Extended Enterprise business pattern on the Patterns for e-business Web site:

<http://www.ibm.com/developerworks/patterns/b2bi/index.html>

The IBM Redbook *Patterns: Building Serial and Parallel Processes for IBM WebSphere Process Server V6*, SG24-7205, also has more information.

5.3.1 Exposed Serial Process

The Exposed Serial Process exposes a sequential business process to external enterprises. This section describes the Application and Runtime patterns for the Exposed Serial Process, provides an product mapping example, and discusses the Exposed Serial Workflow variation, which adds human interaction.

Application pattern

The Exposed Serial Process application pattern, shown in Figure 5-13 on page 66, separates the process logic from application logic that is distributed across organization boundaries. The process logic is governed by Serial Process Rules that define execution rules for each target application, together with control flow and data flow rules. It may also include any necessary adapter rules.

The primary business driver for selecting this Application pattern is composing a business process flow between partners. From an IT perspective, the key driver is improving the flexibility and responsiveness of IT by externalizing the process flow logic from the application logic.

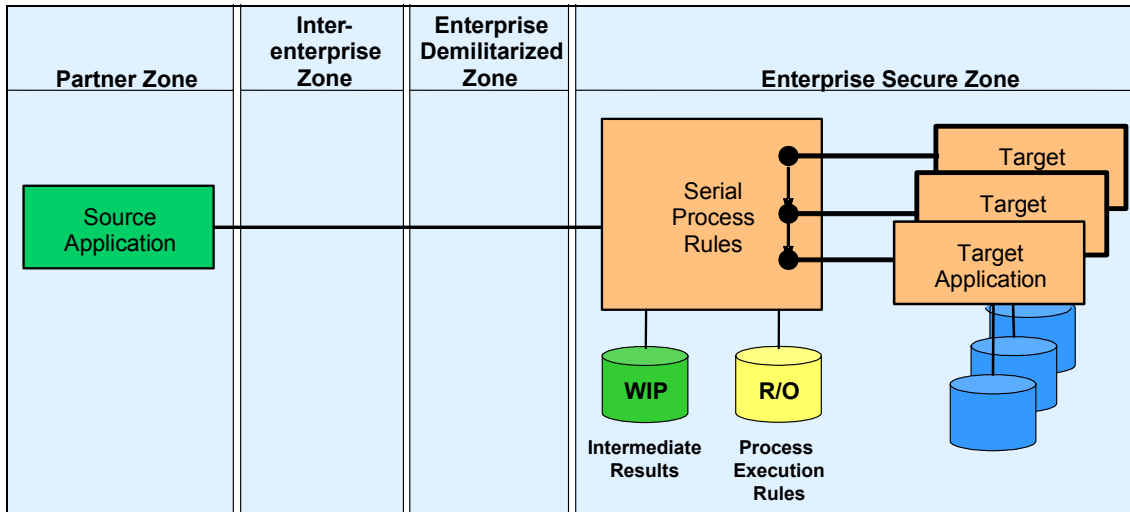


Figure 5-13 Exposed Serial Process application pattern

The Workflow variation of the Exposed Serial Process application pattern, shown in Figure 5-14 on page 67, extends the basic Serial Process orchestration capability by supporting human interaction for completing certain process steps.

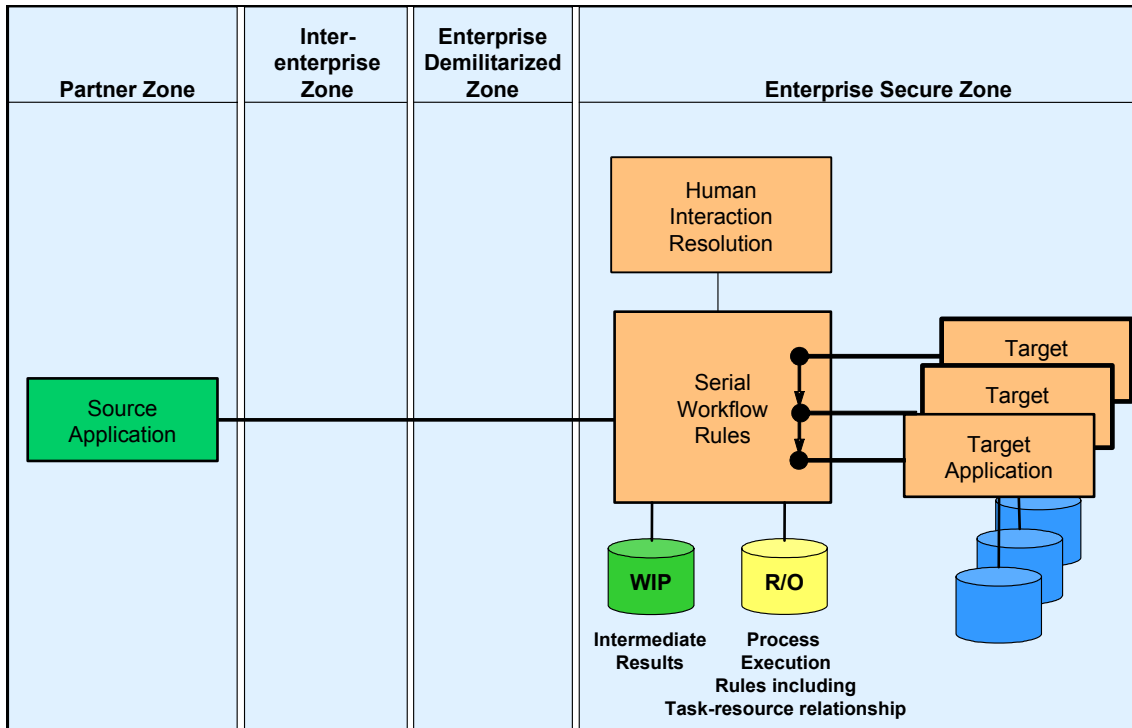


Figure 5-14 Exposed Serial Workflow variation

Runtime pattern

The Exposed Serial Process pattern is similar to the Serial Process pattern, within the Enterprise Secure Zone. Figure 5-15 on page 68 shows the Exposed Serial Process. Notice how all of the process implementation components are located in the Enterprise Secure Zone.

This pattern introduces the following zones:

- Enterprise Secure Zone

The Enterprise Secure Zone is the safe zone of the non-Extended Enterprise patterns. Security is less of a factor in this zone because it is completely within the control of the enterprise.

- Enterprise Demilitarized Zone

Named after the real-world term, and logically not too dissimilar from it, the DMZ provides a second line of defense between the Enterprise Secure Zone and the Inter-enterprise Zone. This zone is protected by the Protocol Firewall.

► Inter-enterprise Zone

The Inter-enterprise Zone is the zone outside the firewalls and where the enterprise connects to the outside world. Any systems deployed in this zone are effectively open to attack and must be capable of defending themselves. This zone is often the responsibility of a third party, either a VAN provider or an Internet service provider (ISP) in the case of the Internet.

► Partner Zone

The Partner Zone is farthest from the enterprise but cannot be dismissed as beyond the sphere of concern of those who implement the Runtime patterns. Each partner typically has their own zone structure behind their own firewalls. The firewalls must be negotiated before you can gain access to their systems.

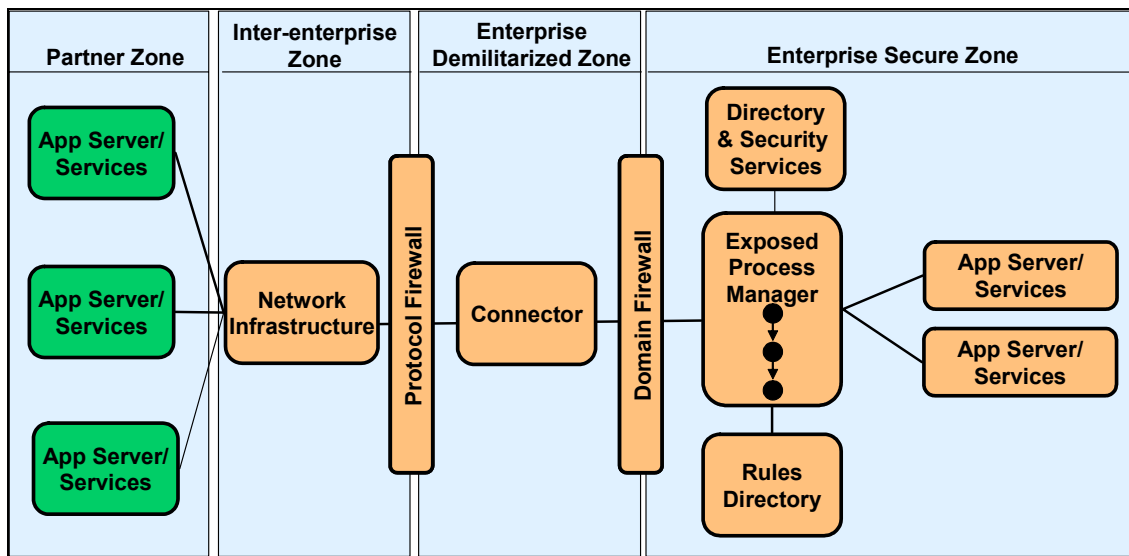


Figure 5-15 Exposed Serial Process runtime pattern

The Exposed Serial Workflow variation, shown in Figure 5-16, adds a Worklist Adapter to the pattern.

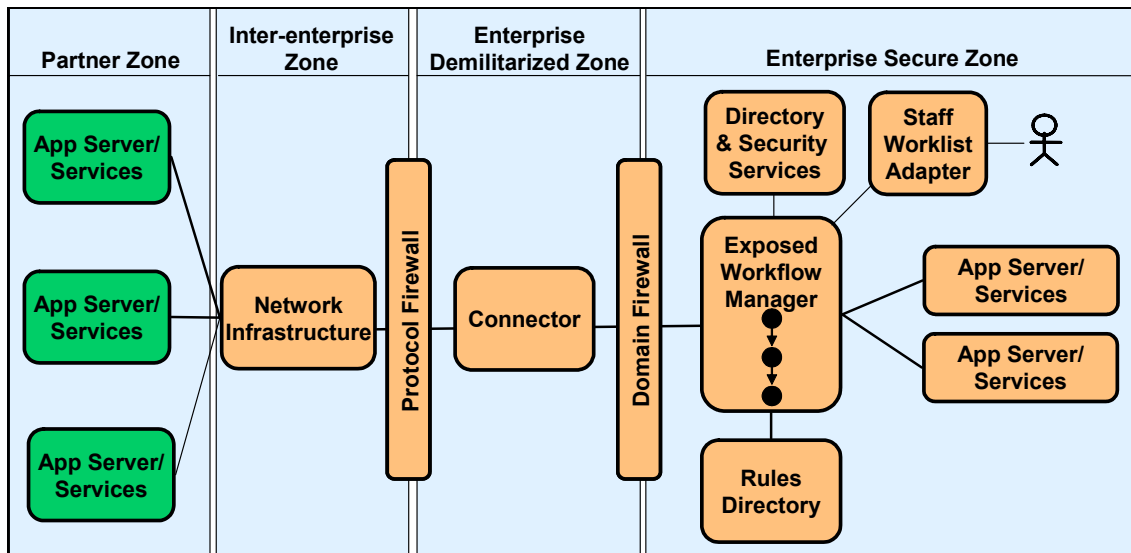


Figure 5-16 Exposed Serial Workflow runtime pattern

Product mapping

In this product mapping for the Exposed Serial Process runtime pattern, shown in Figure 5-17 on page 70, the Exposed Process Manager node is implemented using WebSphere Process Server.

We illustrate two options for implementing services in the Enterprise Secure Zone. We implement one of our target applications on WebSphere Process Server. The second target application illustrates a connection through a WebSphere Business Integration Adapter to an enterprise application, in this case SAP. As in the previous cases, the directories are also hosted on WebSphere Process Server.

An application service at the partner site invokes the automated process instance implemented by the Process Manager node over the Internet using SOAP/HTTPS. The IBM HTTP Server acts as a connector by exposing SOAP/HTTPS to partner organizations while allowing the Process Manager to process standard SOAP over HTTP (SOAP/HTTP) calls.

In the Directory and Security Services node, the service integration bus within WebSphere Process Server is configured to secure all transactions to the external Partner Zone using WS-Security integrity and confidentiality.

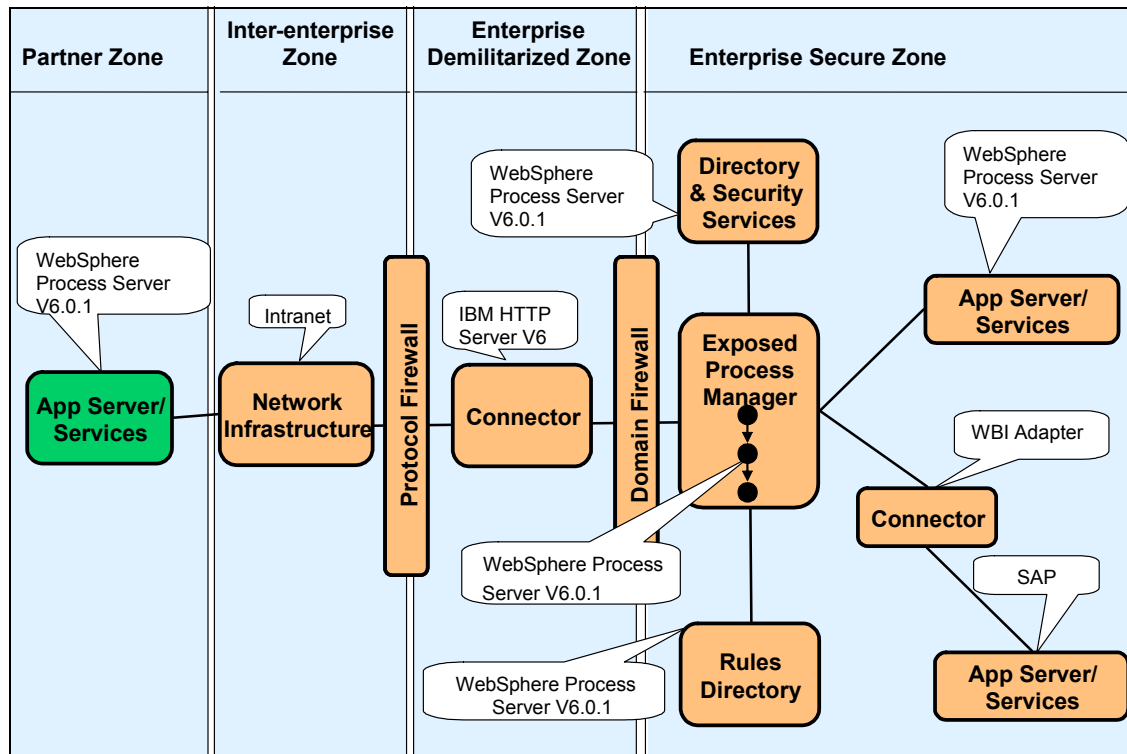


Figure 5-17 Exposed Serial Process: WebSphere Process Server product mapping



Part 2

Business Process Management product capabilities



Modeling capabilities of WebSphere Business Modeler

This chapter describes and discusses the general capabilities of WebSphere Business Modeler. We focus on modeling for execution, highlighting features and functions that are important for model translations to WebSphere Integration Developer and WebSphere Business Monitor.

In this chapter, the following topics are discussed:

- ▶ Basic business modeling and modeling modes
- ▶ Simulation
- ▶ Modeling for monitoring and business measure design
- ▶ Modeling for WS-BPEL transformation
- ▶ Web services import capability

6.1 Business value

Today enterprises are typically under pressure to reduce cost and increase revenues through rapid operational change. Successful enterprises focus on the following questions:

- ▶ How can I design a cost advantage into new business models?
- ▶ How can I design and deploy new business models ahead of competition?
- ▶ How can I scale today's operations to leverage new markets?
- ▶ How can I take cost out of my current operations?
- ▶ How can I improve quality without additional cost?

Defining and modeling business processes is a critical factor in improving business performance. Business processes are often complex because of numerous incremental changes that were made in reaction to unique and transient circumstances. Without formal process documentation and continuous management focus, these process complexities can burden an organization with unnecessary hindrances and bottlenecks. A well-constructed business process model can help locate and eliminate those hidden inefficiencies, costs, and delays.

How can WebSphere Business Modeler help your business?

WebSphere Business Modeler V6 is a powerful tool for business users with all of the functions and features needed to support the various business modeling initiatives in today's companies and organizations.

WebSphere Business Modeler V6 can help your business overcome some of these challenges in the following ways:

- ▶ Many businesses are undertaking specific process improvement initiatives where process redesign is either already underway or inevitable. The analysis and reporting, simulation, and process-comparison capabilities in WebSphere Business Modeler make it a very powerful tool for this approach.
- ▶ Accurate modeling of document processes for legal, regulatory, training or other purposes. WebSphere Business Modeler is easy to use, shares model elements, documents attachments and collaboration features.
- ▶ Modeling can help bridge the common gap that exists between Business Analysts and IT application development. WebSphere Business Modeler was designed specifically to facilitate faster and more accurate communication between the business and technology domains. WebSphere Business Modeler can create artifacts from the business model and make them available in technology development tools to reduce the overall implementation time of new business processes.
- ▶ Defining and capturing specific business-relevant key performance indicators is viable for long-term business agility. One key feature of WebSphere

Business Modeler is its ability to create and document business criteria of interest for monitoring after process implementation and deployment.

Why do companies model their business processes?

Historically there have been many reasons why companies start to model their business processes with modeling tools. Modeling initiatives in companies typically fall at least into one of the following categories:

- ▶ Modeling for compliance and documentation
 - Document processes are used by a business to understand the business process.
 - Companies use output for training, collaboration, and documentation requirements for compliance regulations (Sarbanes-Oxley, Basel II).
 - Linkage to real-time monitoring provides a feedback mechanism for reporting requirements needed for compliance.
- ▶ Modeling for redesign
 - Document both the current state and future state business process and the comparison to determine return on investment (ROI) analysis.
 - Six Sigma and Process Improvement are common methodologies.
- ▶ Modeling for execution
 - Document the business process requirements for a new IT solution.

6.2 Modeling capabilities with WebSphere Business Modeler

This section introduces the following modeling capabilities of WebSphere Business Modeler:

- ▶ Overview of WebSphere Business Modeler features
- ▶ Basic business modeling and modeling modes
- ▶ Simulation
- ▶ Business measures design
- ▶ Modeling for execution and monitoring
- ▶ Importing and reusing existing Web services definitions

6.2.1 Overview of WebSphere Business Modeler features

The following section highlights the features and functions of WebSphere Business Modeler V6.

Process modeling

WebSphere Business Modeler facilitates the creation of visually composed business process models by dragging elements you need into a process flow diagram. A process flow is described as a sequence of tasks and decision elements with multiple branches, linked by connectors. A process can also contain subprocesses.

WebSphere Business Modeler gives you the option to model your processes either in free-form or swimlane layout. The swimlane layout arranges elements in rows according to characteristics that you specify, such as organization unit, location, resource definition, classifier, or roles, as shown in Figure 6-1.

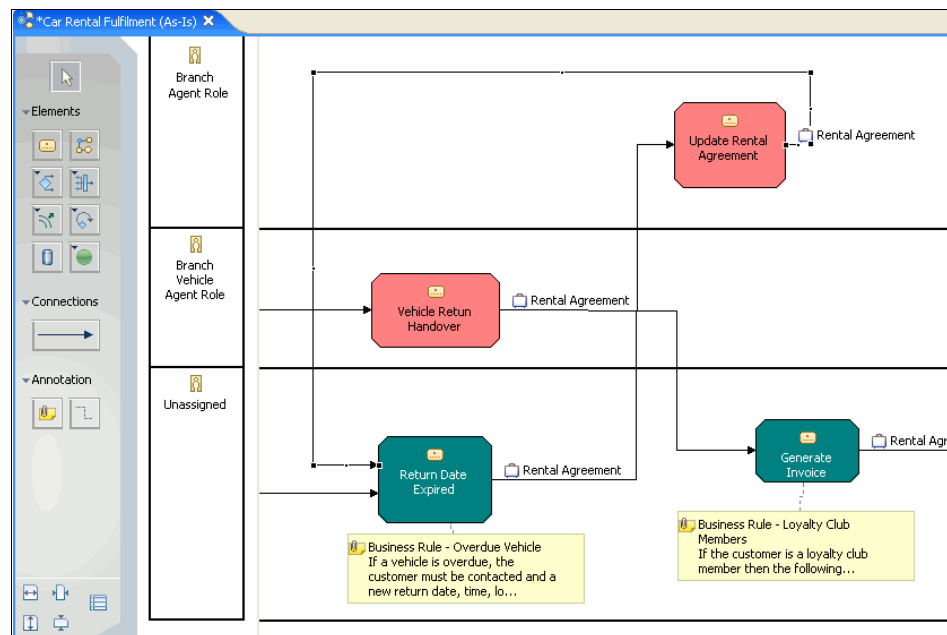


Figure 6-1 WebSphere Business Modeler in swimlane layout

WebSphere Business Modeler offers several business modeling modes to provide different levels of abstraction for business analyst, system analyst, developer, and architect. You can switch between business modeling modes. Modeling modes help to bridge the gap that exists between Business Analysts and IT application development.

Business item modeling

Process models can include any business document, work product, or commodity that is used for a particular business operation. You can model as a business item anything that is created, assembled, inspected, tested, modified,

or worked on. Business items can also undergo changes as they are passed from one step to the next in your process models. For example, a customer order could be specified as open, working, verified, and finally closed as it is passed from task to task in a particular process model.

Resource modeling

Using WebSphere Business Modeler, you can model each of your company's resources, such as employees, computers, vehicles, or electricity. Any person, equipment, or material used to perform a task or a project can be represented and used in your process models. Depending on the level of complexity you require in your process models, you can also specify roles, costs, and timetables for your resources.

Organization modeling

An organization is an entity where people cooperate to accomplish specified objectives. For example, an organization can be an enterprise, a company, or a factory. A typical company consists of one or more organizations; the larger the company, the more organizations it will normally have. With WebSphere Business Modeler, you can model any identifiable organization within your business, such as companies, divisions, or departments. You can also save your organization definitions within a project in order to reuse and revise them as your organizations evolve.

Structure modeling

Structures define the relationships between different entities in an organization. Using WebSphere Business Modeler, you can build structures to show how different types of business entities interact with one another in relationships of varying complexity, and allow for different relationships among the same divisions in the same company. For example, a Marketing department might occupy a certain position in a company's organizational chart yet receive funding from one or more different departments in the company's financial hierarchy. By modeling structures, you can define and illustrate these multiple and varied relationships in your company.

Analysis

WebSphere Business Modeler provides static analysis and dynamic analysis functions that you can use to extract targeted information from one or more elements in your modeling project. You can perform the following analyses:

- ▶ *Resource roles analysis* displays a list of resources and shows the associated roles for each resource.
- ▶ *Type hierarchy analysis* displays all occurrences of a specified organization definition within a set of structure definitions.

- ▶ *Type instance matching analysis* displays a specified type, such as a location definition or a business item, and all the instances (locations or business item instances) based on this type.
- ▶ *Activity throughput analysis* displays the productivity per time unit of activities in a process.
- ▶ *Dynamic analysis* provides information about the results of one or more process simulations based on specified attributes for a particular simulation profile. There are more than 20 types of dynamic analysis.
- ▶ *Comparative analysis* can be performed on the results of two different simulated process results. Typically you compare the As-Is process model to the To-Be process model that you want to implement.
- ▶ *Activity cost analysis* displays the average costs for activity instances including the average revenue, execution cost, idle cost, allocated resource cost, total cost, and profit.
- ▶ *Process instance resource allocation analysis* displays the resources that are allocated per task instance, the allocation duration, the resource shortage, and resource costs.
- ▶ Based on process cases, *process cost analysis* displays the average revenue, average execution cost, average idle cost, average allocated resources cost, and average profit for the process instances that match each case.
- ▶ *Process cost comparison analysis* displays process costs for two processes.

Process simulation

Process simulation provides a powerful visual animated method for analyzing a process by dynamically generating jobs and inputs that travel through one of the possible paths of the process model. Simulation output provides detailed information regarding resource utilization levels, as well as cost and cycle time calculations.

Reports

WebSphere Business Modeler provides features to generate reports on a wide range of process data. Reports summarize different aspects of your business processes and you can easily include your data in spreadsheets, presentations, or print-outs. Available reports include:

- ▶ Weighted average reports
- ▶ Simulation reports
- ▶ Documentation reports
- ▶ Procedure reports
- ▶ Analysis reports

Importing and exporting

WebSphere Business Modeler provides wizards for importing existing models and definitions, and exporting models that you create for use with other applications and workflow runtime environments.

Collaboration with team members

WebSphere Business Modeler provides a built-in capability to connect with WebSphere Business Modeler Publishing Server. WebSphere Business Modeler Publishing Server enhances WebSphere Business Modeler by providing a way to publish business processes and related business information, such as organization diagrams, to a secure Web site. This capability supports the development, documentation, and dissemination of business process models on enterprise and worldwide scales.

By publishing business processes in a Web-based format, multiple people on multiple teams around the globe can view and contribute to the development of the business processes. To accommodate the differences between viewers and contributors, WebSphere Business Modeler Publishing Server provides two stages of publishing. The first stage enables contributors to review the developing business processes and, through comments and attachments, provide the modeling team with feedback and additional information. At a chosen point, the business process models become ready for people to use as a reference in their day-to-day activities. The second stage of publishing makes the business processes available for these viewers.

6.2.2 Basic business modeling and modeling modes

The business analyst, system analyst, IT architect, and application developer vision of how a process should be captured and modeled tends to generate interesting discussion about how the process diagram is best represented. Depending on available information, business analysts can model processes at a very high level of abstraction or at a very granular level in order to visually depict all of the steps that are required to perform a particular business process. In contrast, technical resources tend to view a process in terms of what input and output both processes and tasks require, what services will be required, and what degree of human or application involvement is required to address a specific business requirement or an entire application. In order to satisfy different approaches to modeling business processes, WebSphere Business Modeler provides several different modeling modes. The mode you choose determines the level of detail you see in diagrams, dialogs, and wizards.

You can choose from the following modes:

- Basic Business Modeling (the default)

This mode focuses on creating and displaying sequence flows and filters out low-level technical details of process and information models. This mode is for resources working primarily as business analysts, business modelers, or domain experts. Functionality focuses on modeling the sequences of activities, capturing process logic, and documenting business rules. The profile does not expose the input logic and output logic attribute views. Figure 6-2 shows this mode in four-panel layout with the Project Tree, Attribute panel, Outline view, and the graphical modeling panel with its modeling elements.

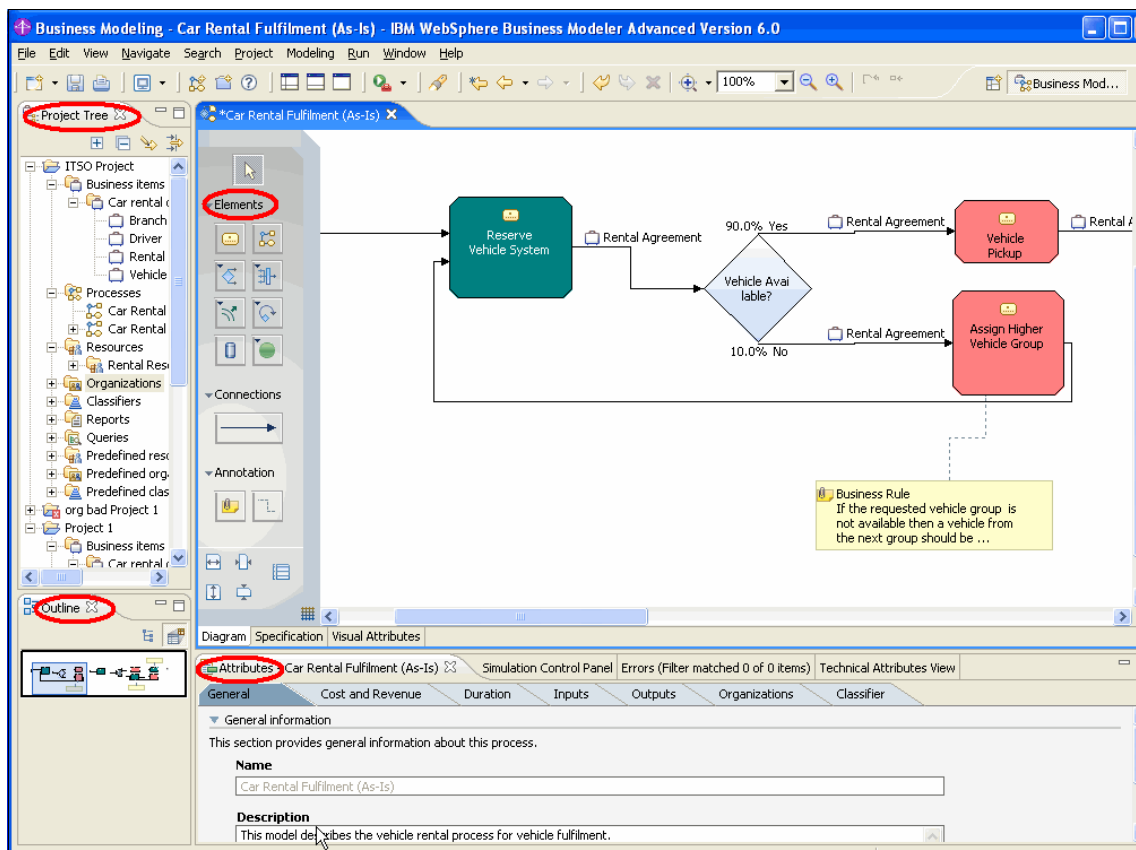


Figure 6-2 Basic modeling view in four-panel layout

► Intermediate Business Modeling

This mode gives the more technically focused user the opportunity to specify and view additional details of process and information models. The mode is geared toward business architects or business process engineers. It allows adding formal expressions, specifying cardinality, and viewing the inputs and outputs in process diagrams.

► Advanced Business Modeling

This mode facilitates the work of technically experienced personnel who prepare models that will be used as the basis for general software application development. This mode is suited for resources working as a solution architect or IT architect.

The following modes, which are specific to IBM workflow engine, can be viewed as advanced business modeling modes with specific support for process implementation leveraging the IBM process engine and IT development tooling:

– WebSphere Business Integration Server Foundation

This mode applies a set of validation rules to the Advanced Business Modeling mode to support exporting models for WebSphere Studio Application Developer Integration Edition V5.1, where an implementation of the process can be created. The validation rules constrain the model to only elements and concepts supported by WebSphere Business Integration Server Foundation V5.1.

– WebSphere MQ Workflow

This mode applies a set of validation rules to the Advanced Business Modeling mode to support exporting the model in FDL, which can then be imported into IBM WebSphere MQ Workflow Buildtime V3.5.

– WebSphere Process Server

This mode applies a set of validation rules to the Advanced Business Modeling mode to support exporting the model into SCA, BPEL, WSDL, and XSD artifacts. For details see 6.2.5, “Modeling for execution and monitoring” on page 86. The exported files are imported into WebSphere Integration Developer, where an implementation of the process can be created. The validation rules constrain the model to only elements and concepts supported by WebSphere Process Server V6. If you plan to use WebSphere Process Server as the runtime platform for models developed in WebSphere Business Modeler, you should begin modeling in this mode.

6.2.3 Simulation

Simulation enables organizations to examine how a process will perform in response to variations of inputs to the process, just as in a real-life work environment. Simulation also provides the ability to vary process input volume over time by adjusting resources and current allocations. Simulation output provides detailed information regarding resource utilization levels and the results of cost and cycle-time calculations.

When you have completed modeling a business process, you can use WebSphere Business Modeler to simulate the running of the process. Simulation enables you to assess the performance of the process, generate statistics, and pinpoint potential areas of improvement. When you make changes to a process or to other model elements (such as available resources), you can then run a new simulation and perform a comparative analysis of the before and after simulation results to quantify the effects of the changes that you made. By doing this, you can assess the costs and benefits of making changes to your business processes.

Figure 6-3 shows a simulation snapshot.

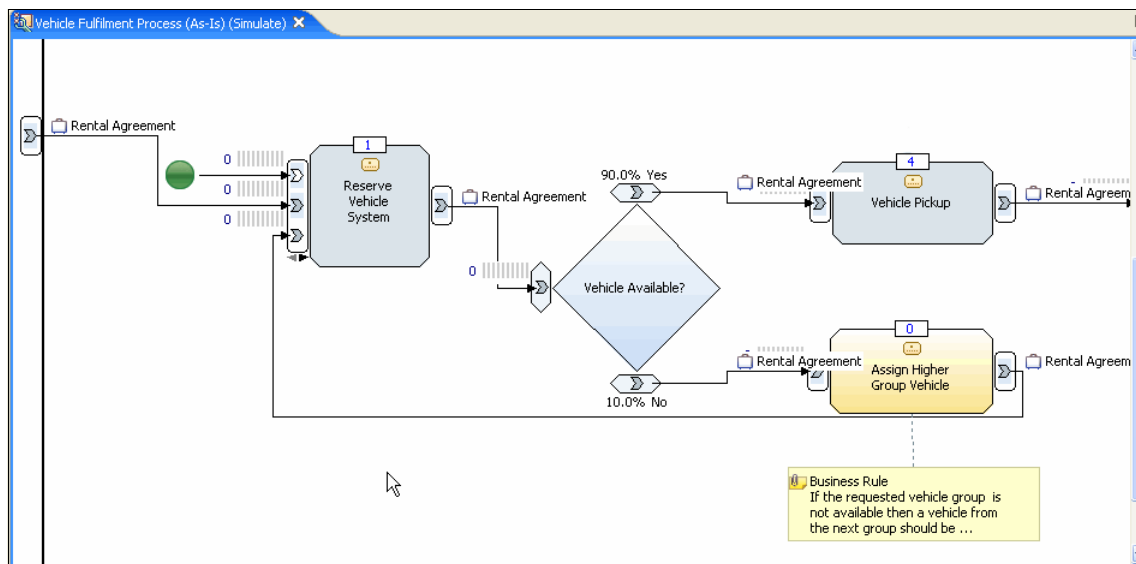


Figure 6-3 Simulation snapshot

6.2.4 Business measures design

Successful Business Process Management (BPM) implementations rely on well-defined key performance indicators (KPIs) defined in a Business Measures Model. A KPI is a performance metric for a specific business activity that has an

associated business goal or threshold. The tracking and analysis of KPIs helps business users gain the insight required for business performance optimization.

WebSphere Business Modeler provides a Business Measures Editor to describe the business performance aspects of a business model as a Business Performance (Observation) Model. This model is used by WebSphere Business Monitor to create custom dashboards for business users to follow the status of the business process, examine and analyze historical information, and even perform trend analysis.

Note: The business measure model is defined by businesspeople and later imported into WebSphere Business Monitor. In most cases, it is independent from the IT integration developers and there is no refinement effort necessary in WebSphere Integration Developer.

The Business Measures Editor provides high-level constructs such as triggers, counters, timers, metrics, and aggregation so that business architects or business process engineers can define the Business Measures independent of the IT integration developers. This high-level construct can be assigned to tasks of the process graph to measure, capture, and calculate values that are of interest for business monitoring.

The Business Measures Editor uses the following high-level constructs.

Metrics

A metric is a measurement of a process or process element that is used to assess business performance with WebSphere Business Monitor. Typically a metric is a variable that is assigned a value based on an incoming event, counter, or timer.

KPI

A key performance indicator (KPI) is associated with a specific process and is generally represented by a numeric value. Typically a KPI is displayed as a gauge in WebSphere Business Monitor. A KPI may have a target and allowable margins, or lower and upper limits, forming a range of performance that the process should achieve. A KPI can be thought of as a metric with limits, and is itself made up of one or more metrics. An example of a simple KPI: Average time for response to a customer inquiry is less than five days.

Triggers

A trigger is a mechanism that detects an occurrence and initiates an action in response. For example, you could set a trigger to update a metric calculation each time a task ends. Triggers can detect such things as the arrival of an input at a specific task, a change in the value of a metric, or a recurring period such as

every three hours. In response, triggers can do things such as start stopwatches, increment counters, and change the values of metrics. Triggers can detect any of the following changes in a process instance:

- ▶ A change of state. Each element, such as a task, has states such as Start, Stop, and Suspend.
- ▶ A change in a metric.
- ▶ A change in a counter.
- ▶ A specific time interval.
- ▶ The arrival of an input (to a task or process).
- ▶ The production of an output (of a task or process).

A trigger can initiate any of the following actions:

- ▶ Evaluation of a metric (the value is recalculated each time the trigger is activated).
- ▶ A change in a stopwatch (stop, start, or reset).
- ▶ A change in a counter (increment, decrement, or set to zero).
- ▶ The production of a situation event.

Counters

Counters are specialized business measures elements that keep track of the number of occurrences of a situation or event.

Stopwatches

If business measures depend on elapsed time or durations, you can define stopwatches in the business measures model to keep track of the timings. You can use a stopwatch to track the elapsed time since a request was sent, for example, or the time since the claim process started. Stopwatches can be used to measure durations within one run of the process, but you can reuse them within a KPI or aggregate metric to determine the average, maximum, minimum, or total duration across multiple runs of the process. Triggers can initiate the following actions on stopwatches.

Start	A stopwatch begins to count the elapsed time duration.
Stop	The stopwatch stops counting and keeps its value. If you start the stopwatch again, it begins counting by adding to the previous stored value.
Reset	Sets the value to zero and stops.

Aggregation

KPIs, metrics, timers, and counters are calculated on a single process run (process instance). KPIs are often defined as averages on multiple process runs. The aggregate function enables calculation of values across multiple runs of the process. The following aggregation functions are supported:

- ▶ Calculate the average
- ▶ Capture the maximum value
- ▶ Capture the minimum value
- ▶ Sum the total of all instances
- ▶ User defined to calculate average values based on other metrics or totals

Situation events

Situation events are generated when a business situation arises that needs immediate management actions such as alarms. A business situation could be an unacceptable customer response time or a declining sales count that has dropped below a predefined threshold. Metrics, KPIs, counters, and triggers can all optionally generate situation events. When the process is running after being deployed, WebSphere Business Monitor will be listening for this situation event. An administrator can specify in WebSphere Business Monitor to display the events in the Alerts view or send a message or e-mail to notify the appropriate person to take action.

You have the option to add a condition. For example, you might not want a situation event to be generated every time the business measure changes, but only when the measure reaches or exceeds a specific value. The situation event condition is evaluated each time the value of the measure changes and you can specify whether multiple events can be generated from the same process instance.

Build a business measure model

Figure 6-4 on page 86 shows how a business measure model is built with the Business Measure Editor on a given process graph. Typically, a trigger starts off or initiates other constructs such as counters, stopwatches, or metrics calculations. The metric or KPI values can be aggregated over all relevant process instances. Finally it is displayed in WebSphere Business Monitor as gauges or other business measures diagrams. In addition you can elect to fire situation events on specific business or process conditions when met. The Action Manager in WebSphere Business Monitor receives these events and can send alarms or initiate actions such as starting a process instance in WebSphere Process Server.

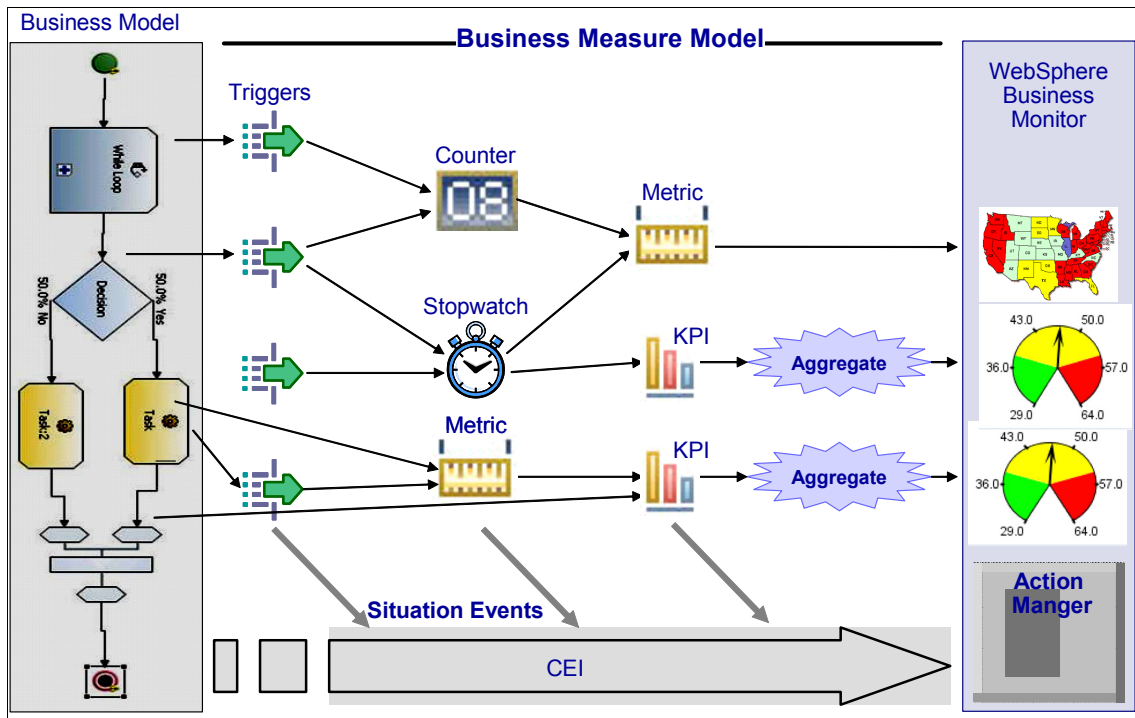


Figure 6-4 How a business measure model is build from high-level constructs

6.2.5 Modeling for execution and monitoring

IBM WebSphere Business Modeler is a process modeling tool for business process management, analysis, detailed reporting, and workflow integration. This section focuses on business modeling elements and their transformation into Web Services Business Process Execution Language (WS-BPEL) constructs that enable integration developers to implement the process for assembly in WebSphere Integration Developer and execution in WebSphere Process Server.

Understanding the different modeling elements and their translation is an important step to ultimately understanding what elements are eligible candidates for modeling a WS-BPEL process. WebSphere Business Modeler provides an extensive variety of modeling elements and artifacts that are not supported in WebSphere Integration Developer. Further, to start modeling with WS-BPEL transformation in mind will limit the necessary rework efforts in WebSphere Integration Developer and keep the business measure model for monitoring in WebSphere Business Monitor in synchronization with the execution model.

The generated WS-BPEL process might not be optimal

WS-BPEL processes, modeled in WebSphere Business Modeler, are designed to preserve the overall control flow, transition logic, and data (process and task level); as a result, the generated WS-BPEL is not intended to be fully optimized and requires refinement changes in WebSphere Integration Developer.

WebSphere Process Server Modeling Mode

WebSphere Process Server Modeling Mode restricts certain elements from being used in the process editor diagram. Only the elements that are relevant to a WS-BPEL process model are included. As shown in Figure 6-5, the following elements are not supported:

- Notification Broadcaster
- Notification Receiver
- Observer
- Time
- Do-While Loop
- For Loop

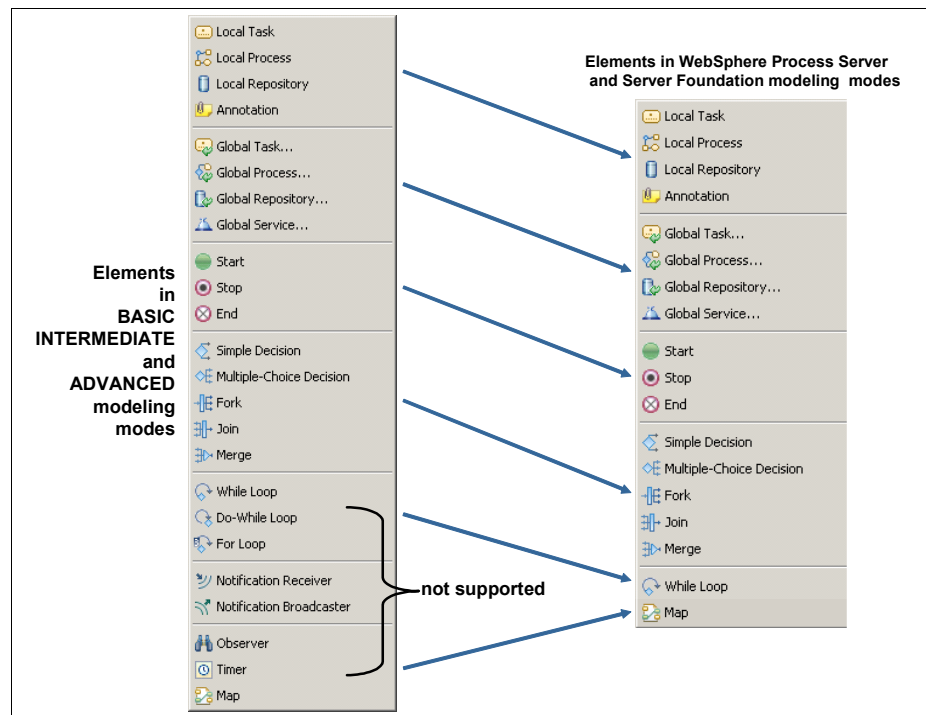


Figure 6-5 Comparing modeling elements in Basic and WebSphere Process Server Mode

Artifacts generated after export

Exporting a project, or specific elements of a particular project or catalog, in WebSphere Process Server (BPEL, WSDL, XSD, SCA) format produces several different artifacts and files. Each of these artifacts contains pertinent data that can be used collectively in a WS-BPEL process to define data inputs and outputs, service invocations using partner links, and control flow. Figure 6-6 shows a high-level target mapping of the elements contained in a process diagram in WebSphere Business Modeler and the artifacts where these elements will be stored.

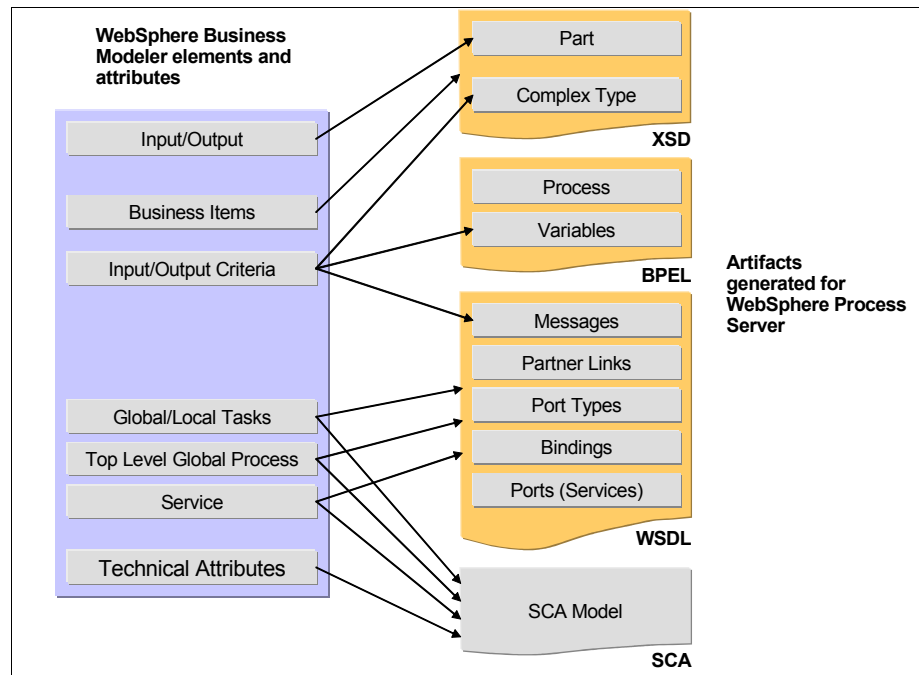


Figure 6-6 Generated artifacts

Modeler elements mapped to WebSphere Process Server

Table 6-1 shows the general mapping of modeler elements into WS-BPEL constructs with WebSphere Process Server mode.

Table 6-1 Mapping of modeler elements

Modeler elements	BPEL, WSDL/XSD constructs	Comments
Top-level global process	WS-BPEL process with a flow structure	
Global process within another process	WS-BPEL invoke and WSDL definitions	
Local subprocess within another process	WS-BPEL invoke activity	
Local task	WS-BPEL invoke activity and WSDL port type	
Global task	WS-BPEL invoke activity and WSDL file with port type	
Service within a process	WS-BPEL invoke activity and WSDL definition with a port type	
Business item	XSD complex type	
Map	Empty Java activity	Java implementation must be created in WebSphere Integration Developer
Connection with an assigned business item	WS-BPEL link and assigning or retrieving from variables	
Decisions	Options set as technical attribute: 1. Links with transition (default) 1. Switch activity	
Expressions	WS-BPEL Java conditions	
Role requirement	WS-BPEL human task	Should be of type Staff or Person
Individual resource requirement	WS-BPEL human task	Should be of type Staff or Person

Modeler elements	BPEL, WSDL/XSD constructs	Comments
While loop	WS-BPEL while with a flow structured activity	A loop condition is mapped to a WS-BPEL Java condition
Local repository	WS-BPEL variable	
Input or Output	WS-BPEL variable and WSDL element (doc/lit wrapped style)	
Input criterion Output criterion	WSDL document/literal wrapped style message	
Join	Empty activity if business items flow true	To capture logic of business items merges

Note: Modeling elements such as Organizations, Roles, Resources, Nodes, and others are not supported or relevant for the transformation to WebSphere Integration Developer.

Process mapping

A WebSphere Business Modeler process maps to a WS-BPEL process and a top-level flow activity. The input and output criteria defined for a WebSphere Business Modeler process represent the entry and exit point of the process in WebSphere Integration Developer. The type of WS-BPEL activities that are generated to represent the input and output criteria depends on the type of operation set for the process in the technical attributes. The export supports two types of WS-BPEL operations:

- ▶ The request/response operation expects a request message and returns a response message back to the caller.
- ▶ The one-way operation expects a request message but either has no response message or it uses a callback-type response message.

The entry point of the process exports to a WS-BPEL receive or pick activity and the exit point exports to a reply for a request/response operation or an invoke activity for one-way with callback operation. This is determined using the following rules:

- ▶ If a WebSphere Business Modeler process has a single input criterion defined, it exports to a single BPEL receive activity.

- ▶ If the process has more than one input criterion, the transformation to WS-BPEL depends on whether a correlation is defined:
 - If a correlation is not defined, a WS-BPEL pick activity is exported. Each input criterion exports to an onMessage activity within the pick.
 - If a correlation is defined, multiple WS-BPEL receive activities are exported along with an empty WS-BPEL correlationSet.

In addition, there is a WSDL partner link and a portType interface generated from the input and output criteria of the process. If a callback response is defined, the export also generates a callback portType.

Local task mapping

A task exports to a WS-BPEL partner link. The export generates a portType interface for each task. Same as the input and output criteria of a process, the technical attributes of the task determine whether the task maps to a one-way operation or a request/response operation. One difference is that the callback response is not supported. The export therefore generates a WS-BPEL invoke activity with a request and response for a request/response operation and an invoke activity with only a request for a one-way operation.

Local process mapping

A local process exports to a WS-BPEL scope activity with a top-level flow activity. The contents of the process export to various WS-BPEL activities that are connected with WS-BPEL links. There is no direct mapping of input or output criteria because a scope activity does not have an interface. The export automatically reconnects the connectors that link activities outside the local process to activities inside the local process through the input criteria.

Connector mapping

A connector that does not have an associated business item exports as a WS-BPEL link. If multiple connectors are exiting from the same output in the same output criteria and connecting to inputs in the same input criteria, the export generates a single link between the same source activity and target activity. A connector that has an associated business item exports as a shared variable between WS-BPEL activities.

Decision mapping

WebSphere Business Modeler uses two types of decisions: exclusive decisions where only one path occurs at one time, and inclusive decisions where several paths are followed in parallel at one time.

- ▶ An exclusive-type decision dependent on its technical attributes set exports as a WS-BPEL switch activity or as WS-BPEL links with TransitionCondition.
 - If the decision exports as a switch activity, each output branch exports as a case in the switch. A switch activity explicitly represents a decision node in WS-BPEL.
 - If the decision exports as WS-BPEL links (default), formal expression conditions defined in WebSphere Business Modeler are translated to TransitionConditions.
- ▶ An inclusive-type decision only exports as a WS-BPEL link with TransitionConditions.

While loop mapping

A while loop exports as a WS-BPEL while activity with a top-level flow activity. The content of the while loop exports as various WS-BPEL activities that are connected using WS-BPEL links. Because the while activity has no interface, the export automatically reconnects the connectors that link activities outside the loop to activities inside the loop. The loop condition of the while loop, if defined, exports to Java code in the WS-BPEL while loop condition.

Local repository mapping

A local repository exports as a WS-BPEL variable. The export achieves the copying data to and from the local repository that occurs in WebSphere Business Modeler by defining the same WS-BPEL variable for the request or response parameters of invoke, receive, and reply activities. In this way, the data held by the repository is implicitly retrieved or stored.

Human task generation

The export generates two types of human tasks: inline task, or as a human task component for Human Task Manager (HTM), which is an SCA component that has the implementation type of human task.

- ▶ To export as an inline task, in WebSphere Business Modeler the task must have a role requirement or an individual resource requirement. The requirement must use the predefined Person or Staff resource templates or a definition based on one of those templates. For a role requirement of a task, the type must be specified as the predefined Person or Staff, or a resource definition based on the Person or Staff template.

- To export as a human task component, the implementation type must be set to human task in the technical attributes.

SCA component generation

The export generates an SCA component for each process, global task, local task, and service element that is not generated as an inline task. The generated component contains component references to create the component wiring information displayed in the Assembly Editor in WebSphere Integration Developer.

The technical attributes define a set of component properties including the implementation type of the component. The implementation type can be process, human task, Java, state machine, or business rule group. If the technical attributes specify an implementation type, the export generates the default implementation based on the implementation type. The default implementation provides a starting point to implement the component. For a WebSphere Business Modeler process, the implementation type is always *process*.

In addition to generating components, the export also generates an SCA export for the global elements (process, global task, and service) because these elements are visible and may be used by other processes. This action ensures that the SCA components representing the global elements are visible between modules and that they can access each other.

WSDL portType generation

The export generates a portType as the interface of a WebSphere Business Modeler process, global task, local task, or service. Their input criteria exports to operations within the portType. For a process that allows a callback-type response for an output criterion, the export generates a callback operation for each of its callback output criteria and places them in the callback portType. The input and output criteria also define the signature (that is, the WSDL input and output attributes) of the operation. The input criteria map to an input message and the output criteria map to an output message.

The exported portType conforms with the document/literal wrapped/JAX-RPC wrapper style format. This means that the WSDL message contains a single part that refers to a wrapper XSD element. The wrapper XSD element has a complex type that contains a sequence of elements. Each of these elements then represents an input or output parameter to the operation.

Important: Names assigned to input and output elements in WebSphere Business Modeler can contain white space. When viewing the corresponding WSDL message part name in the WebSphere Integration Developer, all white space is removed. Further, the word “Part” is appended to the end of the input/output element name.

Technical attributes

WebSphere Business Modeler provides Technical Attributes views to specify technical implementation details, which are used to implement a process, global task, or service in WebSphere Process Server. Solution architects or IT specialists can specify the details to reduce the refinement effort in WebSphere Integration Developer. The technical attributes enable you to specify the values used in the export in place of the generated default values.

Figure 6-7 shows the technical attributes available on process level. You can specify different target namespaces, process names, and whether the process should be a persistent long-running flow or runs as a subprocess (child). On the interface level you define WSDL interface information, such as the namespace and the portType name. The Request/Response window enables you to define the type of operation, such as request response or one-way.

Car Rental Fulfilment (To-Be)

Specification

- General
- Interface
- Request
- Response
- Implementation

BPEL Attributes

Define BPEL process information below

Target namespace

Process display name

Car Rental Fulfilment

Process name

CarRentalFulfilment

☒ Process is long-running

☐ Process runs as child process

Figure 6-7 General technical attributes on process level

On task level, the technical attributes let you select the component you want to use in WebSphere Integration Developer to implement the task. The implementation type can be process, human task, Java, state machine, or business rule group. The export generates the SCA component wiring based on the specified implementation type.

6.2.6 Importing and reusing existing Web services definitions

The use of existing assets, such as Web services, presents two options during business modeling:

- ▶ Top-down approach

Business modeling is typically a top-down approach. Business people model the process flow optimized for their business needs and goals. Thereafter the IT architects and specialists take over the business flow and refine it toward

building an SOA composite application. Reuse of services and components mainly takes place in the integration layer.

► Bottom-up approach

Reuse existing service implementations for business-level modeling.

Figure 6-8 shows these two approaches.

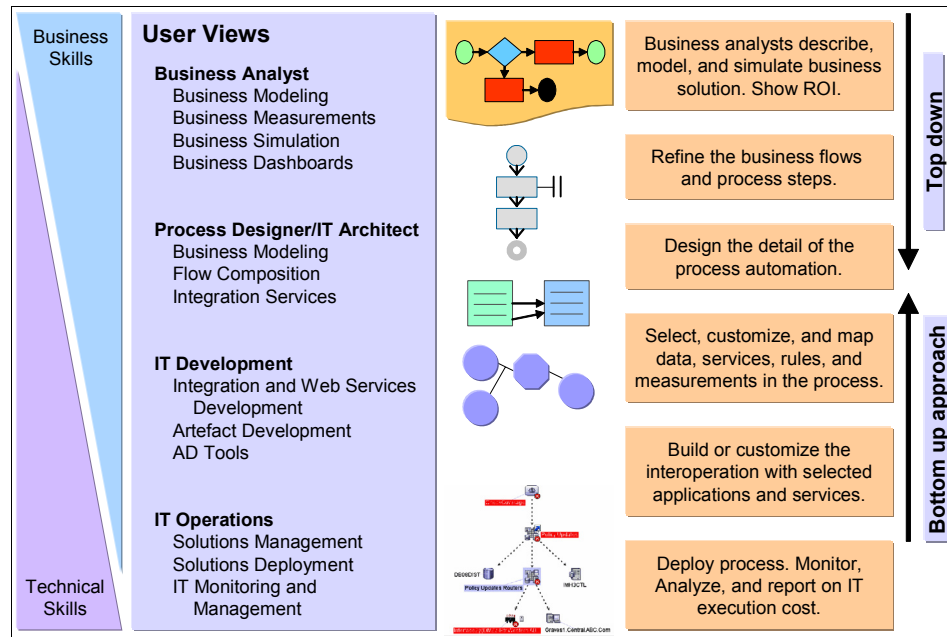


Figure 6-8 Top-down and bottom-up approaches

Web Service Interface import technology preview

WebSphere Business Modeler V6 includes as a technology preview a Web Service Interface import feature, which is accessible from the Import wizard. As a technology preview, it is a not-supported feature in this release. The intention of this technology preview is to enable you to use existing Web Services interface definitions for business modeling with WebSphere Business Modeler, and therefore use the bottom-up approach.

The following transformations take place in WSDL import:

- Each Web Services portType creates a Global Services element in WebSphere Business Modeler. You can model the business process using services that represent the existing Web Services portTypes (Figure 6-9 on page 96).

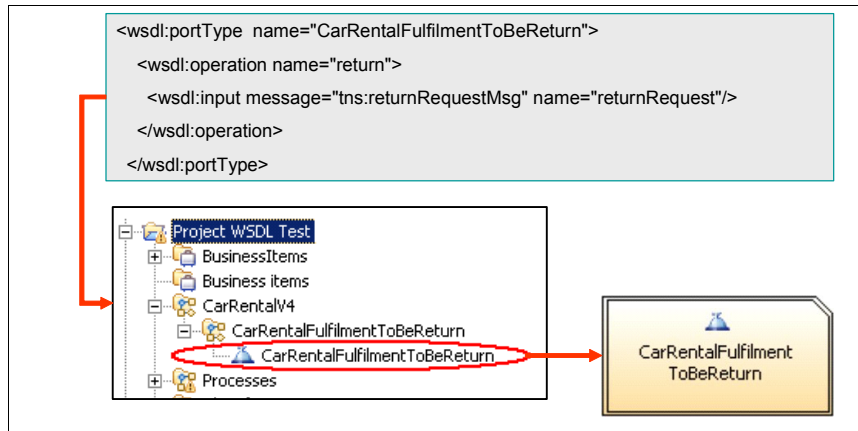


Figure 6-9 WSDL portType conversion on import

- WSDL types definitions are translated to Business Items (Figure 6-10).

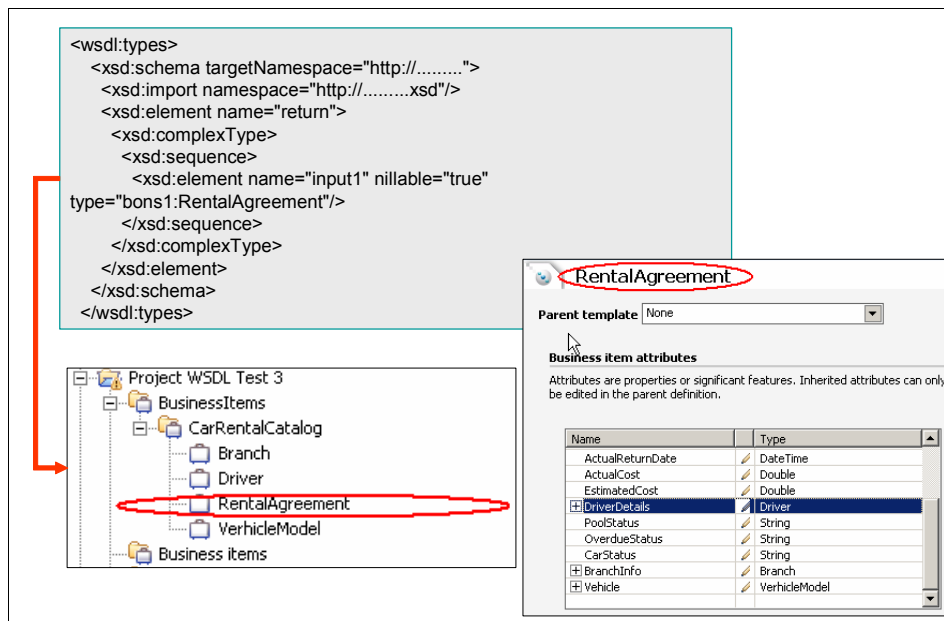


Figure 6-10 WSDL type conversion on import

The WSDL import supports the following Web services styles:

- RPC
- Document literal

The following elements in a WSDL file are not supported:

- ▶ Document literal wrapped
- ▶ Services, Ports, and Bindings
- ▶ Operation faults
- ▶ Solicit-Response and Notification type operations

If you export a service created by the Web Service Interface import to WebSphere Process Server, an import with a Web service type binding will be created instead of a component.

Importing WSDL interfaces

To import definitions from WSDL files, complete the following steps:

1. In the project tree, right-click and select **Import**. The Import wizard appears.
2. Select **Web Service Interface (.wsdl)** and click **Next**.
3. Click **OK** on Web Services Import is a technology preview.
4. Click **Browse** to select the source directory that contains the WSDL files you want to import.
5. In the Files list, select the file or files. You can select more than one file by holding down the Ctrl key.
6. In the Target project field, select an existing project from the pull-down list or click **New** to create a new project.
7. When you have finished specifying your import options, click **Finish**. A window opens when the import process is complete.
8. If there were any errors or warnings during the import process, click **Details** to read them. Otherwise, click **OK**.



Assembly capabilities of WebSphere Integration Developer

In this chapter, we explore the capabilities of WebSphere Integration Developer by answering a number of key questions:

- ▶ What is the business value of using WebSphere Integration Developer to build business process management applications?
- ▶ What do I need to know about the IBM SOA programming model?
- ▶ What is the role of the integration developer and why is it crucial to the success of my service-oriented architecture efforts?
- ▶ What are the key component types I can build with WebSphere Integration Developer?
- ▶ What are the important integration services I can build with WebSphere Integration Developer?
- ▶ What are the essential terms I need to know to start working with WebSphere Integration Developer?

7.1 Business value of using WebSphere Integration Developer

It has been estimated that up to 70% of IT spending is for system and application integration. Integration consists of connecting existing systems to facilitate business processes that span these systems. The systems to be integrated can be anything from mainframe systems, client/server applications, databases, messaging systems, ERP and CRM systems, application servers, and beyond.

Technology that has been brought into an organization over time to solve some business problem, as well as technologies that have been brought in as a result of mergers and acquisitions, eventually must be integrated as part of common business processes. As time goes on and the number of systems to integrate increases, the time and expense involved in integrating these systems can become prohibitively high.

The advent of message-oriented middleware was the first step in the solution to this problem, in that it provided a way for applications to communicate using messages with assured delivery. The next step was enterprise application integration (or EAI), with the typical example being the hub-and-spoke architecture. With EAI, the routing and transformation of messages could be handled by a common messaging hub. The next quantum leap to solve the integration problem is service-oriented architecture, or SOA.

In addition to integrating a company's IT systems, SOA is a way to integrate the business itself. It does this by organizing and integrating the business around reusable services and processes. SOA provides a real answer to IT integration because it is defined on common industry standards that represent these services and processes. The emergence of standards-based services and processes makes it possible to integrate disparate systems and technologies at a much lower cost and complexity than previously possible.

How does this relate to WebSphere Integration Developer? WebSphere Integration Developer is based on the IBM SOA programming model, and as such it provides several key business benefits that will help your company attain the benefits of SOA:

- ▶ Enables specialization of development roles into business process analysts, service developers, and integration developers, which maximizes development effectiveness and provides a bridge between line-of-business and IT.
- ▶ Empowers an IT organization to create reusable services that are built to integrate with other systems rather than to stand alone, which maximizes return on investment for development resources.

- ▶ Simplifies the integration programming model and enables IT to quickly respond to changing requirements from line-of-business, which enables a business to be more agile in the marketplace and more adaptable to change.
- ▶ Provides the capability for IT organizations to integrate disparate heterogeneous systems and components (such as mainframe transaction systems, ERP systems, Java application servers, .NET servers, and MQ-enabled systems) and enables them to achieve the benefits of SOA while achieving maximum use of existing IT systems.

7.2 The IBM SOA programming model

Integration projects have long been too difficult because there are myriad different but similar technologies. There are many ways to represent or interact with data and there are many ways to describe or invoke existing application logic. Figure 7-1 shows a few relevant technologies in this area.

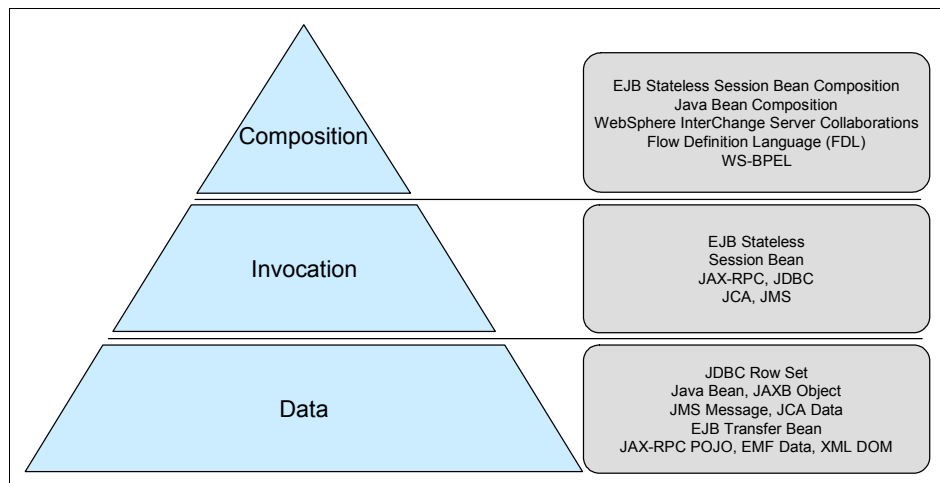


Figure 7-1 Traditional programming model

Too often up to now, most of the time and resources in a development effort have been spent picking and choosing among various implementation technologies to use and then integrating them, instead of working at the level of solving a business problem with smart development tools.

The IBM SOA programming model offers a new level of abstraction at each of these levels so that the task of integrating applications and services becomes much easier (see Figure 7-2 on page 102).

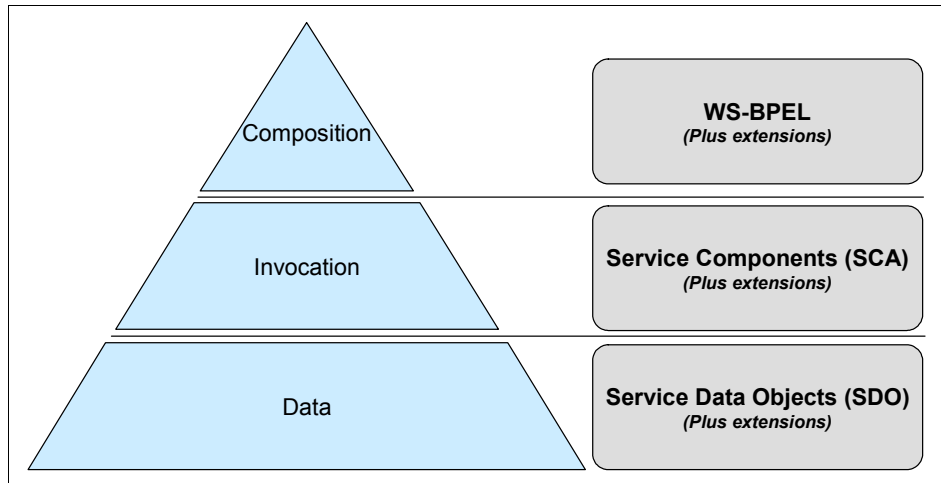


Figure 7-2 IBM SOA programming model

Each of these emerging standards has been supported and is being adopted by major industry leaders in addition to IBM. At the data level, the SOA programming model offers Service Data Objects (SDOs) as an abstraction. SDO is used to implement common business objects and provides a universal means of describing and accessing data. Extensions to SDO are implemented in WebSphere Process Server and include support for metadata, context information, and change history. These extensions are being added into the next version of the emerging SDO standard.

At the invocation level, the SOA programming model offers Service Component Architecture (SCA). SCA describes all integration artifacts as service components with well-defined interfaces. This abstraction enables you to define business components abstractly, separate from their technical implementation. SCA also introduces the concept of a module, which groups service components and provides further specification and encapsulation of services.

WebSphere Integration Developer provides an *assembly editor* where the developer can group service components into *modules* and specify which service *interfaces* are exposed by the module to outside consumers. It also defines common *libraries* to store artifacts that will be shared across modules, such as interfaces or data maps. Modules expose their functionality by way of *exports*, and use the functionality of other modules using *imports*. Modules can be connected using *wires* to form complete integration solutions or composite applications.

Tip: We provide definitions for each of these highlighted terms at the end of this chapter to reference as you work with WebSphere Integration Developer.

The concepts of SCA enable a developer to encapsulate integration logic within modules. This means that a change to service components in a module does not affect any of the other modules in the overall solution as long as the interface of the changed module stays the same. This concept has been applied throughout WebSphere Process Server. All integration artifacts in WebSphere Process Server (processes, business rules, human tasks, and so on) are represented as SCA service components. This creates an environment with great flexibility. It is possible, for example, to replace a human task for an approval with a business rule for automatic approval simply by replacing the service components in the assembly diagram without changing either a business process or the caller of the business process.

Additionally, with SCA you can invoke service components using both synchronous and asynchronous programming styles, whereas in the past each would have its own separate programming model. For example, you can assemble modules into composite applications where asynchronous channels between service components and modules can increase the overall throughput and flexibility of the system.

At the composition level, the SOA programming model offers Web Services Business Process Execution Language (WS-BPEL). WS-BPEL defines a model and grammar for describing the behavior of a business process based on interactions within the process and its interactions with external partners. It can be used to specify both the public interfaces for the partners and the description of the executable process. A partner can be any entity that provides a service, consumes a service, or both.

WS-BPEL provides the means to specify business processes that consist of service invocations and the ability to expose the actual business process as Web services. The interaction with each partner occurs through Web service interfaces, and partner links encapsulate the structure of the relationship at the interface level. The WS-BPEL process defines the sequencing of service interactions with partners. The process also provides the coordination to achieve a business goal and the state management and logic necessary for the coordination.

For a more extensive treatment of the SOA programming model, refer to the series of articles and podcasts on IBM developerWorks® at:

- ▶ Introduction to the IBM SOA Programming Model
<http://www.ibm.com/developerworks/webservices/library/ws-soa-progmodel/>
- ▶ WebSphere Technical Podcast series
<http://www.ibm.com/developerworks/podcast/websphere>

For more about SCA/SDO and WS-BPEL emerging standards, refer to:

- ▶ Specifications: Service Component Architecture and Service Data Objects
<http://www.ibm.com/developerworks/webservices/library/specification/ws-scasdosumm>
- ▶ Business Process Execution Language for Web Services version 1.1
<http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel>

7.3 Role of the integration developer

To understand the value of WebSphere Integration Developer, you must understand the role of the integration developer in the SOA lifecycle. Consider the SOA lifecycle with its main phases of Model, Assemble, Deploy, and Manage. In the Model phase, the business process analyst is responsible for modeling, analyzing, and optimizing the business process at an abstract level, then exporting the process model to the integration developer for the assembly phase. Next is the Assemble phase (the primary focus of this chapter). The integration developer is primarily responsible for this phase of the lifecycle.

The integration developer's responsibilities include taking the set of abstract artifacts (data schemas, process definitions, and service types) created by the business process analyst and providing the concrete implementation details to create a fully functional composite application. Some examples of these implementation tasks are:

- ▶ Identifying and integrating new business logic services, including services developed by J2EE developers using tools such as Rational Application Developer or Rational Software Architect, or services developed by .NET or PHP developers
- ▶ Integrating and repurposing existing assets as services, for example by connecting to enterprise information systems using WebSphere Adapters
- ▶ Assigning specific enterprise services (such as Web services, business rules, human tasks, Java components, and adapter services) to process tasks within the WS-BPEL process diagram
- ▶ Configuring transactional behavior and compensation into the business process to provide appropriate transaction semantics as well as handling for exception conditions
- ▶ Creating maps to transform business objects and interfaces from one format to another

- ▶ Configuring a directory and other settings for the human task service so that the process knows how to find appropriate users and roles to work on human workflow tasks
- ▶ Creating business rule templates that enable business analysts to manage business rules independently and to change business rules dynamically according to the needs of the business

As you can see, the integration developer role is an important one. This role does not require knowledge of Java, although it can be helpful. Instead, the role requires knowledge of how to integrate services and systems into the SOA programming model supported by WebSphere Integration Developer. The integration developer works with both the business process analyst and the service developer to implement composite applications. An effective integration developer is key to creating successful composite applications that can be managed and maintained over the life cycle of your SOA.

7.4 Getting to know WebSphere Integration Developer

This section introduces the main features provided by WebSphere Integration Developer for use by the integration developer.

7.4.1 Key component types

The following components are key aspects of WebSphere Integration Developer to an integration developer.

Business processes: the basis of composite applications

A business process is the set of business-related activities, rules, and conditions that are invoked in a defined sequence to achieve a business goal (for example, an order-to-fulfillment process for a manufacturer). As such, there is nothing inherently IT-related about business processes in themselves. However, with WebSphere Integration Developer your processes form the very basis of the applications you develop. This is accomplished by the creation of WS-BPEL business processes.

WS-BPEL defines a model and grammar for describing the behavior of a business process based on interactions between the process and its interactions with external partners. It can be used to specify both the public interfaces for the partners and the description of the executable process itself. A partner can be any entity that provides a service, consumes a service, or both.

- ▶ A process model that describes the logical order in which the different activities of the process are being executed
- ▶ The individual services or components that implement the specific activities

WS-BPEL provides systematic mechanisms for dealing with business exceptions and processing faults. WS-BPEL implements a framework for defining how individual or composite activities in a process are to be compensated in cases where exceptions occur or a partner requests the reversal of a scope of work.

The diagram illustrates the relationships between components in a service-based architecture. It features three main components: an **Export** component (green arrow), a **Service Component** (blue box), and an **Import** component (purple arrow). The **Export** component has an **I** (Interface) role. The **Service Component** has an **I** (Interface) role and two **R** (Reference) roles. The **Import** component has an **I** (Interface) role. The **Service Component** is associated with an **Implementation** of **Business Process** (yellow box). The **Import** component is associated with an **Implementation** of **Java** (yellow box). The **Export** component is associated with an **Implementation** of **Standalone Reference** (green box). The **Service Component** is associated with an **Implementation** of **Service Component** (blue box).

A business process in turn invokes services as SCA components. WS-BPEL partner links in the process are resolved to SCA components or external services using the assembly editor in WebSphere Integration Developer. A WS-BPEL

process uses an invoke activity to invoke other services. That activity specifies the partner and the operation that is to be called on behalf of the process, the data that is used as the input message to that operation, and, in the case of a request-response operation, where the result data is to be stored. Figure 7-4 shows an example of WS-BPEL process for order fulfillment in the Business Process Editor.

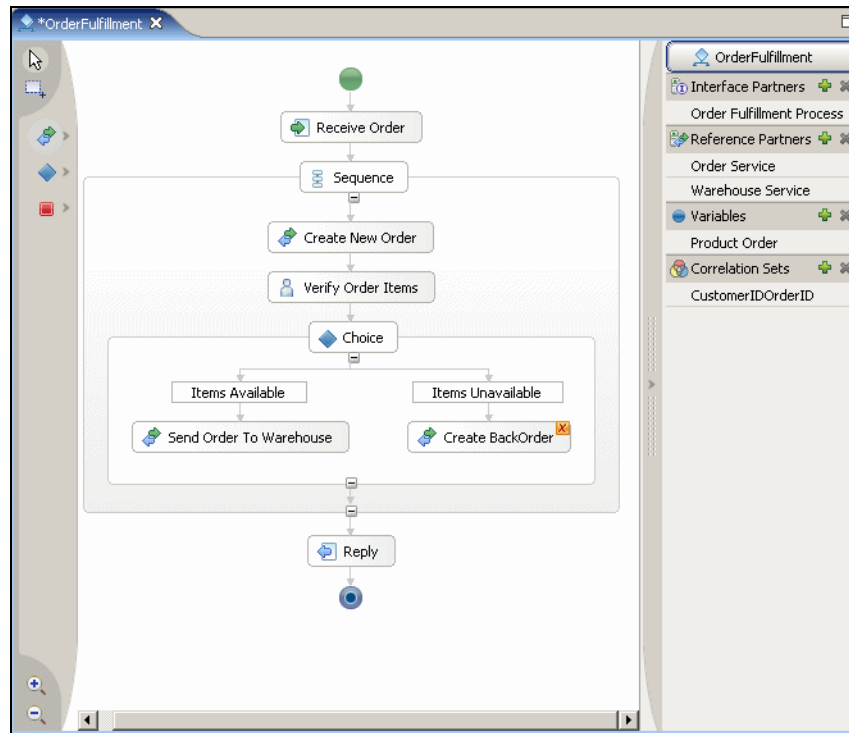


Figure 7-4 Example order fulfillment business process in the Business Process Editor

WS-BPEL processes can transparently handle one-way, request-response operations, and synchronous and asynchronous interaction modes, using the same syntax for all cases. The business flow manager (inside WebSphere Process Server at runtime) dynamically detects the type of service implemented by the corresponding partner and uses the appropriate invocation. This enables the integration developer to create processes that are independent from the physical realization of the involved services.

Human tasks: connecting people to processes

Human workflow is a key complement to process automation in creating composite applications. Human interaction steps are often required for input and initiation of an automated process as well as for manual activities in that process.

Human interaction is often required to review and correct an error or exception situation in a process.

Many challenges arise when you add human interaction to a process integration solution, such as:

- ▶ How to make sure that the correct people are involved
- ▶ How to add capabilities that model real-life human task management, where tasks are claimed, transferred, and completed or are overdue and need escalation
- ▶ How to make sure the human interaction provides a suitable interface without disrupting the flexibility to change the business process

The Human Task Manager provides different types of human tasks to solve different integration problems.

Participating task	A service creates a work item for human interaction (for example, a service implemented in WS-BPEL).
Originating task	Human interaction invokes a service (for example, a business state machine).
Human task	Human interaction invokes a service that creates a work item for another human to complete.

WebSphere Integration Developer can help the integration developer configure all of these tasks, including such runtime details as their user registry mapping information. At runtime, the Human Task Manager in WebSphere Process Server dynamically binds the human task to the appropriate staff that need to perform the tasks, as shown in Figure 7-5 on page 109. The human tasks are presented to the users through customized Web user interfaces based on Java Server Pages or portlets. These tasks can then be claimed, transferred, worked, completed, escalated, and so forth by the people performing the tasks.

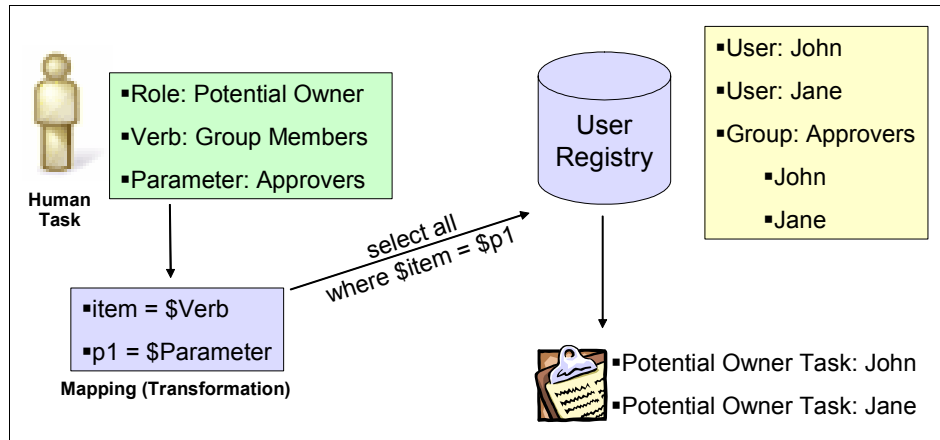


Figure 7-5 Assigning tasks to users at runtime

Human tasks themselves can be invoked as SCA components; therefore integration with other SCA components, such as WS-BPEL business processes, becomes very easy. As with any SCA component, SDO provides the standard data format for passing data to and receiving data from human tasks.

Business state machines: designing processes using states

A state machine model describes the dynamic behavior of an application or business process by focusing on the events that cause a transition from one state to another. A business state machine is an implementation of a business model that executes (that is, moves from state to state) based on real-time events. With a business state machine, you can model event-based processes easily and efficiently. States and state transitions form the basis of the process. Business logic is embedded in the transitions between two states. This means that WebSphere Process Server contains two types of business processes: WS-BPEL and state machines.

You should use a WS-BPEL process when:

- ▶ Steps in a process tend to happen in sequence.
- ▶ Some event handling and looping is present.

You should use a business state machine when:

- ▶ The business process is heavily event driven.
- ▶ The reaction to these events is dependent on the process state.
- ▶ The process might revert to prior states.
- ▶ Some sequential steps are involved.

Note: At runtime, a process modeled as a state machine is still described as a WS-BPEL process. The process is deployed to the same business process engine that is used for ordinary WS-BPEL processes. The main difference between a business state machine and a WS-BPEL process is noticeable at development time. One process style is better suited for a certain class of business processes than another, but at runtime the difference is minimal.

When discussing state machines, a number of terms are often used and they have specific meaning in the context of state machines:

State	One of several discrete individual stages that are organized in sequence to create a business transaction.
Composite states	An aggregate of one or more states that can be used to decompose a complex state machine diagram into an easy-to-comprehend hierarchy.
Operations/events	An external prompt that attempts to trigger a movement from one state to another.
Transitions	The movement that occurs through the recognition of an appropriate triggering operation, the evaluation of the conditions necessary for execution to flow through it, and the determination of what actions can occur if execution is allowed.
Conditions	Guards the transition and allows the transition to the next state only if the incoming operation evaluates to <i>true</i> . Otherwise, the current state is maintained.
Actions	An operation that is executed at one of three distinct locations in the business state machine: during a transition, when a state is entered, or when a state is exited.
Correlation	The record that is used to keep track of multiple participants in the same business transaction. The state machine uses correlations to distinguish the parties in their initial interactions so that they can recognize each other in the future.

Events map to operations that are available in the SCA component of the state machine. Business objects, primitives, or both can be used as input/output parameters. Part of the input parameter is used to correlate to a state machine instance.

A timeout can cause a transition. For example, a state transition can occur when a certain duration is exceeded (one day, for example) or when the state becomes expired (for example, December 31, 2006).

If a transition has no event (operation) or timer, it is considered to be an automatic transition. Automatic transitions enable sequential processing. Automatic transitions in conjunction with conditions can provide default behavior.

A business state machine is an SCA component. The interfaces of such an SCA component are used to define the events of the state machine. In other words, SCA operations are events in a state machine. The actions in a state machine can refer to other SCA components. Business objects can be defined and assigned to instance variables within the state machine.

Business rules: externalizing dynamic business policies

Business rule groups are a means of implementing and enforcing business policy through the externalization of business rules. Externalization enables the business rules to be managed independent of other aspects of an application. This independence provides dynamic updating capabilities to the business rules, which can result in a more agile business.

Often business analysts focus on business policy to describe guidelines that drive a business and can range from marketing heuristics, accounting principles, government regulations, manufacturing standards, or supply management quality of service levels, to any other aspect of the business that relies on specific conventions. Business rules represent a common way to implement and enforce these business policies.

There are two styles of business rules:

- ▶ If-then rule set
- ▶ Decision table

Business rules are invoked through a business rule group component, which provides the interface for the business rule. Each operation defines a business need and not a specific rule implementation.

Business rules, by their nature, change over time to support, for example, changing business policy or changing government regulations. The ability to schedule rules to be in effect during specific time periods provides the flexibility for an organization to respond to changes. With the integrated effective dates, the operation can be implemented with more than one business rule with each implementation, which is effective for a specified period of time. Effective dates enable a client application to invoke the business rule group component as though it were a time in the past, present, or future.

Both WebSphere Integration Developer (for the developer) and the Web-based Business Rule Manager application for the business analyst support authoring of business rules. The business rule group component in WebSphere Integration Developer includes support for deploying, installing, dynamically authoring, and

executing the business rules. Updated business rules can in turn be exported from the Business Rule Manager and re-imported into WebSphere Integration Developer.

Business rules are invoked through a business rule group component, never directly. The business rule group component can be invoked as an SCA service. As with any SCA service, the client does not need to be concerned with the type of implementation that is used to process the request. The client simply interacts with the SCA component as a black box to meet a desired business need. Figure 7-6 shows an example business rule group containing two business rules.

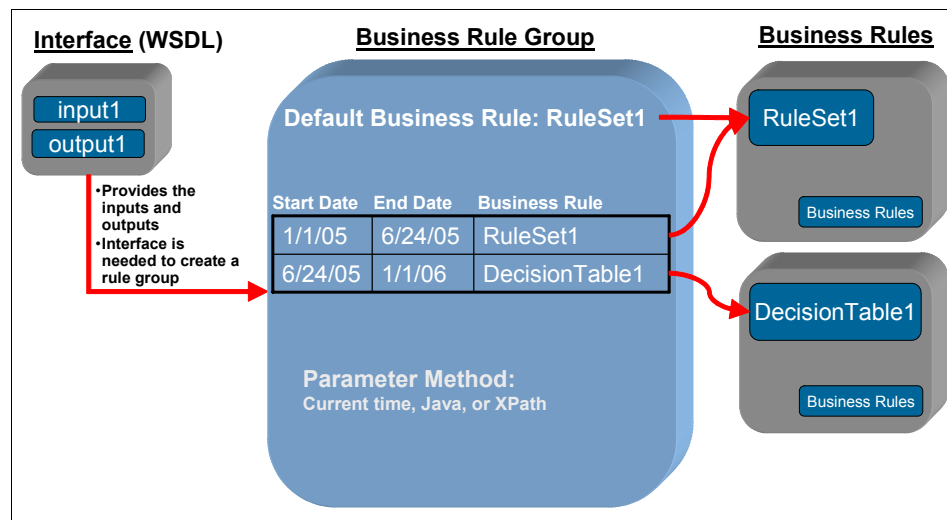


Figure 7-6 Business rule group as an SCA component

A *rule set* is a set of one or more if-then condition/action statements that are evaluated sequentially. Therefore, the first business rule listed will be the first one evaluated when called on at runtime. The second is evaluated second, the third is third, and so forth until the last one is evaluated.

Rule sets hold any number of action/if-then conditions. The if-then condition contains the decision logic, and the action condition performs an action based on the decision. A rule set can evaluate multiple conditions and can fire multiple rules. This might be necessary when a rule is found true and executed, but more processing must be done to execute any other business rules in that rule set.

A *decision table* represents a multi-dimensional nested if-then structure. Think of it as a rule set that can handle more complex decisions than a simple if-then. The decision table operates in a tree decision structure, a set of “if” conditions with “then” actions that are defined at the intersection points of the table. Conditions

are evaluated in a nested order (tree view). The decision table evaluates one or more conditions, but executes only one rule.

Use decision tables when:

- ▶ You have rules with multiple clauses/variables in the conditional statements.
- ▶ You want to execute only one rule.

Use rule sets when:

- ▶ You have rules with a few clauses/variables in the conditional statements.
- ▶ You want to execute multiple rules.

Templates provide the mechanism for supporting Web-based rule authoring at runtime. The templates provide a constrained method for a business analyst to author rules in real time. Templates are applicable to both (if-then) rule sets and decision tables.

Templates enable application developers to:

- ▶ Control which aspects (conditions/actions) of a business rule can be modified by the business analyst
- ▶ Define constraints (that is, valid discount range, customer status, gold/silver/bronze, and so on)
- ▶ Define the structure of an if-then rule, then create instances of this template
- ▶ Define the structure of a condition case, action, or both for a decision table
- ▶ Define a natural language string that is used in the Web-based tool for presenting the rule to the business analyst

Business rules are also important for supporting business process management because they help businesses of any size proactively change processes to respond to competitive threats, new opportunities, or changing customer demands. Large enterprises with many processes tend to move slowly when responding to customer demand changes, but can no longer afford to move slowly due to ever-present competition from nimbler, smaller competitors. With business rules, processes can be changed in minutes so that businesses of all sizes can respond and change much more quickly.

Business rules are a boon to increased IT flexibility because developers are freed from tasks that a more experienced business person can (and should) do. In other words, a developer does not have to change an application every time a simple rule has to be changed. A business analyst who is more in tune to business needs can make that change swiftly. Another attribute of business rules that improves IT efficiency is the ability to have multiple processes call a business rule at the same time. This multi-threading enables multiple processes to run at the same time and share a common repository of business rules.

EIS imports and exports: leveraging key IT systems

The ability to connect to and service-enable their Enterprise Information Systems (EIS) is crucial to most organizations that adopt SOA. WebSphere Integration Developer can use WebSphere Adapters to connect to these EIS and transform their functions into services that in turn can be made part of process-based composite applications. It does this by the use of EIS imports and exports.

Java: leveraging key investments in Java and J2EE

A POJO (plain old Java object) can also act as the implementation of a service. Such a service has an interface that is nothing more than a Java interface declaration. For a POJO, you have the choice between a Java type of invocation or a Web Services Description Language (WSDL) type of invocation. Example 7-1 shows a simple Java interface called HelloWorld.

Example 7-1 Java interface definition

```
public interface HelloWorld {  
    public String sendMessage(String message);  
}
```

A service is described in a service definition file that contains information about the services that are called, if any, and the exposed interfaces. Example 7-2 shows a simple service information file. The interfaces section describes the exposed interfaces, and the implementation section lists the Java class that implements this service.

Example 7-2 Service information file

```
<?xml version="1.0" encoding="UTF-8"?>  
<scdl:component xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:java="http://www.ibm.com/xmlns/prod/websphere/scdl/java/6.0.0"  
  xmlns:ns1="http://HelloWorld/HelloWorldInterface"  
  xmlns:scdl="http://www.ibm.com/xmlns/prod/websphere/scdl/6.0.0"  
  xmlns:wSDL="http://www.ibm.com/xmlns/prod/websphere/scdl/wSDL/6.0.0"  
  displayName="HelloWorld" name="HelloWorld">  
  <interfaces>  
    <interface xsi:type="wSDL:WSDLPortType"  
      portType="ns1:HelloWorldInterface">  
      <method name="sendMessage"/>  
    </interface>  
  </interfaces>  
  <implementation xsi:type="java:JavaImplementation"  
    class="com.ibm.itso.sca.sample.HelloWorldComponent"/>  
</scdl:component>
```

In addition to POJOs, WebSphere Integration Developer can also import session Enterprise JavaBeans™ and transform them into callable SCA components.

Web services: integrating with standards-based services

A Web service can be any service, of whatever programming language or technology implementation, that is accessible using a SOAP interface. It is simple to import standard Web services into WebSphere Integration Developer and make use of these services in your composite applications. All that is required is to locate the service interface file, which is composed in WSDL, and import it into the WebSphere Integration Developer environment.

7.4.2 Integration services

This section introduces several WebSphere Integration Developer features for integrating interfaces, message content, relationships, and so forth.

Interface maps: adapting incompatible interfaces

Flexible and scalable business integration and SOA-based solutions often require support for different types of transformations in order to accommodate and adapt to the heterogeneity of IT systems and services. WebSphere Integration Developer delivers flexible and powerful transformation capabilities through interface maps.

The interface map component provides resolution and reconciliation of differences among interfaces found between SCA components. A map can be created that understands one interface (`cancelOrder`) and invokes another interface (`updateOrderStatus`) as shown in Figure 7-7. In addition, interfaces can map the data on these interface calls through a data map.

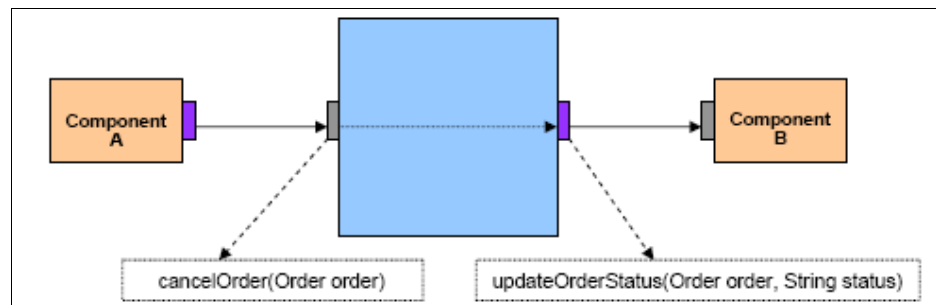


Figure 7-7 Interface mediation component

An interface map is an SCA component that is logically separate from the source and target components. Figure 7-8 on page 116 shows the high-level view of the

relationship between maps, services (EIS imports/exports), and the business process.

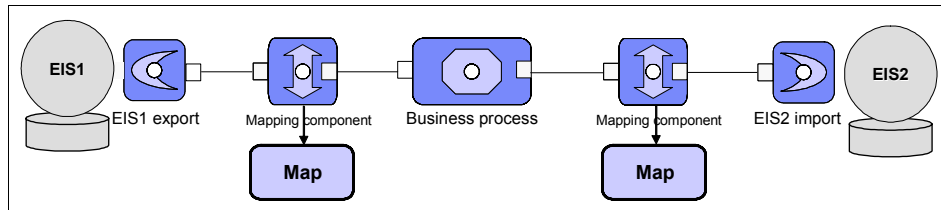


Figure 7-8 Role of mapping components in a WebSphere Process Server flow

Data maps: translating business objects between systems

WebSphere Integration Developer contains a rich graphical mapping tool to create data maps, which provides the structural and semantic transformation of business objects. The mapping tool offers move, join, extract, and assign functionality for data fields as well as custom mapping (through custom activities or Java code). It also can create submaps for complex business object structures and relationship mapping.

In Figure 7-9, an adapter interacts with a back-end enterprise information system (for example, SAP) to create, read, update, or delete events of application-specific business objects (for example, SAPCustomer and SAPOrder). The adapter publishes these events to a set of business processes that are modeled using generic business objects (for example, Customer and Order) for further processing. In this case, the data map, which is called from within the interface map, allows the request to be transformed into a canonical, or generic, format (generic business object) from an application-specific format (application-specific business object). This built-in functionality results in the transformation of the operation name (interface map) and the content and structure of the parameters that are being passed as part of the operation (data map).

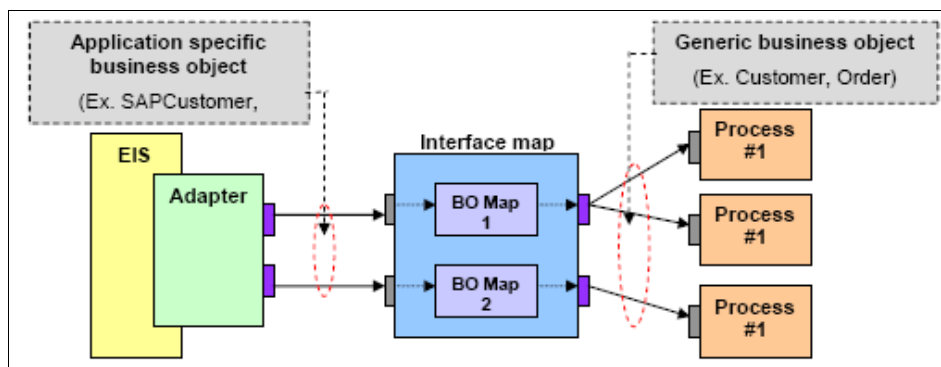


Figure 7-9 Data maps used by an interface map

Relationships: maintaining a single view of related items

In business integration scenarios, accessing the same data (for example, customer records) in various back-end systems is often a requirement. A common problem for keeping data in sync is that different back-end systems use different keys to represent the same data. The relationship service in WebSphere Process Server is used to relate these disparate data sources. As a business object is converted from one application-specific representation into another, WebSphere Process Server can dynamically maintain a database of keys, which enables the mapping of one data record into another between disparate back-end systems.

Figure 7-10 shows the concept of a dynamic identity relationship with a customer identifier maintained in four applications. These four systems use different identifiers and store customer information in a different format or object model. A typical integration requirement is to propagate new or changed records in all four systems. In conventional middleware solutions, this solution is extremely difficult to develop and, more important, to manage continuously. WebSphere Process Server automates this activity through a set of capabilities that provide an integrated framework for managing dynamic relationships.

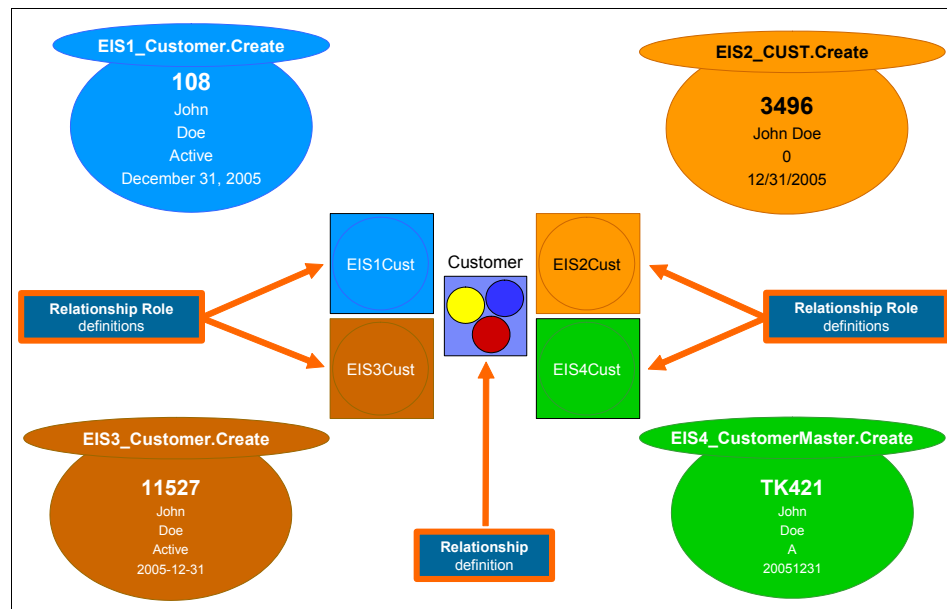


Figure 7-10 Multiple sources of customer record information

Figure 7-11 on page 118 shows how WebSphere Process Server maintains information about each application. Basically, a new identifier is created for each customer, and that identifier refers to the specific identifiers in each application. Thus, when information is received from the application that is carrying identifier

108, then that identifier is replaced with the generic identifier 42 during processing in WebSphere Process Server. When the business object is sent to one or more target applications, the generic identifier 42 is replaced with the appropriate identifier for that system (for example, 3496 in the second application).

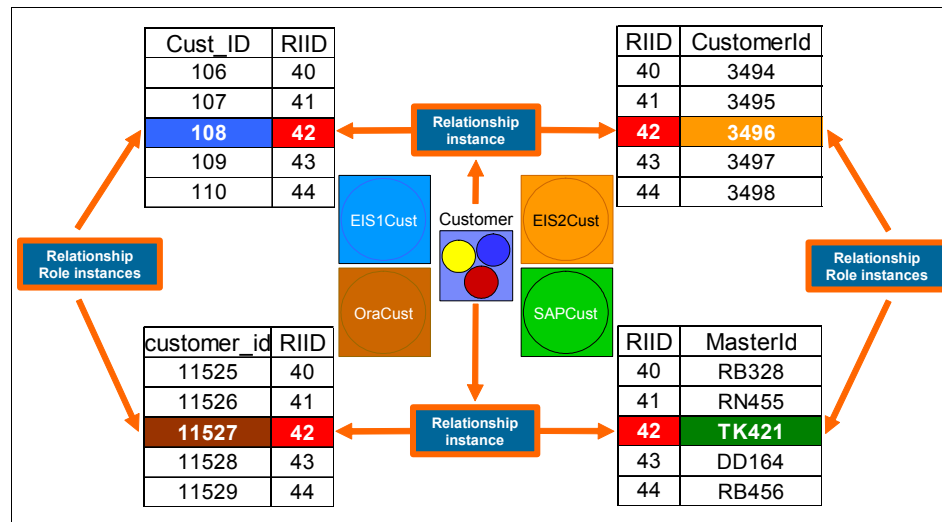


Figure 7-11 Relationship instances linking multiple customer identifiers

Creating and maintaining these links between identifiers is completely automatic in WebSphere Process Server. Relationships are available directly from business object maps and simplify the management of disparate back-end data representations.

Selectors: providing dynamic service binding at runtime

The selector component provides a dynamic selection mechanism for invoking service components at runtime. The objective of the selector component is to:

- ▶ Determine dynamically which implementation of a target destination to invoke based on some defined set of criteria, data, or logic. (Currently the selector component only supports selection based on date and time.)
- ▶ Decouple the client application from a specific target destination implementation, allowing for dynamic selection and invocation of a target destination.
- ▶ Allow new implementations of target destinations to be added to the system without requiring changes to the client application.
- ▶ Allow new SCA implementations of a target destination to be added to the selector component dynamically through a Web interface without requiring a restart of the application or server.

An SCA component itself, a selector has an interface (Figure 7-12). Note that the interface of the selector is the same as all interfaces of the target components.

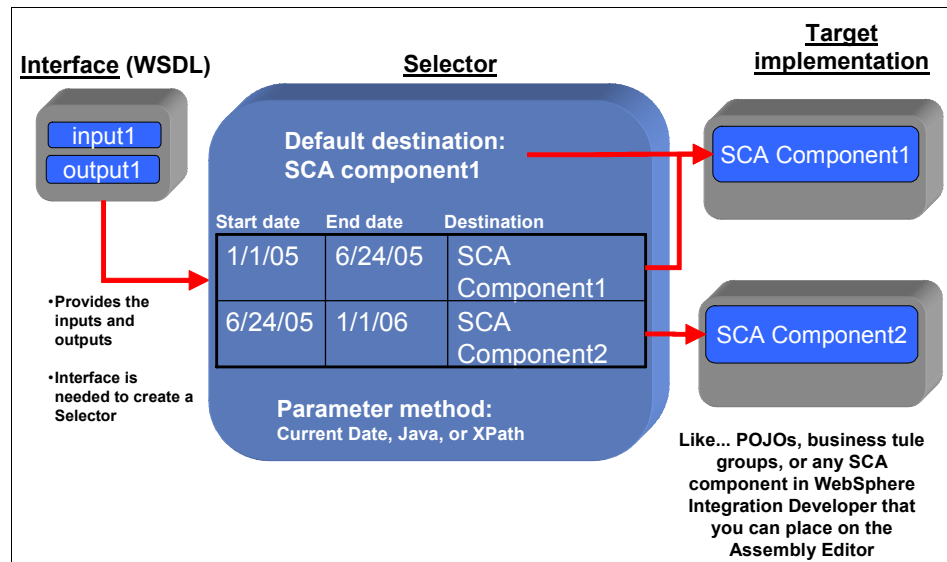


Figure 7-12 How selectors work

Visual snippets: creating business logic without coding

Visual snippets are small functions that provide pieces of business logic. These can be created by WebSphere Integration Developer using its visual snippet editor, as shown in Figure 7-13. They can then be used inside many components of a composite application, including business processes, business rules, Java components, and maps. This capability gives the integration developer the power to create pieces of business logic without Java coding. The visual snippet editor can use the full power of the Java language in addition to its visual capabilities, so snippets can also be created by developers with Java proficiency, and can then be shared and reused among integration projects.

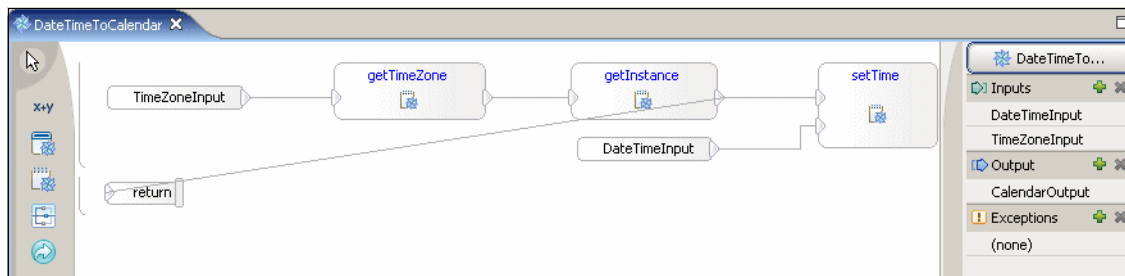


Figure 7-13 Visual snippet editor provides visual creation of business logic

7.4.3 Essential terms and concepts

Finally, to complete this chapter about capabilities of WebSphere Integration Developer, we provide concise definitions of its most important terms and concepts. With an understanding of these, you will be well prepared to start developing solutions with this revolutionary integration toolset.

Module	A group of related components that are managed together in WebSphere Integration Developer, are organized under a single assembly diagram, and are deployed as a single unit to WebSphere Process Server
Library	A group of shared artifacts (such as interfaces, business objects, and data maps) that are shared across different modules
Assembly diagram	A visual representation that wires together components into a module
Interface	A set of exposed operations that define a service component, which the implementation of the component must adhere to
Reference	An external interface that a component depends on, typically provided by another component
Import	A reference to the interface of an external service, often used to provide a linkage between components in different modules or to invoke external services such as Web services or EIS services
Export	A reference created by a component within a module that exposes its interface outside the module so it can be used by other (SCA) modules or by non-SCA clients
Wire	A visual representation of linkages between references and interfaces, which provides the essential means to connect components together in an assembly diagram



Deployment capabilities of WebSphere Process Server

This chapter describes the business value and capabilities of WebSphere Process Server as an SOA-based Business Process Management (BPM) runtime environment. You can deploy, run, and administrate BPM applications in the runtime environment.

In this chapter, the following topics are discussed:

- ▶ Business value
- ▶ Capabilities
 - Capabilities as a Service Component Architecture (SCA) and WS-BPEL runtime environment
 - Capabilities to work with external resources
 - Installation capabilities
 - Capabilities to run BPM applications
 - Administration capabilities
 - Capabilities to configure security to use with human tasks
 - Version control capabilities

8.1 Business value

In this section, we describe the following important business values provided by WebSphere Process Server:

- ▶ Reliable and simple operation environment
- ▶ Protection of investment
- ▶ Agility built into operations

8.1.1 Reliable and simple operation environment

WebSphere Process Server provides an environment with the same level of reliability as the market-proven WebSphere Application Server, on which it is built. WebSphere Process Server has the following characteristics:

- ▶ Performance

WebSphere Process Server offers the market-proven performance of WebSphere Application Server.

All WebSphere Process Server components run as standard Java classes, Enterprise JavaBeans (EJB), and Message Driven Beans (MDB). WebSphere Process Server performance heavily depends on its base J2EE runtime, WebSphere Application Server.

- ▶ Security

The base J2EE runtime, WebSphere Application Server, addresses all of the necessary security requirements such as authentication, encryption, authorization, and non-repudiation.

All WebSphere Process Server components run as standard J2EE resources. WebSphere Process Server can be configured to authenticate users, secure communications, authorize access to users, and secure the J2EE applications.

- ▶ Scalability

The base J2EE runtime, WebSphere Application Server, addresses load balancing and high-availability requirements.

The WebSphere Application Server workload management module distributes incoming work requests to application servers according to the capacities of the different machines. It also provides failover when servers are not available.

The WebSphere Application Server high-availability manager makes the components of WebSphere Process Server, such as the messaging engine, highly available.

Because all of the BPM applications, integration modules, and services are running on single runtime, the operation environment is very simple, which can save on system management costs.

8.1.2 Protection of investment

WebSphere Process Server runtime is based on open standards. Conformance to open standards can protect a company's IT investment by providing a continuous road map for IT systems.

WebSphere Process Server uses the following open standards:

- ▶ Java 2 Enterprise Edition (J2EE)
J2EE standards includes enterprise application packaging, Enterprise JavaBeans (EJB), Java Servlet, Java Server Page (JSP™), Java Connector Architecture, and Web services support.
- ▶ WS-BPEL
WebSphere Process Server fully supports Web Services Business Process Execution Language (WS-BPEL) standards as its primary business process language.
- ▶ Common Base Event (CBE)
WebSphere Process Server formats all event objects by using the standardized event format CBE.
- ▶ Web services standards
WebSphere Process Server supports Web Services standards that are not included in J2EE standards, such as WS-Security and WS-AtomicTransaction.

8.1.3 Agility built into operations

WebSphere Process Server provides the Business Rules Manager tool. By using this tool, business users and process administrators can change the business rules dynamically without coding. As a result, the company can adapt to changes in the business environment quickly.

Using the selector component, the BPM applications can change the service used for a given task dynamically. Administrators can replace the services by changing the configuration without coding.

8.2 Capabilities

In this section, we discuss the following important capabilities that WebSphere Process Server provides:

- ▶ Capabilities as a SCA and WS-BPEL runtime environment
- ▶ Capabilities to work with external resources
- ▶ Installation capabilities
- ▶ Capabilities to run BPM applications
- ▶ Administration capabilities
- ▶ Capabilities to configure security to use with human tasks
- ▶ Version control capabilities

8.2.1 Capabilities as an SCA and WS-BPEL runtime environment

WebSphere Process Server is an integration engine that natively supports SCA and WS-BPEL.

SCA is a next-generation implementation method for SOA applications where all applications are reusable, interoperable, replaceable, and scalable. Pre-built components that are based on SCA are important to speed up the development cycle and adapt to change quickly.

The pre-built SCA components include: Interface Map, Business Object Map, Relationship, Selector, Business Rule, Business State Machine, Human Task, and WS-BPEL.

WS-BPEL has the important role of making a business process visible, manageable, and changeable in a standard way.

SCA runtime

Set Quality of Service (QoS) to specify how SCA components interact with each other at runtime. WebSphere Process Server determines how client components interact with target components using service qualifiers.

Service qualifiers can specify the following characteristics:

- ▶ Transaction support

Set each service component to run in global transaction or in its own local transaction. Using this service qualifier, you can group several service components into one global transaction or separate them into several transactions.

- **Reliability**

If service components interact with other service components asynchronously, set the quality of asynchronous message delivery. Using this qualifier, set the message delivery to be assured without loss or set the message delivery to be possibly lost with performance increase.

- **Security**

If security permissions should be checked against service components, you can set a security qualifier. The security qualifier specifies which role the calling client should have and which role the service component should have. This role is then implemented using WebSphere Application Server security functionality.

- **Session management**

An activity session is one of the methods to group service components into a unit of work. Any number of service components can be grouped into one activity session or separated into several activity sessions.

WS-BPEL runtime

WebSphere Process Server supports WS-BPEL as the primary implementation language for business process.

WebSphere Process Server supports transactions of business process by relying on the base J2EE engine. It provides capabilities for grouping multiple activities into one transaction. You can set transaction boundaries in a business process.

Because there are resources that do not support the two-phase commit protocol, WebSphere Process Server supports compensation transactions for these resources. Undo operations can be set for each activity in a short-running process. The undo activities are automatically performed in reverse order when failure occurs. You can also set compensation activities as a process flow in a long-running process. If failure occurs in a specific scope of process, we specify which compensation flow should be run to compensate that scope.

More information

- More information about SCA can be found at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.ovw.doc/doc/covw_sca.html

- More information about pre-built SCA components can be found at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.ovw.doc/doc/covw_sca_servicecomp.html

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.ovw.doc/doc/covw_support.html

- More information about WS-BPEL can be found at:

<http://www.ibm.com/developerworks/library/specification/ws-bpel/>

8.2.2 Capabilities to work with external resources

To make BPM applications interact with the outside world and to make the integration points changeable, reusable, and manageable, the integration engine should have various, consistent ways to access outside resources.

WebSphere Process Server uses J2EE standards to provide access points to resources such as data sources, resource adapters for various applications and technologies, and Java Messaging Service (JMS) resources.

Data sources

For service components or services to connect to relational databases, you can configure JDBC providers and data sources. A Java Database Connectivity (JDBC) provider is an object that provides a connection to a physical database using a vendor-specific JDBC driver. A data source is an object to connect to a specific database instance. The connection per data source is managed in a connection pool.

Resource adapters

WebSphere Process Server supports J2EE Connector Architecture (JCA) V1.5.

Enterprise Information Systems (EIS) or adapter vendors can develop resource adapters that are based on JCA and make them compatible with WebSphere Process Server.

Besides the vendor-provided adapters, WebSphere Process Server provides several pre-built adapters such as the WebSphere Adapter for CICS, IMS, Siebel, JDBC, and SAP as a WebSphere Process Server package.

If the SCA applications are developed to use resource adapters using EIS exports and EIS imports in WebSphere Integration Developer, the resource adapters are installed automatically when the SCA applications are installed on the WebSphere Process Server.

JMS resources

WebSphere Process Server supports JMS V1.1 and V1.0.2.

EIS vendors or messaging vendors can develop messaging engines or messaging applications based on JMS and make them compatible with WebSphere Process Server.

Besides the vendor-provided messaging infrastructure, WebSphere Process Server provides a pre-built messaging infrastructure called the default messaging engine.

If the SCA applications are developed to use JMS resources using JMS exports and JMS imports in WebSphere Integration Developer, the JMS resources should be configured in the WebSphere Process Server using the administrative console or wsadmin scripts.

8.2.3 Installation capabilities

WebSphere Process Server provides a J2EE standard way to install BPM applications so that all the applications are installed and managed the same way.

You can install BPM applications by using one of the following methods:

- ▶ Administrative console installation wizard

The easiest way to install BPM applications is using the browser-based administrative console guided by step-by-step instructions as shown in Figure 8-1 on page 128.

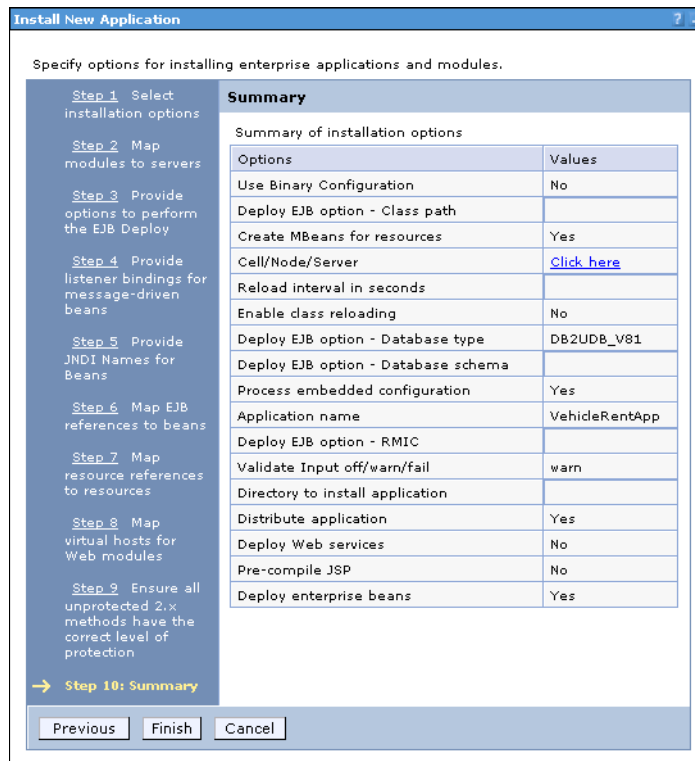


Figure 8-1 Installation using the administrative console

► wsadmin scripts

You can install BPM applications by using wsadmin scripts from the command prompt. Create scripts to install BPM applications from a command prompt without interacting with a browser-based administrative console.

► Java application programming interface

You can develop an installer program by using the MBean API. Using this programmatic approach, you can automate installation tasks. Detailed information and a sample application can be found at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tjmx_install_app.html

► WebSphere rapid deployment

This is a useful installation method for users who do not know the J2EE deployment method. Also, this method provides a way to dynamically apply

the changed J2EE artifacts without using other deployment tools. Detailed information can be found at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.etools.wrd.freeform.doc/topics/cwrdrapid.html>

- ▶ Java programs that use J2EE DeploymentManager (JSR-88) methods
You can develop an installer program using the J2EE DeploymentManager (JSR-88) API. This is another programmatic approach to install J2EE applications. Detailed information and sample application can be found at:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/trun_app_jsr88.html

8.2.4 Capabilities to run BPM applications

After the BPM application is deployed into the runtime server, it must be tested so that testers, developers, and administrators can quickly and easily find out whether the BPM applications are working as intended.

The Business Process Choreographer (BPC) Explorer is a pre-built browser-based client where users can work on tasks and process administrators can administer process templates, process instances, task templates, and task instances.

The BPC Explorer can be used to test human tasks quickly without actually building client applications. After testing is complete with the default BPC Explorer, build a custom BPM Web client using Java Server Faces (JSF) where pre-built client components can be used, or use another client programming technology.

BPC Explorer as end user tool

A user can perform the following tasks by using BPC Explorer:

- ▶ Start a new process instance if authorized.
- ▶ Display tasks assigned to a user.
- ▶ Claim tasks to start working on a task.
- ▶ Save tasks in progress to continue to work on later.
- ▶ Complete tasks.

BPC Explorer as an administrator tool

An administrator can perform the following tasks using BPC Explorer:

- ▶ Start a new process instance.
- ▶ Suspend and resume process instances.
- ▶ Terminate process instances abnormally.

- ▶ Delete process instances from the system.
- ▶ Administer failed compensations to resolve compensation failure:
 - Display the list of failed compensations.
 - Skip the current compensation action.
 - Retry the current compensation action.
 - Stop the failed compensation.
- ▶ Start a new task instance to assign work items to users.
- ▶ Transfer a work item to another person.
- ▶ Create a work item to assign it another potential owner.
- ▶ Delete a work item that is no longer needed.

Figure 8-2 shows an example of the administration BPC Explorer Web page.

Process Instance

Use this page to view information about a process instance and, optionally, to work on the process instance. [?](#)

Available Actions

Compensate Terminate Delete Suspend Resume Restart Activities ... Tasks ...

Process Description

Process Instance Name Vehicle Fulfilment Process ToBe
Description This process is started after the car rental agreement was filed by an external system. The process handles car handover and car return to the rental agency.
State Running

Details Process Input Message **Activities** Events Tasks Custom Properties

Activity Name	State	Activity Kind	Owner	Activated
Vehicle Collection	Running	Invoke		3/9/06 9:54:51 PM
Assign Higher Group Vehicle	Skipped	Invoke		3/9/06 9:54:51 PM
Reserve Vehicle In System	Finished	Invoke		3/9/06 9:54:49 PM
Vehicle Fulfilment Process ToBe Receive	Finished	Receive		3/9/06 9:54:29 PM

Items found: 4 << Page 1 of 1 >>

Figure 8-2 Process instance administration using BPC Explorer

8.2.5 Administration capabilities

Managing the running BPM applications is critical job for system administrators and process administrators. Using various management tools, the running BPM applications should be visible and manageable in a responsive way.

WebSphere Process Server provides the following administrator tools:

- ▶ Administrative console

The administrative console can configure resources, deploy applications, manage applications, and troubleshoot problems. Detailed common tasks that are associated with the administrative console are found at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tcon_console.html

- ▶ Business rules manager

Business rules manager gives great flexibility to dynamically change business rules that control process flow.

- ▶ Failed event manager

Failed event manager, part of the administrative console, is used to search, inspect, resubmit, and delete failed process instances.

- ▶ Common base event browser

Common base event is a standard way to format event data. The common base event browser, part of the administrative console, is used to search, view, and track common base events that are sent from business processes and SCA components.

Business rules manager

If business rules are introduced during SOA application development, administrators can change these rules without restarting business processes. WebSphere Process Server provides a browser-based business rules manager tool for adding new rule sets and applying them immediately or based on a date/time criteria.

For example, an administrator can change a daily vehicle rental charge rate or the discount rate for a specific royalty type, as shown in Figure 8-3 on page 132.

Rules

Close Template List

Name	Rule	Action
Init	Init	
CalculateRentalDuration	Calculate rental hours and days	
DailyChargeForClassA	If vehicle type is Class <input type="text" value="A"/> , then daily charge is <input type="text" value="50"/> USD	
DailyChargeForClassB	If vehicle type is Class <input type="text" value="B"/> , then daily charge is <input type="text" value="70"/> USD	
DailyChargeForClassC	If vehicle type is Class <input type="text" value="C"/> , then daily charge is <input type="text" value="90"/> USD	
AssignDiscountRateForTypeACustomerRule	If driver's royalty type is <input type="text" value="A"/> , then apply <input type="text" value="40"/> % discount	
AssignDiscountRateForTypeBCustomerRule	If driver's royalty type is <input type="text" value="B"/> , then apply <input type="text" value="20"/> % discount	
CalculateFinalPrice	Calculate final price	

Templates

To create a new rule, fill in data in a template and click "Add" button.

Template Name	Name	Rule	Action
Assign daily charge per vehicle type	<input type="text"/>	If vehicle type is Class <input type="text"/> , then daily charge is <input type="text"/> USD	
Assign discount rate per royalty type	<input type="text"/>	If driver's royalty type is <input type="text"/> , then apply <input type="text"/> % discount	

Figure 8-3 Editing a business rule in the Business Rules Manager

Failed event manager

If failure occurs when running integration applications, the administrator can manage the failure through the failed event manager tool.

In the failed event manager, the administrator can inspect the event message that caused the failure situation and determine whether it can be safely ignored or if this failure situation should be fixed and the event message should be resubmitted. Depending on the decision, the administrator can discard the event or resubmit it after the situation is resolved.

Selector component administration

If a selector component is introduced when developing SOA applications, administrators can change the target service component which the business processes call without restarting the business processes.

Administrators can change the target component based on date/time criteria as shown in Figure 8-4 on page 133. For example, if a new service module which uses a newly introduced system needs to be called from next month, the administrator can set the selector component in advance to call the new module from the next month.

An administrator can also change the target component and make it the current default destination without setting date/time criteria in order to apply the change immediately.

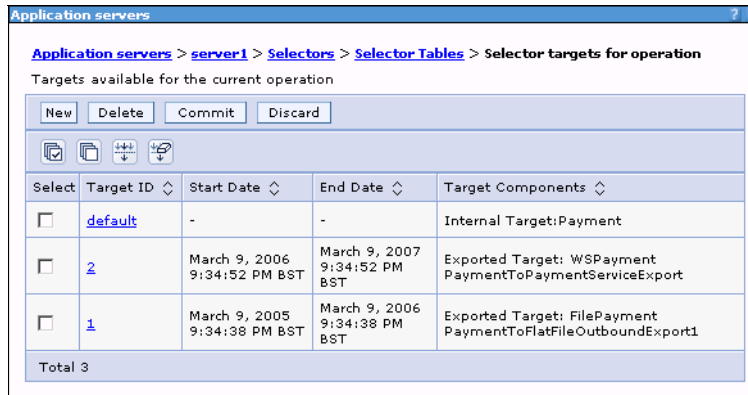


Figure 8-4 Selector component administration

8.2.6 Capabilities to configure security to use with human tasks

To work with human tasks, usually security should be configured to authenticate and give authorization to appropriate users. WebSphere Process Server provides the ability to configure a user registry.

The following user registries are available:

- ▶ Local operating system (OS) user registry

The WebSphere Process Server security default is the Local OS user registry, which tests human tasks with authentication and authorization against the user registry of the operating system where WebSphere Process Server is running. Details on how to configure this can be found at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tsec_localos.html

- ▶ Lightweight Directory Access Protocol (LDAP) user registry

The advantage of using LDAP user registry is that it is a standard protocol, so an existing LDAP registry can be used. Details on how to configure security with the LDAP user registry can be found at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tsec_ldap.html

- ▶ Custom registry

WebSphere Process Server provides a way to create a custom user registry to work with any kind of user registry such as a file, database, or other custom user repository.

- File registry

Our scenario uses a sample file registry to show how to implement a custom user registry. Detailed information can be found at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tsec_tbucs.html

- Database registry

You can configure a custom user registry to use a database such as DB2 Universal Database as the user repository.

8.2.7 Version control capabilities

Version control is important for changing processes smoothly as the business environment and technical environment change. There are two targets of version control: version control of processes and version control of services with which processes are interacting.

Version control of business process

SCA offers great flexibility when changing a process model. Because each component can exist independent of other components, if something in a business process changes, the integration module can be reassembled quickly using the existing components.

- ▶ If only process flow is changed and there is no interface change, use the existing service components to assemble a new version of process module.
- ▶ If a new activity is added into the new version of the process, simply create a new service component that performs a service for the newly added activity and connect them.
- ▶ If the interface is changed in the new version of the process, you must change the interface for the service component that is using the interface. If using interface maps between a business process and service components, you have to change only the interface map component, even if the process interface is changed.

Version control of services

The selector component provides a useful way of implementing version control capabilities in WebSphere Process Server.

The selector component provides the capability for process administrators to change target services dynamically as illustrated in Figure 8-5.

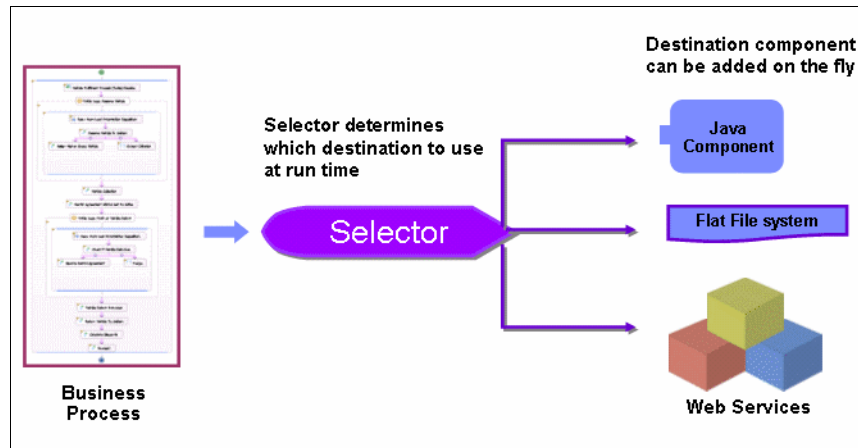


Figure 8-5 Version control of services using selector



Management capabilities of WebSphere Business Monitor

This chapter introduces WebSphere Business Monitor as a key product for performing the Manage phase of the business process lifecycle. WebSphere Business Monitor provides process monitoring capabilities to monitor process execution, analyze the process performance, create reports, and provide real-time feedback for continuous process improvement.

This chapter includes the following sections:

- ▶ Business value of using a process monitoring tool
- ▶ WebSphere Business Monitor capabilities:
 - Real-time business measure extraction
 - Business situation handling
 - Performance analysis and reporting
 - Using historical data for continuous process improvement

9.1 Business value

In the ever-changing business world, comprehensive analysis and process modeling is only a start for overall process management. An optimized business process might seem to be 100% perfect at modeling time, but organizations must to effectively monitor their business processes during execution to determine whether the real running processes are in fact meeting their business goals. Business process monitoring can help businesses examine their running operational processes, react to business events, and keep track of key performance indicators. This can help the business respond quickly to competitive threats, changes in the business environment, or new market opportunities. Process monitoring can thereby find areas of improvement in key business processes and enable an enterprise to increase and maintain its competitive advantage.

Using process monitoring helps define how the business *measures* the operational progress toward achieving goals, *controls* operations by adding the relevant communication and action, *reports* the performance status of key processes to key decision makers, and uses operational analysis feedback to *improve* processes.

9.1.1 Basic activities of process monitoring

Process monitoring involves several distinct activities (summarized in Figure 9-1 on page 139). As a business process runs, the following activities should be performed:

- ▶ Process execution details logged, documented, and profiled.
- ▶ Business metrics measured by analyzing the process execution data.
- ▶ Use analyze of business metrics to detect business situations that require management attention.
- ▶ Determine appropriate control actions to take in response to business situations, whether in mitigating a risk or taking an opportunity. Depending on the nature of the situation, appropriate handling could vary from invoking a defined process to taking an ad hoc action.
- ▶ Analyze process performance data and business measures to identify process improvements.
- ▶ Collect and warehouse business measures for use in historical analysis.

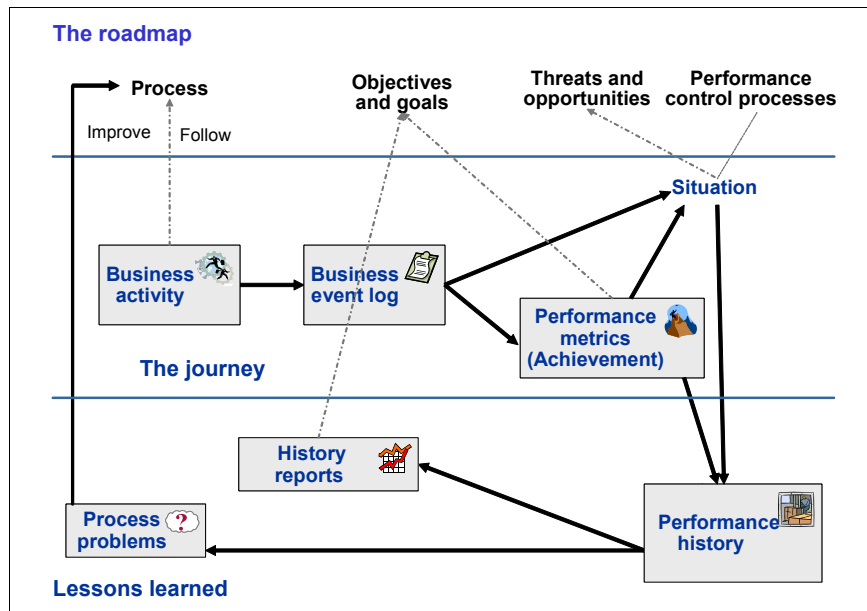


Figure 9-1 Basic activities of process monitoring

9.1.2 Process monitoring using BPM-enabling solutions

For greatest effectiveness, business analysts should give careful thought to Business Process Management (BPM) and monitoring early in the lifecycle. During the process modeling phase, they need to answer the following questions:

- ▶ What strategic goals should be achieved? How will process performance be measured? What are the performance metrics and the key performance indicators?
- ▶ What problems and opportunities occur during business operations? How will they detect business anomalies and business situations?
- ▶ What aggregation dimensions will they need to build their historical analysis?

A business modeling tool should enable a line of business to create these process performance criteria, metrics, indicators, and situations. WebSphere Business Modeler fits these criteria for a business modeling tool.

A process execution environment should be able to:

- ▶ Instantly collect process execution logs and extract specified metrics and performance indicators.
- ▶ Detect the modeled business situations and handle them by appropriate responses.

- Provide business intelligence needed to store, mine, and analyze business measures.
- Present analysis output into customizable reports and charts.
- Use collected data to analyze performance for continuous process improvement.

WebSphere Process Server fits these criteria for a process execution environment.

The role of the business modeling tool and process execution environment are mapped to the basic activities of process monitoring in Figure 9-2.

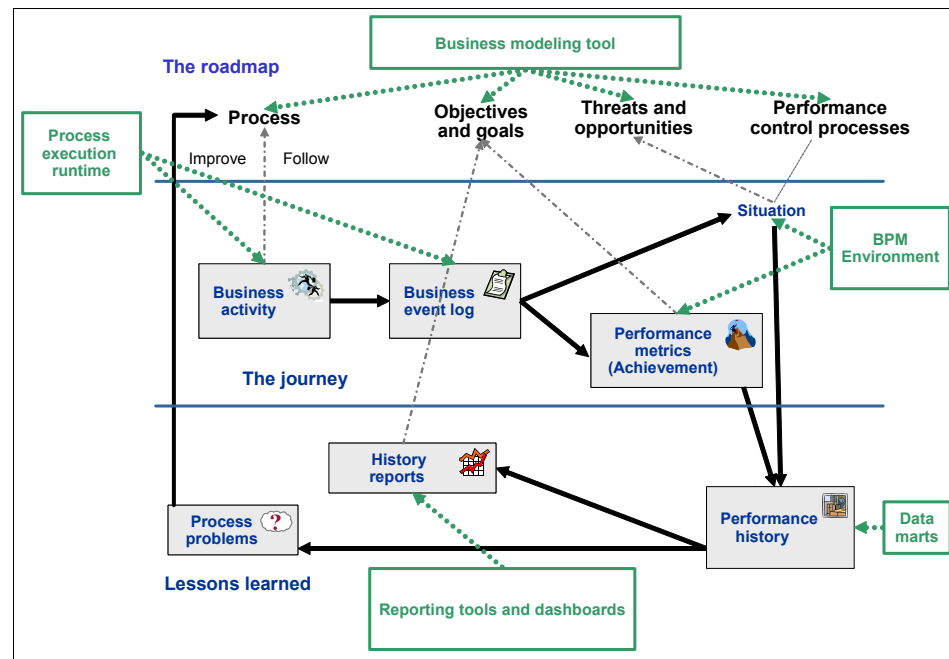


Figure 9-2 Process monitoring using BPM-enabling solutions

9.1.3 Real-time process monitoring using IBM BPM solutions

IBM provides an integrated set of products that leverage open standards to apply the BPM concepts. These products, and their roles, are:

- Define performance measures

IBM WebSphere Business Modeler Advanced Edition enables a business user to define business measures using the business measures editor.

- ▶ Log process execution to CEI
IBM WebSphere Process Server can be configured to perform real-time logging for running process instances. Process server logs process execution events using the standard Common Base Event (CBE) format. Events are posted to the Common Event Infrastructure (CEI). The CEI enables different process instances to emit events that can be captured by business monitors.
- ▶ Monitor process performance
IBM WebSphere Business Monitor provides the capability to monitor the workflow performance from CEI data.

9.2 WebSphere Business Monitor capabilities

WebSphere Business Monitor V6.0, an enterprise application suite, measures business performance, monitors processes and workflow, and reports on business operations. The information that is captured can help you to identify problems, correct faults, and change processes to inject more efficiency into business operations.

Note: WebSphere Business Monitor can monitor business-level events, but not down to the IT level. It can be used to track bottlenecks in a business process but you need to use other tools to track IT performance, such as IBM Tivoli Composite Application Manager for SOA.

WebSphere Business Monitor logs process execution at runtime in CBE format. At the time of writing, the only CBE-compliant runtime engine that could be used with WebSphere Business Monitor is WebSphere Process Server V6.0.1.

WebSphere Business Monitor enables an administrator to create a collection of portlets and views that display performance analysis and reporting. This collection of portlets is known as a *dashboard*.

Figure 9-3 on page 142 shows the relationship between WebSphere Business Monitor, WebSphere Process Server, and WebSphere Business Modeler.

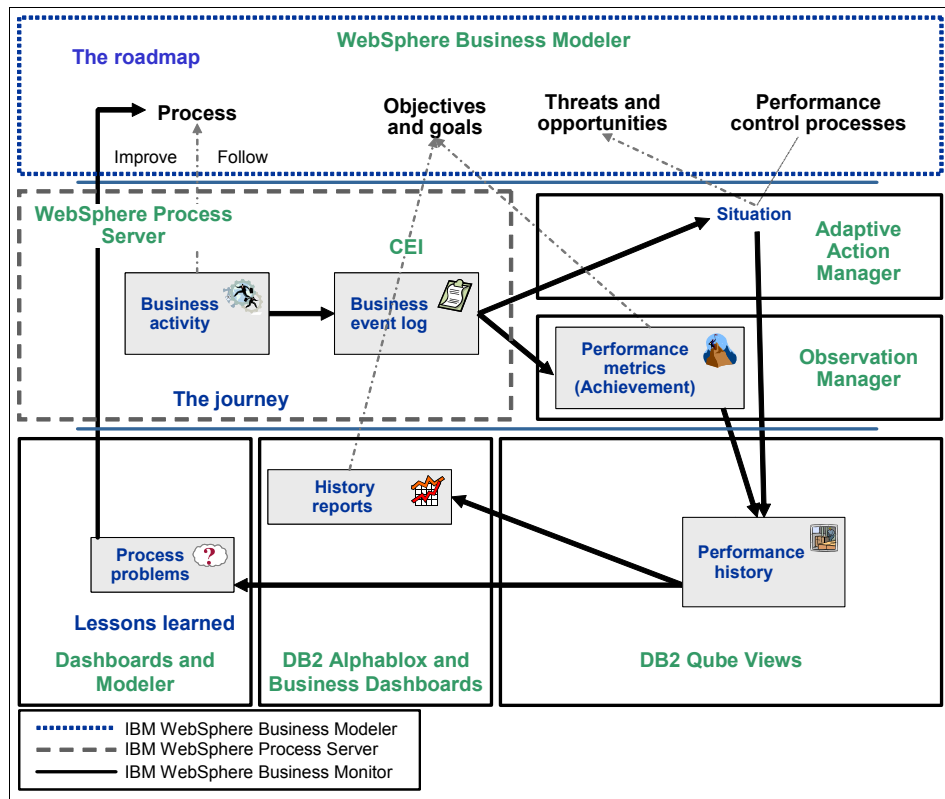


Figure 9-3 IBM BPM Suite

9.2.1 Real-time business measure extraction

After a line of business manager defines process business measures or observation items of interest at a business level using IBM WebSphere Business Modeler V6, the modeled business measures can be exported into a WebSphere Business Monitor archive. This exported model provides WebSphere Monitor with answers to pertinent questions:

- ▶ Which process log events should be gathered by the WebSphere Business Monitor runtime?
- ▶ What are the business performance measures for this process?
- ▶ What event and measurement values represent a business situation?
- ▶ How are business measures extracted and business situations detected from the gathered events?

Using this information, WebSphere Business Monitor performs these actions:

- ▶ Receive process execution events from CEI.
- ▶ Filter-in events specified by the business measures model.
- ▶ Use the model to calculate and update business measures based on the input from the filtered events.
- ▶ Use the model to detect business situations. When detected, a situation event will be submitted to CEI for further situation handling.
- ▶ Periodically publish the updated business measures into a historical data mart for further historical analysis.

9.2.2 Business situation handling

WebSphere Business Monitor enables users to handle business situation events that are defined in WebSphere Business Modeler V6 and detected by the Monitor runtime. A WebSphere Business Monitor administrator can define an appropriate action to handle a given business situation. A WebSphere Business Monitor user might handle the situation by two types of actions:

- ▶ Notification actions

A notification should be sent when the situation occurs. Notification might take the form of e-mail, cell phone message, pager message, or a dashboard alert.

- ▶ Automated actions (service)

A specific service should be invoked in reaction to the situation. WebSphere Business Monitor invokes a Web service, or a WS-BPEL process through a Web service invocation.

A WebSphere Business Monitor administrator defines appropriate actions for business situations using the WebSphere Business Monitor administrative console. The Adaptive Action Manager is the component responsible for handling business situation events.

9.2.3 Performance analysis and reporting

Business users and executives need to view runtime performance information for running business processes as well as to perform historical analysis and make performance reports and charts. Users can display the monitoring data through a set of dashboard views defined according to display properties and content.

A dashboard view is a type of display supported by one or more categories. It is targeted to address specific functions or responsibilities of a particular role. Each view type supports a number of properties that customize the display.

WebSphere Business Monitor dashboards:

- ▶ Help to monitor and manage business performance
- ▶ Help executives retrieve information quickly and efficiently
- ▶ Enable executives to personalize the analysis and display of business performance reports to focus on the key performance indicators (KPIs)
- ▶ Provide a graphical display of business-critical information, using visual cues such as color to help improve problem determination and decision making
- ▶ Visualize performance data such as KPIs and metrics, which can be summarized in reports and graphs
- ▶ Analyze and investigate business situations by using the drill-down capabilities to trace situations to individual events and inspect event details
- ▶ Support actions and alerts that are part of managing business performance

The dashboard contains many view types, which are discussed in this section.

Active Instances view

The Active Instances view shows the details of a process at runtime (Figure 9-4). It displays information about running instances as they happen, the instances of a specific process, and the runtime values of selected business measures.

Actions	Diagram	City 	Country	Location type	Loyalty Club ID	Number of times overdue	Requested vehicle class
		Manchester	England	Airport		1	A
		Raleigh	USA	Suburbs		0	C

Figure 9-4 Active instances view

Reports view

The Reports view (Figure 9-5 on page 145) displays performance reports about historical values of aggregate business measures relative to a time axis. The Reports view helps you to analyze historical values of numeric business measures that belong to a subject area (business measures group). The analysis is typically represented in tables and graphs.

Within this subject area, you can use business measures as dimensions to further break down the business measures values that are being analyzed. The business measures must already be defined as dimensions in the business measures model. This view requires a business measures model that contains one or more subject areas. Each can contain one or more dimensions for further analysis.

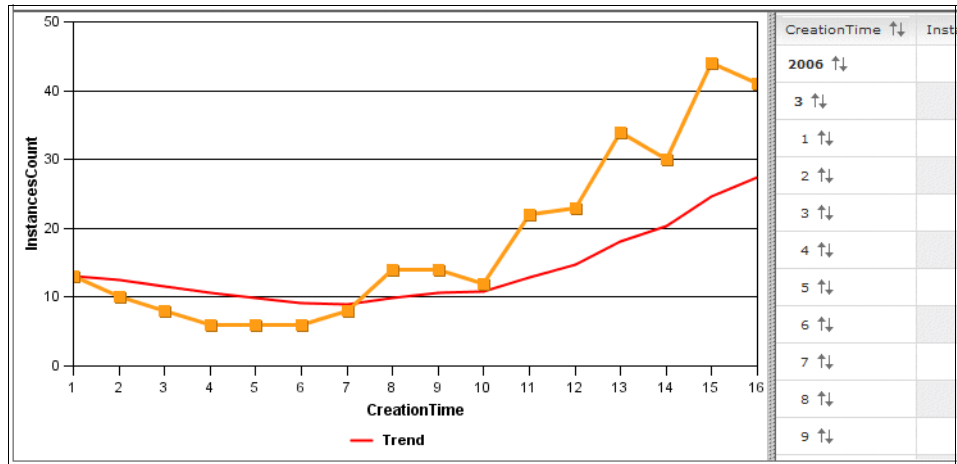


Figure 9-5 Reports view

Scorecards view

The Scorecards view (Figure 9-6) enables users to monitor groups of KPIs based on some configurable business perspectives, where for each KPI, information relative to its target value is available. A scorecard is a set of performance measures that are linked to objectives and goals of a specific business area. Business users select the KPIs pertaining to their various areas of responsibility and place them in perspectives (categories) on their scorecards. On the Scorecards view, users can easily watch the runtime values of vital numeric KPIs against their defined targets. This view requires a business measures model that contains numeric KPIs with defined targets. For KPIs with non-numeric values, the Scorecards view displays the KPI value and an N/A (not applicable) icon instead of KPI status.

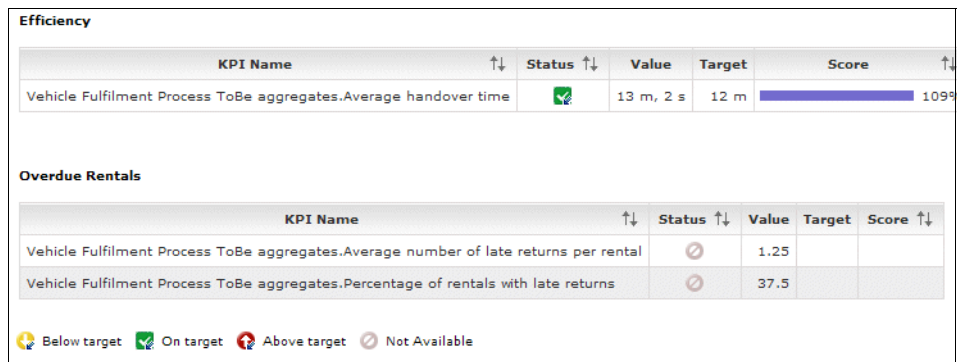


Figure 9-6 Scorecards view

Key performance indicators (KPIs) view

The KPI view (Figure 9-7) displays all KPI information so that business users can easily monitor them and take action if necessary.

KPIs are quantifiable measurements of the improvement or deterioration in the performance of an activity critical to the success of a business. These measurements break down key areas of your business to show how they contribute to business results. KPIs differ depending on the business. For example, in telesales, the timely answering of customer calls is a key business activity. The percentage of calls answered within the first minute can be one of its KPIs.

KPI	↑↓	Status ↑↓	Value
Vehicle Fulfilment Process ToBe aggregates.Average handover time		⊘	13 m, 2 s
Vehicle Fulfilment Process ToBe aggregates.Average number of late returns per rental		↑	1.25
Vehicle Fulfilment Process ToBe aggregates.Percentage of rentals with late returns		↑	37.5

⬇ Below limit ✓ Within limits ↑ Above limit ⊘ Not Available

Figure 9-7 KPI view

Gauges view

The Gauges view (Figure 9-8) visually represents the values of KPIs relative to KPI limits or relative to the KPI target, in the form of a gauge that looks like an automobile speedometer or tachometer.

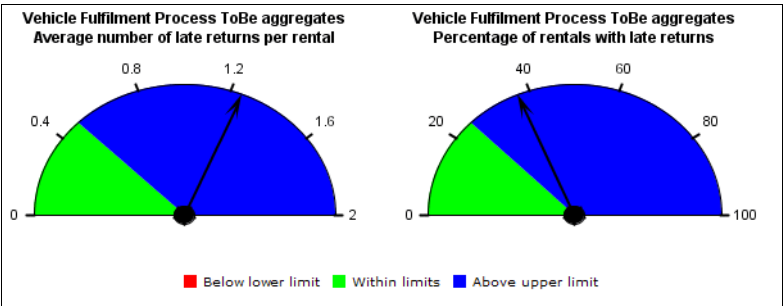


Figure 9-8 Gauges view

Dimensions view

With the Dimensions view (Figure 9-9), users can generate multidimensional reports that analyze different business aspects of historical business-performance data. Charts and grids present data for analysis against different dimensions.

A dimension is a conceptual axis over which a business is analyzed. For example, a retail business performance might be analyzed by time, products, and stores dimensions. Each of the dimensions has one or more levels that together define the overall hierarchy of the dimension. The time dimension might have year, quarter, and month levels.

Assume that you have collected a lot of data (for example, sales figures for every product your company makes), and you need to retrieve information from this data to find answers to questions such as:

- ▶ What are the total sales for each product by store?
- ▶ Which products are selling best over time?
- ▶ Who is your highest-performing salesperson?

To answer these and other questions, you can use the Dimensions view. It extracts, organizes, and summarizes your data. You can use it to analyze the data, make comparisons, detect patterns and relationships, and discover trends. This view requires a business measures model that contains defined dimensions and metrics, which are needed for the multidimensional report for each subject area (business-measures group).

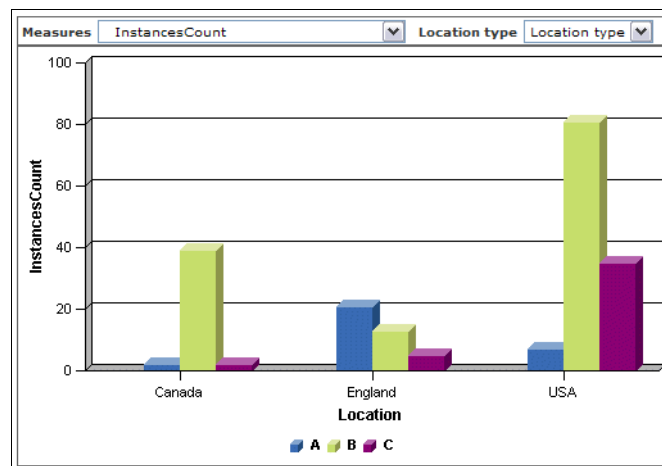
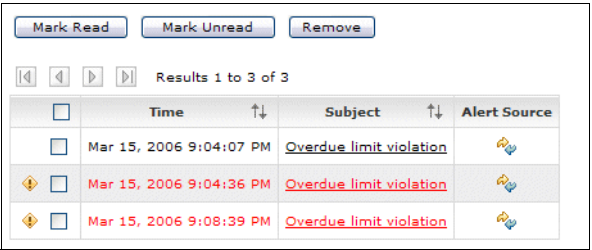


Figure 9-9 Dimensions view

Alerts view

The Alerts view (Figure 9-10) displays the alert notifications for a specific user. Alerts are one of the ways by which business users can be notified in case of business situation. A WebSphere Business Monitor administrator can configure the Adaptive Action Manager to respond to a given business situation by sending a dashboard notification (*alert*), which could be viewed later by the notified user in their Alerts view.



The screenshot shows the Alerts view interface. At the top, there are three buttons: "Mark Read", "Mark Unread", and "Remove". Below these are navigation icons and the text "Results 1 to 3 of 3". The main part of the interface is a table with the following columns: a checkbox, "Time", "Subject", and "Alert Source". There are three rows of data, all with the subject "Overdue limit violation".

<input type="checkbox"/>	Time	Subject	Alert Source
<input type="checkbox"/>	Mar 15, 2006 9:04:07 PM	Overdue limit violation	
<input type="checkbox"/>	Mar 15, 2006 9:04:36 PM	Overdue limit violation	
<input type="checkbox"/>	Mar 15, 2006 9:08:39 PM	Overdue limit violation	

Figure 9-10 Alerts view

Process Diagram view

The Process Diagram view (Figure 9-11) displays process status on a process graph with visual indicators showing the status of the process. You can also display the status of a specific process instance in the process graph by highlighting the activities that have been performed in the instance.

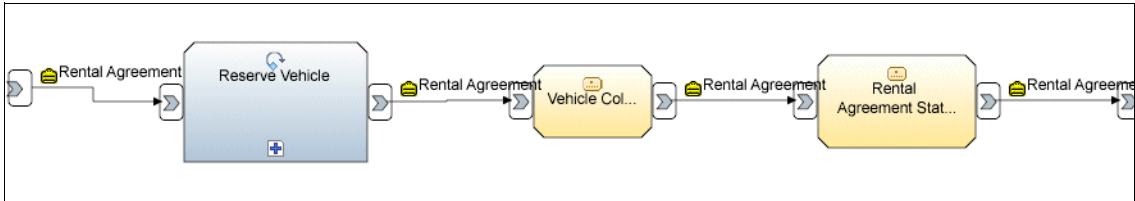


Figure 9-11 Process Diagram view

Organizations view

The Organizations view (Figure 9-12) retrieves organization and employee information from a user registry such as LDAP. You can select an organization or a set of employees to use to filter the data of the Reports view.

The screenshot displays the 'Organizations view' interface. At the top, a tree view lists various organizations: Admins, Big Organization, Bright People, Engineering, Other People, Stupid People, TestUsers, Training, academy, anderson council, barbers, beatles, and black sabbath. Below this, there are search filters: 'Search for:' with a dropdown set to 'Employee', 'Organization:' with an empty text box, '*Last name:' with the text 'cooper', and 'First name:' with an empty text box. A 'Search' button is located to the right of the first name field. At the bottom, a table lists employees with columns for 'Name' and 'E-mail'. The table contains six rows of data, each with a name and an email address, and a small icon to the right of each row.

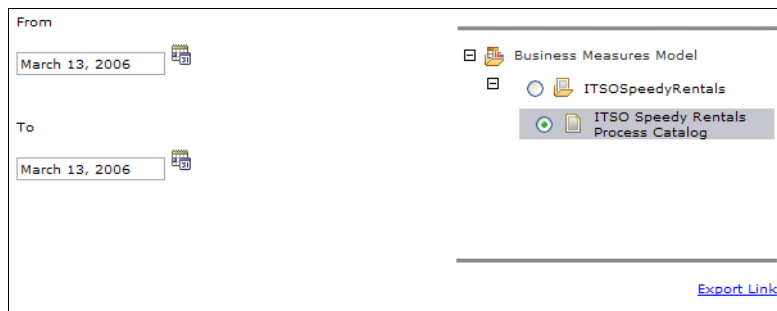
Name	E-mail
Cooper, Walter	WZCooper0082@IBM.com
Cooper, Howard	HGCooper0064@IBM.com
Cooper, Paul	PYCooper0067@IBM.com
Cooper, Gaylord	GKCooper0094@IBM.com
Cooper, Freddy	FZCooper0054@IBM.com
Cooper, Judith	JFCooper0049@IBM.com

Figure 9-12 Organization view

9.2.4 Using historical data for continuous process improvement

Line of business managers need feedback from executed business processes to perform process improvement. WebSphere Business Monitor provides this capability with the *Export Values dashboard* (Figure 9-13 on page 150), which enables further analysis and manipulation of processes information by exporting

the average values for a selected business measures model. Those values are then imported in WebSphere Business Modeler for continuous improvement of the business processes.



The screenshot shows a web interface for exporting data. On the left, there are two date selection fields. The top field is labeled "From" and contains the date "March 13, 2006". The bottom field is labeled "To" and also contains the date "March 13, 2006". On the right side, there is a tree view showing a hierarchy of business measures models. The root node is "Business Measures Model", which has a sub-node "ITSOSpeedyRentals". Under "ITSOSpeedyRentals", there is a node "ITSO Speedy Rentals Process Catalog" which is currently selected and highlighted with a grey background. At the bottom right of the interface, there is a blue hyperlink labeled "Export Link".

Figure 9-13 Export values view



Part 3

Scenario implementation



Business scenario and solution architecture

This chapter introduces the fictional company used in the working example of this book, ITSO Speedy Rentals. This fictional company is based on a vehicle rental agency, and we focus on their need to construct a vehicle fulfilment process.

This chapter contains the following information:

- ▶ Describes the business model of ITSO Speedy Rentals.
- ▶ Defines the IT requirements for a vehicle fulfilment process for ITSO Speedy Rentals.
- ▶ Creates a solution architecture using a process to determine the SOA scenario and Patterns for e-business that apply to the development of a vehicle fulfilment process.
- ▶ Summarizes how the ITSO Speedy Rentals vehicle fulfilment process was modeled, assembled, deployed, and managed, including links to appropriate chapters containing step-by-step instructions for completing those tasks.

10.1 Business model

This section provides an initial overview of the fictional ITSO Speedy Rentals business model, explains the existing business process, rules, and measures, and defines the roles in ITSO Speedy Rentals.

10.1.1 Initial context

The ITSO Speedy Rentals company is a vehicle rental agency that conducts business from 1,500 locations worldwide. The company offers a variety of current-model cars on a short-term rental basis (daily, weekly, and monthly) at locations including airports, residential areas, and resorts.

ITSO Speedy Rentals has a manual, and poorly documented, vehicle fulfilment business process that enables a customer arriving at a rental location to rent and return a car from the vehicle inventory.

Currently ITSO Speedy Rentals does not maintain any business processes in formal notation, and throughout the branches, processes are followed using outdated company training material.

The business wants its operation processes to be defined in a manageable and understandable formal notation. A documented analysis of the processes should reveal, for example, potential bottlenecks, cost savings, customer satisfaction improvements, and so forth. Based on this information, a new business process implementation should be built, taking advantage of the latest SOA technologies.

10.1.2 Existing business process

The existing ITSO Speedy Rentals process for vehicle fulfilment is:

1. Customer reserves a vehicle.

The customer may reserve a vehicle for rental through several channels: online, a call center, or a local branch office. Customer details are entered into a head-office reservation system, and a rental agreement process is initiated.

2. Vehicle is reserved for the customer.

The vehicle is selected based on the customer's requests and the availability of the vehicle model for a selected rate group. If the vehicle is not available for the selected group, then a vehicle from the next group level will be selected.

3. Vehicle is collected or delivered.

The customer can either collect the vehicle from the local branch or have the vehicle delivered to a specified address.

4. Customer drives the vehicle.

The customer has possession of the vehicle for the agreed duration. The customer can choose to return the vehicle before or after the agreed time limits. The local office must be informed of a late return and update the rental agreement.

5. Customer returns the vehicle.

The customer returns to a specified local branch to return the vehicle. A local branch agent inspects the vehicle for possible additional surcharges such as fuel, additional mileage, and damage. All additional charges are included in a final invoice and given to the customer. This final invoice may differ from the estimated rental cost.

6. Invoice is generated.

The invoice either can be generated by the local office at the time of vehicle return or recalculated based on any additional charges such as additional days used, mileage charges, and so forth.

10.1.3 Business rules and business measures

The ITSO Speedy Rentals management team has identified the following business rules in their business process, and would like to track the following business measures during business process execution.

Business rules

The business rules for the ITSO Speedy Rentals vehicle fulfilment process are:

► Vehicle reservation

If the vehicle group requested by the customer is not available at time of collection, the customer will be allocated a vehicle from the next group level free of charge.

► Rental agreement

Any changes made by the customer or rental agent must be reflected in the vehicle rental agreement.

► Vehicle overdue notice

If a vehicle is not returned at the planned date and time, the customer must be contacted and a new return date, time, and location confirmed. The existing rental agreement should be updated with the new details.

► Loyalty discounts

A discount will be applied to ITSO Speedy Rentals loyalty club members at time of final payment calculation.

Business measures

ITSO Speedy Rentals wants to track and monitor several business measurements:

- ▶ Vehicle reservation

The time and cost to reserve a vehicle for a customer and have that vehicle delivered for customer pick-up.

- ▶ Vehicle overdue

The cost of overdue vehicles returned to a branch. This measurement should break out by location (such as airport or downtown) and reflect resource cost per branch location.

- ▶ Vehicle return

The time taken for the customer to return a vehicle and complete the necessary paperwork. The measurement should show the cost and time per branch location.

10.1.4 Roles in the organization

Figure 10-1 shows the ITSO Speedy Rentals organization structure.

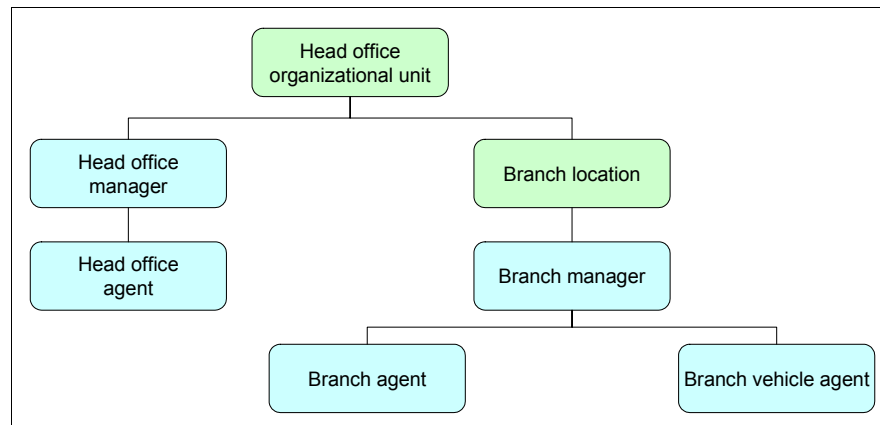


Figure 10-1 ITSO Speedy Rentals organization chart

The roles in this organization are:

Head office manager Responsible for all local branches.

Head office agent Manages payment transactions from local branches.

Branch manager Responsible for all upgrade requests and managing branch staff. Might also process customer rentals during busy periods.

Branch agent	Processes the rental agreement and ensures that the vehicle is reserved and prepared for the customer.
Branch vehicle agent	Responsible for preparing the vehicle, delivering the vehicle to the customer for pick-up, and the handling of all returned vehicles.

10.2 IT requirements

This section introduces functional and non-functional requirements for the ITSO Speedy Rentals scenario. We also show a high-level system context diagram for the proposed IT solution.

10.2.1 Functional requirements

We have outlined the following key functional IT requirements:

- ▶ **FR1: Document the existing business process in a modeling tool.**
Build a model of the existing ITSO Speedy Rentals business process, outlining the tasks in the process and ownership of the tasks. An IT modeling tool can be used to analyze this documented business process.
- ▶ **FR2: Calculate the cost and time of each step in the business process.**
By running simulated workloads through the documented business process, analyze the cost and time required for each step, identify bottlenecks, and consider areas for improvement.
- ▶ **FR3: Create a new business process using WS-BPEL, Web services, Java objects, business rules, and human tasks.**
Construct a new business process that includes the identified process improvements, using SOA technologies to execute a business process and interact with Web services.
- ▶ **FR4: Over time, identify improvements to the WS-BPEL process, implement them, and redeploy.**
Using a monitoring tool, monitor process instance execution, and use this data to identify process improvements. Model, assemble, and redeploy new business processes that implement these improvements.

10.2.2 Non-functional requirements

We have outlined the following non-functional IT requirement:

- NR1: Monitor process execution and capture business metrics.

To enable continuous business process improvement, monitor every instance of the vehicle fulfilment process, and capture business metrics for analysis. This data can be used to create improvements to the business process implementation over time.

For simplicity, we do not address other typical non-functional requirements such as security, transactionality, performance, and availability.

10.2.3 System context diagram

Based on the IT requirements outlined in this section, a modeling tool is used to assemble the business process shown in Figure 10-2. This business process uses WS-BPEL and will invoke SCA services including Web services, business rules, Java objects, and human tasks. This WS-BPEL business process will be monitored during execution.

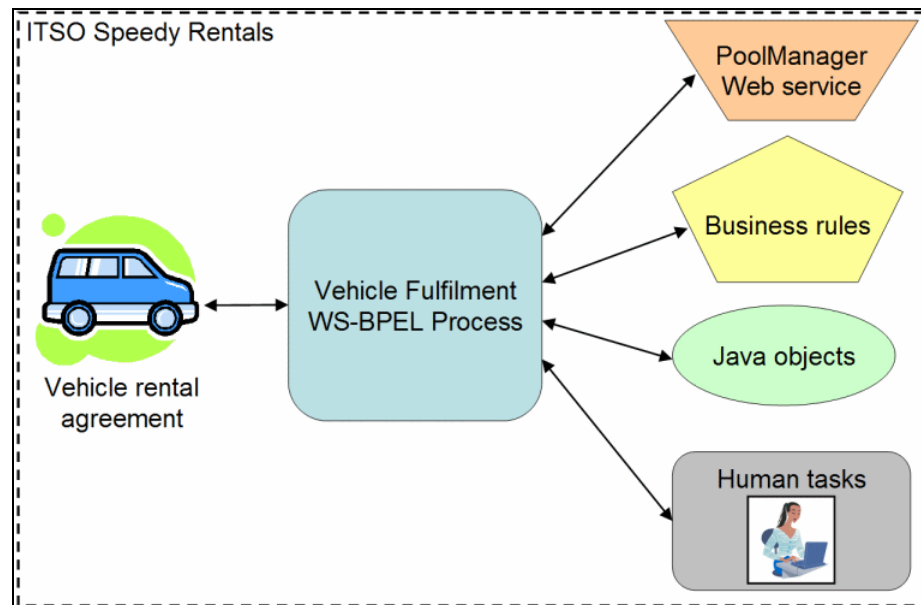


Figure 10-2 System context diagram

10.3 Solution architecture

This section demonstrates how to use the process for applying SOA scenarios and patterns to accelerate the design of the solution architecture for the ITSO Speedy Rentals scenario. It follows the process shown in Figure 10-3.

Note: For information about this process, see Chapter 4, “Process for applying SOA scenarios” on page 43.

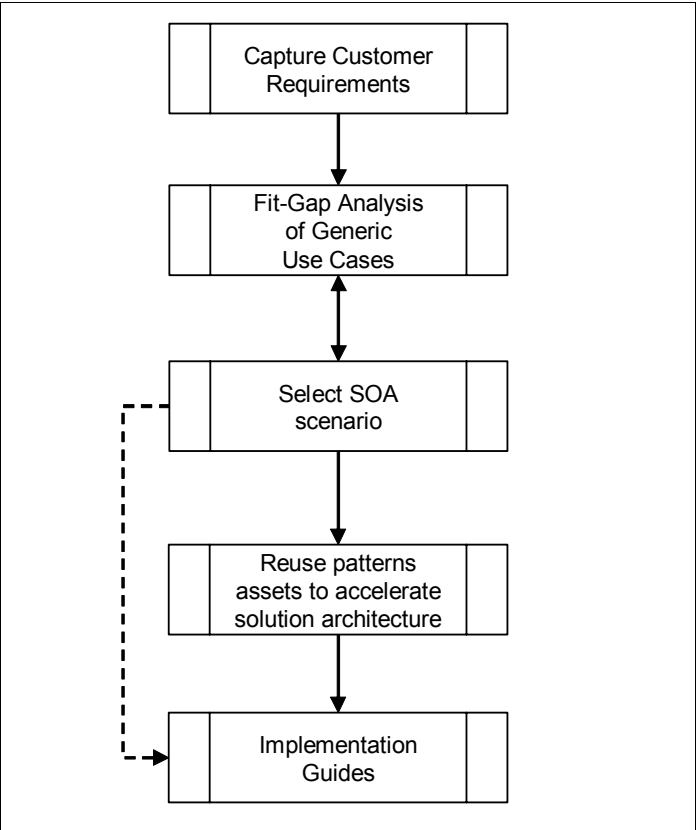


Figure 10-3 Process for using SOA scenarios and patterns

10.3.1 Apply fit gap analysis to ITSO Speedy Rentals

Table 10-1 on page 160 provides a summary of the functional requirements of ITSO Speedy Rentals, and the generic use cases they apply to.

Table 10-1 ITSO Speedy Rentals fit gap analysis

ITSO Speedy Rentals functional requirements	Generic use case
FR1: Document the existing business process in a modeling tool.	U9: Improve an existing business process flow through business process and policy modeling and simulation.
FR2: Calculate the cost and time of each step in the business process.	
FR3: Create a new business process using WS-BPEL, Web services, Java objects, business rules, and human tasks.	U10: Implement a new business process flow.
FR4: Over time, identify improvements to the WS-BPEL process, implement them, and redeploy.	U11: Analyze existing business process flow using monitoring.

10.3.2 Select the SOA scenario

At this stage in the process, the generic use cases that apply to the ITSO Speedy Rentals scenario have been identified. Now the generic use cases are used as criteria to select the appropriate SOA scenario containing software capable of fulfilling the requirements. This task is accomplished by using the SOA scenario selection table.

The full SOA scenario selection table is provided in 4.1.1, “SOA scenario selection table” on page 46. Table 10-2 shows a subset of this table, with the generic uses cases that fulfill the functional requirements of the ITSO Speedy Rentals scenario.

Table 10-2 SOA scenario selection criteria identifying the Business Process Management scenario

Generic use cases selection criteria	SOA scenarios				
	Service Creation	Service Connectivity	Interaction and Collaboration Services	Business Process Management	Information as a Service
U9: Improve an existing business process flow through business process and policy modeling and simulation.				X	
U10: Implement a new business process flow.				X	
U11: Analyze existing business process flow using monitoring.				X	

As determined from Table 10-2, the appropriate SOA scenario for ITSO Speedy Rentals is the Business Process Management SOA scenario.

10.3.3 Reuse patterns assets to accelerate solution architecture

This section demonstrates how to apply the reusable assets found in the Patterns for e-business to accelerate the design of the solution architecture.

Select the Business or Application pattern

Chapter 5, “Patterns for e-business and Business Process Management” on page 49, identified the following Application Integration patterns as applicable to the Business Process Management scenario:

- ▶ Serial Process (and the Serial Workflow variation)
- ▶ Parallel Process (and the Parallel Workflow variation)

The following Extended Enterprise pattern was also identified as applicable to the Business Process Management scenario:

- ▶ Exposed Serial Process (and the Exposed Serial Workflow variation)

To determine the applicable pattern that applies to the ITSO Speedy Rentals scenario, ask the questions summarized in Figure 10-4.

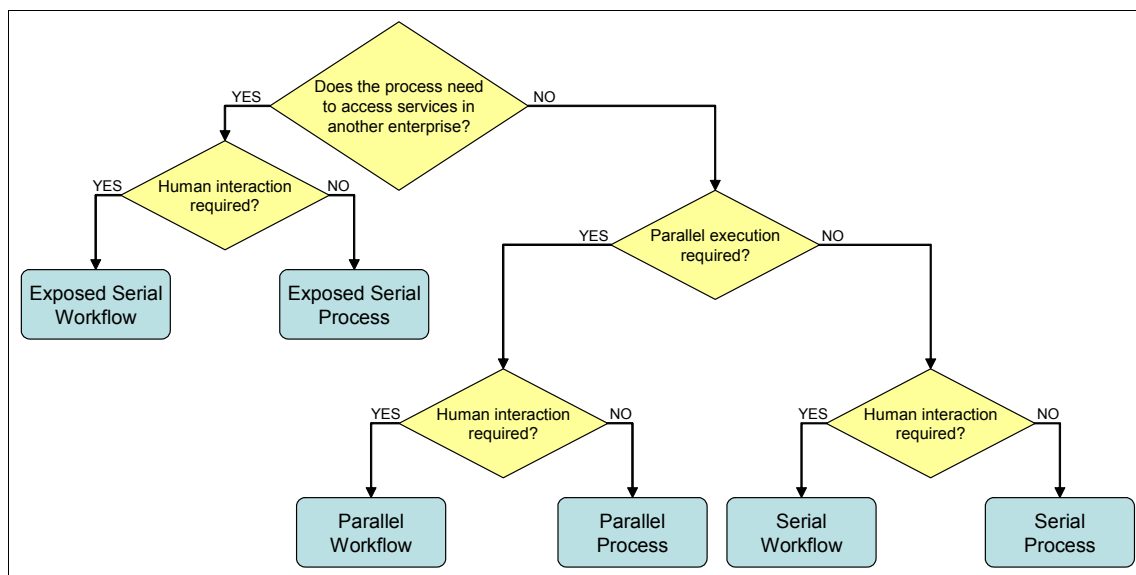


Figure 10-4 Pattern selection flow chart

1. Does the process need to access services in another enterprise?

The ITSO Speedy Rentals vehicle fulfilment process interacts with resources exclusively within the ITSO Speedy organization, so it does not need to

access other enterprises. The Extended Enterprise pattern is ruled out, and the search falls to the Application Integration pattern.

2. Is parallel execution required?

The ITSO Speedy Rentals vehicle fulfilment process is sequential in nature. One task has to complete before the next one can begin, thus ruling out the Parallel Process and Parallel Workflow Application Integration patterns.

3. Is human interaction required?

Several of the tasks in the ITSO Speedy Rentals vehicle fulfilment process require the interaction of a person. Therefore, human interaction is required, so the Serial Workflow variation is the correct choice of pattern for ITSO Speedy Rentals.

Runtime pattern and product mapping

The Serial Workflow Application pattern meets the needs of ITSO Speedy Rentals, so the Serial Workflow runtime pattern variation of Serial Process applies. This pattern is shown in 5.2.1, “Serial Process” on page 53.

A Runtime pattern can have many product mapping options. Figure 10-5 shows the product mapping implemented in this book for ITSO Speedy Rentals.

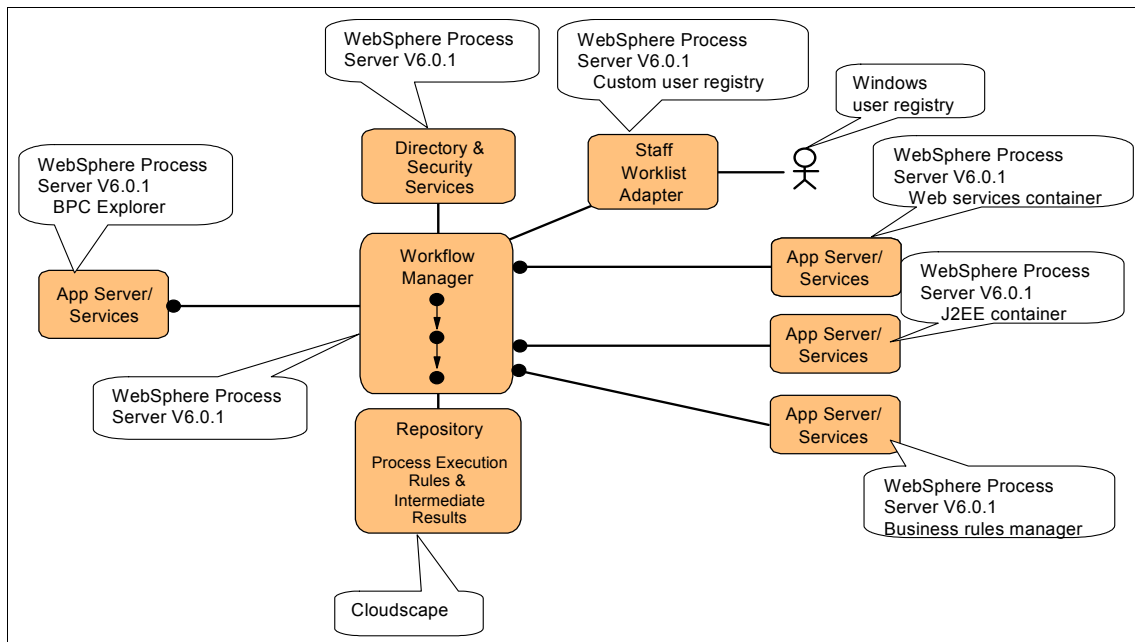


Figure 10-5 Serial Workflow: WebSphere Process Server product mapping

Note: Product mappings only show products used in the Deploy part of the SOA lifecycle. The products used for the Model, Assemble, and Manage SOA lifecycle stages are not pictured in a product mapping. Determine which products to use for these other stages by examining the selected SOA scenario, which in this case is Business Process Management.

10.4 Implementation details

This section provides an overview of the development of the ITSO Speedy Rentals scenario solution, and points you to the relevant chapters to find step-by-step instructions about how to construct this scenario.

10.4.1 Model

In the Model stage of the SOA lifecycle, perform the following tasks for ITSO Speedy Rentals using WebSphere Business Modeler V6:

1. Build a model of the current vehicle fulfilment process, identifying the tasks and resources used.
2. Simulate workload through the current vehicle fulfilment process, and analyze the results of this simulation to identify improvements to the process.
3. Model a new process that implements these improvements and is built with a view to deploying the model to WebSphere Process Server.
4. Create a business measures model for the new process.
5. Export the model and business measures model.

Note: For step-by-step instructions for completing these tasks, see Chapter 11, “Model with WebSphere Business Modeler” on page 167.

10.4.2 Assemble

In the Assemble stage of the SOA lifecycle, perform the following tasks for ITSO Speedy Rentals using WebSphere Integration Developer V6:

1. Import the new process model from WebSphere Business Modeler and add implementation details and logic to it to create WS-BPEL processes.
2. Add Web services, a Java component, and business rules invocation.
3. Configure human tasks.
4. Test the business process.

5. Export the business process.

Note: For step-by-step instructions for completing these tasks, see Chapter 12, “Assemble with WebSphere Integration Developer” on page 263

Components of the WS-BPEL business process

Figure 10-6 shows the relationships of each of the tasks in the business process you will assemble, and how they will be implemented.

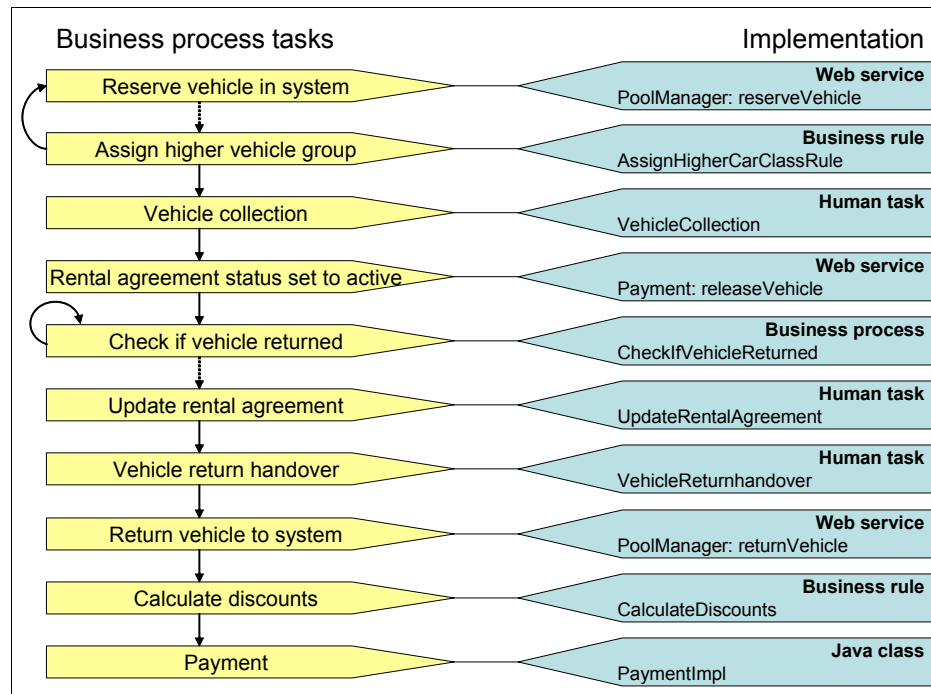


Figure 10-6 Business process tasks and implementation

Components of the PoolManager Web service

The WS-BPEL business process you will assemble uses the PoolManager Web service. This is a pre-existing Web service that you will not have to create or modify. It is used to update the status of a given vehicle in the Reserve database.

The PoolManager Web service is built from a session bean, and two entity beans that expose tables in the Reserve database, as shown in Figure 10-7 on page 165.

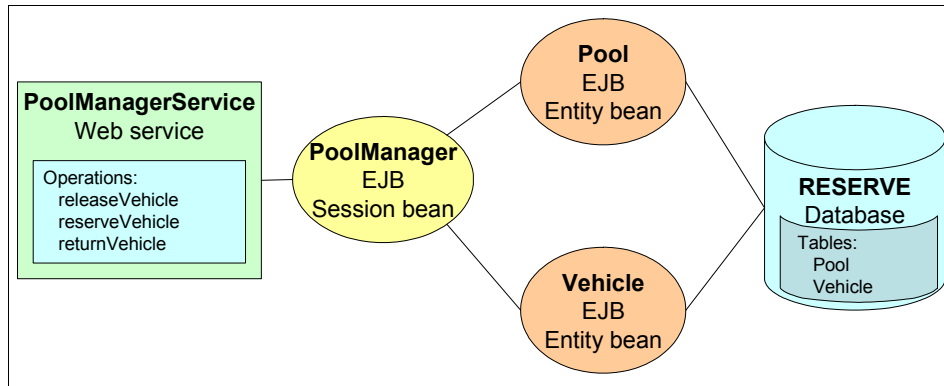


Figure 10-7 PoolManager Web service

10.4.3 Deploy

In the Deploy stage of the SOA lifecycle, you will perform the following tasks for ITSO Speedy Rentals using WebSphere Process Server:

1. Prepare the server environment and deploy the vehicle fulfilment process to the server.
2. Test the business process.
3. Use the Business Rules Manager to dynamically change business rules.
4. Use a selector to dynamically change the implementation of the payment task.

Note: For step-by-step instructions for completing these tasks, see Chapter 13, “Deploy with WebSphere Process Server” on page 333.

Selector for the payment task

A selector is used for the payment task to enable the implementation of this task to be dynamically changed without having to reassemble or deploy the process. Using a selector, change the implementation of the payment task to one of three implementation types, as shown in Figure 10-8 on page 166.

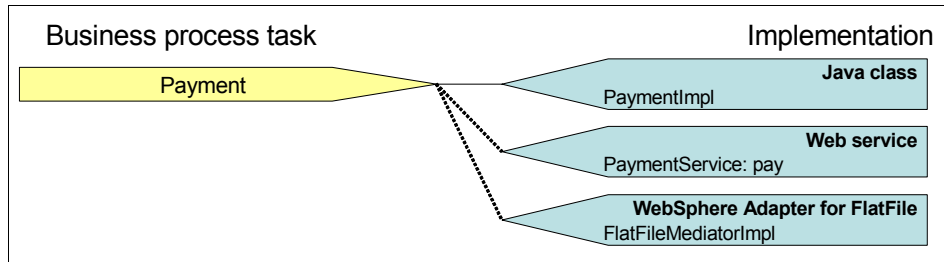


Figure 10-8 Payment implementation types

10.4.4 Manage

In the Deploy stage of the SOA lifecycle you will perform the following tasks for ITSO Speedy Rentals using WebSphere Business Monitor:

1. Import the business measures model.
2. Create custom dashboards.
3. Monitor the process.

Note: For step-by-step instructions for completing these tasks, see Chapter 14, “Manage with WebSphere Business Monitor” on page 353.

10.4.5 Continuous improvement

With the Model, Assemble, Deploy, and Manage stages of the SOA lifecycle complete, perform another interaction of the SOA lifecycle as follows:

1. Analyze the information captured by WebSphere Business Monitor in the Manage stage of the SOA lifecycle.
2. Using this information, look for areas of improvement. In the case of ITSO Speedy Rentals, this analysis showed a need to add an additional task and create additional business measures. To add them, return to the Model stage and update the model in WebSphere Business Modeler.
3. Export the new model, and import it into WebSphere Integration Developer for the new Assemble phase.
4. Export the assembled business process and deploy it to WebSphere Process Server for the new Deploy phase.
5. Monitor the new business process (with the additional business measures and task) in WebSphere Business Monitor in the new Manage phase.

Note: For step-by-step instructions for completing these tasks, see Chapter 15, “Continuous process improvement” on page 417.



Model with WebSphere Business Modeler

This chapter describes how we created our business process model for the ITSO Speedy Rentals scenario using WebSphere Business Modeler. It discusses:

- ▶ **Creating the As-Is process model**
Describes how to document the business process in place at ITSO Speedy Rentals in WebSphere Business Modeler.
- ▶ **Simulating the As-Is current process model**
Describes how to run simulation of the As-Is model to identify problems with the current business process.
- ▶ **Creating the To-Be process model**
Based on the lessons learned from the simulation of the As-Is process, model a new process for implementation in WebSphere Process Server.
- ▶ **Modeling for WebSphere Process Server**
Specific additions to the model for WebSphere Process Server.
- ▶ **Creating the business measures model**
Define business measures for use in WebSphere Business Monitor.
- ▶ **Exporting the To-Be model to WebSphere Integration Developer and WebSphere Business Monitor**

11.1 Overview of the process

In this chapter the approach is to create two models: the “As-Is” model that represents the businesses current business processes, and a “To-Be” model that defines the new business processes to be deployed. Figure 11-1 shows our modeling approach.

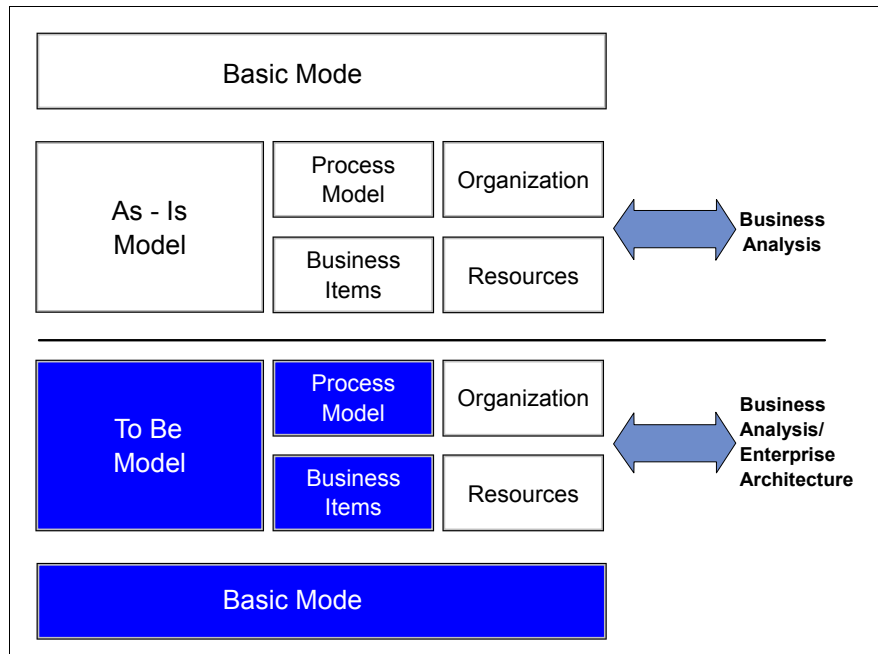


Figure 11-1 Modeling approach

This As-Is model is our *analysis model*; that is, its main purpose is to document the current business processes and provide a level of understanding to the business analyst and to the enterprise architect before they begin to develop the implementation To-Be business model.

Both the business analyst and the enterprise architect are responsible for the To-Be implementation business model, although the main effort will be by the enterprise architect who will ensure that the model is ready for export to WebSphere Process Server.

The scenario builds two projects in WebSphere Business Modeler: one for the As-Is model and one for the To-Be model. After the As-Is model is completed, elements of this model are reused to build the To-Be project, such as the organization catalog, resource catalog, and business items.

11.2 Modeling the current business process

When creating the business process model, you should understand the current business processes in operation. First the business analyst investigates the current business processes at ITSO Speedy Rentals. This normally involves staff interviews and reviewing available documentation (such as training material, use cases, operation manuals, and so forth).

For the purpose of this book, assume that a business analyst has carried out sufficient analysis of ITSO Speedy Rentals to produce a detailed model of the current vehicle fulfilment process, referred to here as the As-Is model.

To ensure that the model produced in WebSphere Business Modeler is accurate and correct, sketch the current process flow, document any business items (documents, products, records), list all key resources (people, equipment, material), and list any known business rules (decision logic) that are associated with that flow.

Documenting the As-Is process

The business analyst's first step is to gain an understanding of the vehicle fulfilment process. The main goal during this phase is to record:

- ▶ Process flow and the main activities
- ▶ Required business items
- ▶ Resources used

Upon reviewing the flow, record:

- ▶ Main process tasks
- ▶ Descriptions of each task
- ▶ Data required for each task, including the inputs and outputs of each task
- ▶ Possible alternative tasks
- ▶ Roles and resources associated with each task

Note: This chapter refers to a task as being a basic building block representing an activity in a process, and each task will perform some function.

The next step is to document the process flow, using a document editor or simply sketching it with paper and pen. Figure 11-2 on page 170 shows the Vehicle Fulfilment process flow for the ITSO Speedy Rentals scenario.

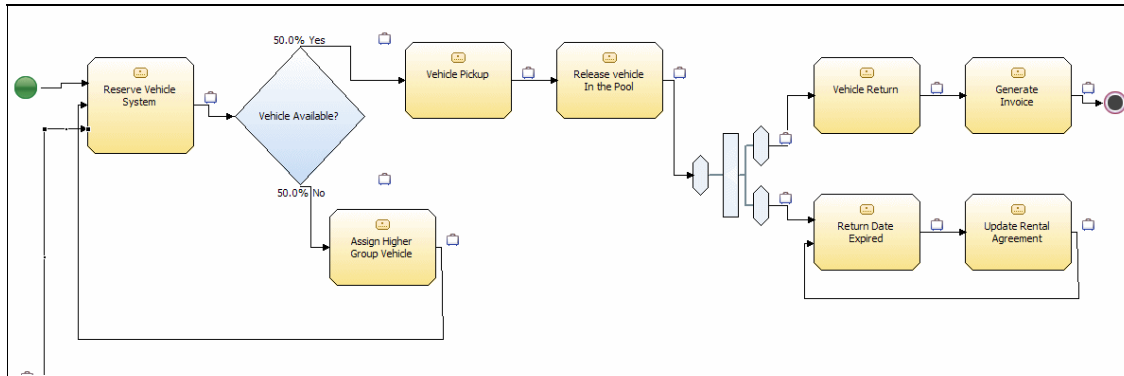


Figure 11-2 Vehicle Fulfillment process flow

11.3 Creating the As-Is process model

The As-Is process model represents the current business process. Using WebSphere Business Modeler to model this business process offers the following advantages:

- ▶ Ability to simulate the current business process to highlight potential problems and bottlenecks.
- ▶ Ability to use the As-Is model as the basis for the To-Be model, which can then be exported to WebSphere Integration Developer for assembly.

This section describes how to use WebSphere Business Modeler for:

- ▶ Configuring WebSphere Business Modeler
- ▶ Creating the As-Is process flow
- ▶ Creating the business items
- ▶ Connecting tasks with associated data
- ▶ Adding classifiers
- ▶ Adding resources and roles
- ▶ Defining the organization
- ▶ Adding resources and organizations to the process

Tip: If you do not want to follow all of the steps in this section, you can import the completed As-Is model from the additional material supplied with this book in the \Modeler\As-Is\ directory. Import the file:

ITSO Speedy Rentals As-Is Project.zip

11.3.1 Configuring WebSphere Business Modeler

Note: The example in this chapter used WebSphere Business Modeler V6.0 with Interim Fix 3 installed.

Start WebSphere Business Modeler and configure the project used throughout this scenario:

1. Create the project workspace directory.
 - a. Double-click the **WebSphere Business Modeler** icon and start the application.
 - b. Enter the directory for storing your project files (Figure 11-3) and click **OK**.

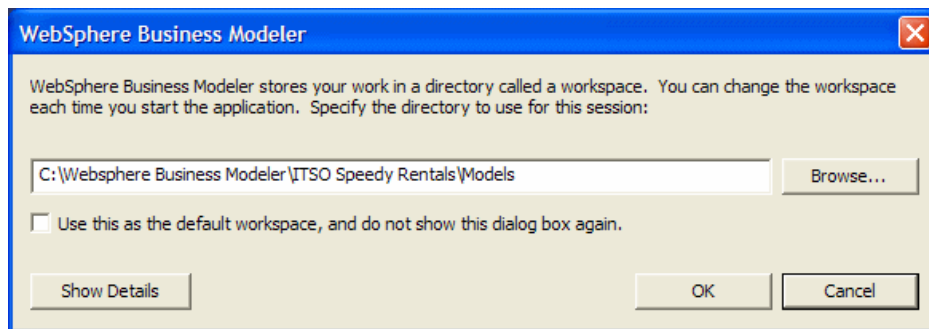


Figure 11-3 Select directory for model

2. In the Quickstart wizard, create the project name, catalog, and process (Figure 11-4), then click **Next**.
 - a. Enter the Project name ITS0 Speedy Rentals As-Is Project.
 - b. Enter the Process catalog name ITS0 Speedy Rentals Process Catalog.
 - c. Enter the Process name Vehicle Fulfilment Process AsIs.

Tip: It is important to use a meaningful naming convention. This scenario maintains the named function (for example, the process catalog is named ITS0 Speedy Rentals Process Catalog).

Quickstart wizard

Start your model by specifying a project name, a process catalog name, and a name for the process to create in the catalog.

Specify the process, process catalog, and project names and then click Finish.

Project name:
ITS0 Speedy Rentals As-Is Project

Process catalog name:
ITS0 Speedy Rentals Process Catalog

Process name:
Vehicle Fulfilment Process AsIs

☒ Always show this wizard on startup

Launch Help

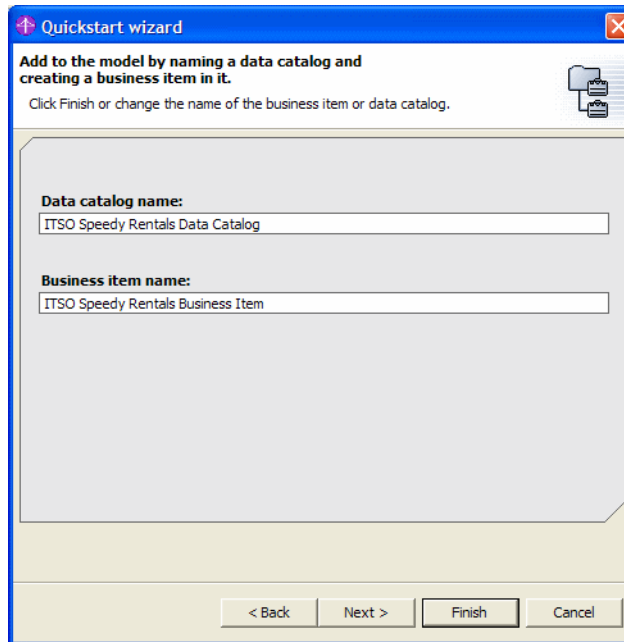
< Back Next > Finish Cancel

Figure 11-4 Quickstart Wizard for project and process naming

3. Create the data catalog and business item.

The data catalog holds any business items that are required by the process model. This scenario includes the full name of the functionality (Figure 11-5). Make the following entries then click **Next**.

- a. Enter the Data catalog name ITS0 Speedy Rentals Data Catalog.
- b. Enter the Business item name ITS0 Speedy Rentals Business Item.



The screenshot shows a 'Quickstart wizard' dialog box with a blue title bar. The main area is white and contains the following text: 'Add to the model by naming a data catalog and creating a business item in it.' Below this is a smaller line of text: 'Click Finish or change the name of the business item or data catalog.' To the right of this text is a small icon of a folder with a document. The main area has two text input fields. The first is labeled 'Data catalog name:' and contains the text 'ITS0 Speedy Rentals Data Catalog'. The second is labeled 'Business item name:' and contains the text 'ITS0 Speedy Rentals Business Item'. At the bottom of the dialog box are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Figure 11-5 Quickstart wizard for data and business item creation

4. Specify the default process editor layout.

Two editor layouts are available: free-form (used in this book) and swimlane. Select **Free-Form Layout** and click **Finish** (Figure 11-6).

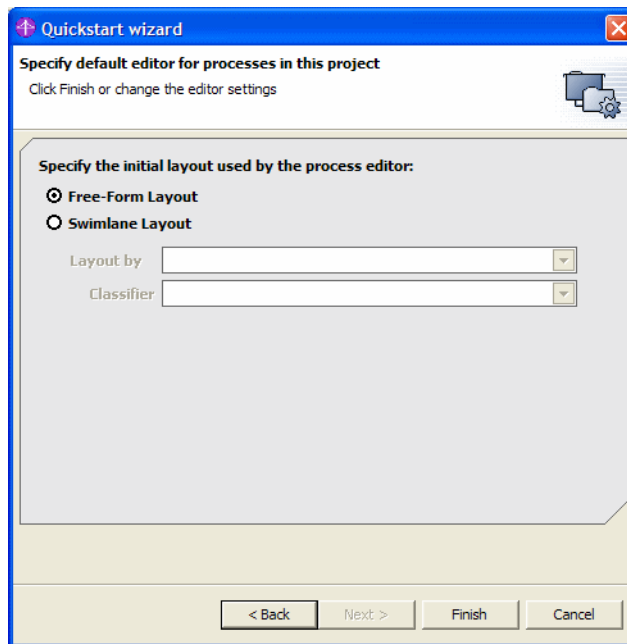


Figure 11-6 Quickstart wizard for process editor layout

5. WebSphere Business Modeler opens the process editor workspace, as shown in Figure 11-7.

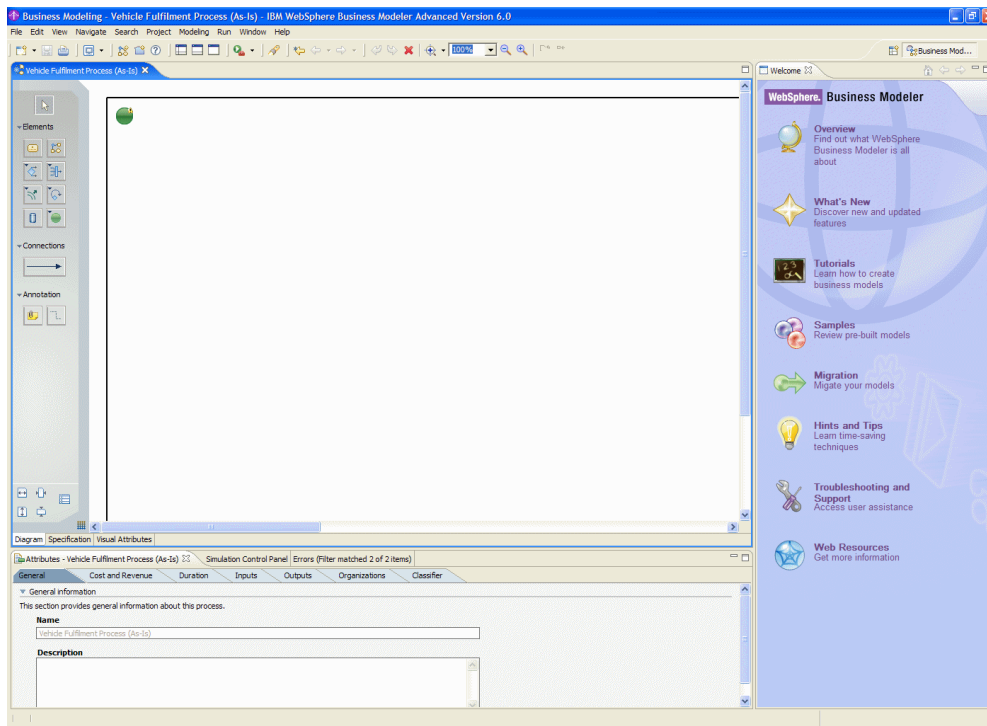


Figure 11-7 First view of editor

Note: Several role-based modes are available to the user. The process editor, by default, presets the mode to Basic. You can check your current mode by selecting **Modeling** → **Mode** from the menu.

11.3.2 Creating the As-Is process flow

WebSphere Business Modeler was used to create the As-Is process model, that is, the model of the current process. First, explore the WebSphere Business Modeler desktop and customize it to your requirements.

1. Customize the workspace, choosing either the 1-pane, 2-pane, or 4-pane layout.
 - a. Close the **Welcome** window for a larger workspace area.
 - b. Click the **4-pane layout** button as shown in Figure 11-8.

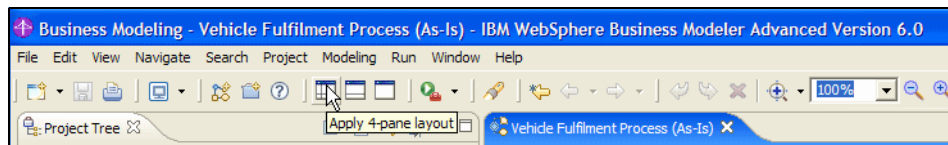


Figure 11-8 4-pane layout

Tip: 4-pane layout contains these views:

- Project Tree view
This provides a structured view of everything in your project. From here you can select elements and edit them.
- Process Editor view
Contains the editor for drawing your model. This is viewed in either free-form or swimlane format.
Whenever you want more room for the diagram, double-click the header tab of this view to maximize it to the entire WebSphere Business Modeler window. Reduce it back to its original size by double-clicking the view header tab again.
- Outline view
This view is for navigating the model being viewed in the model editor. For example, usually the model will be too large to see in the available workspace; this view enables you to select the area to be viewed.
- Attributes view
This view shows the details of any elements you select in the process model diagram. You can click each of the tabs to see more information about the element.

- c. In the project tree, expand **ITSO Speedy Rentals As-Is Project** (Figure 11-9).

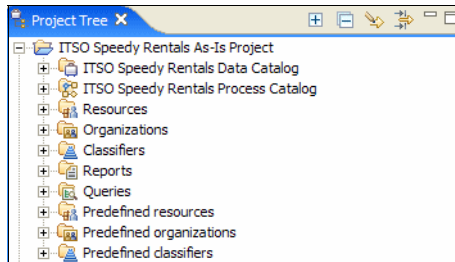


Figure 11-9 ITSO Speedy Rentals As-Is project tree

The project tree contains:

- ITSO Speedy Rentals Data Catalog
Performs the function of a folder, enabling you to group a related set of business items and notifications, created to model your business operations.
- ITSO Speedy Rentals Process Catalog
Performs the function of a folder, enabling you to group a set of processes, tasks, repositories, and services, created to model your business operations.
- Resources
Performs the function of a folder, enabling you to group a set of resource definitions, resources, roles, and timetables, created to represent the resources of your organization.
- Organizations
Performs the function of a folder, enabling you to group a set of organization locations and structures. Our scenario keeps all organization information for the model in a single folder. More complex organizations might keep this in separate catalogs for different groups.

2. Add a new local task in the process.
 - a. Expand **ITSO Speedy Rentals Process Catalog**.
 - b. Double-click **Vehicle Fulfilment Process AsIs** to open the model in the process editor.
 - c. Double-click the **Vehicle Fulfilment Process AsIs** header to increase the process editor to full size, or click the **1-pane layout** button. Figure 11-10 shows the process editor.

Tip: Click **View** → **Toggle grid on editor** for accurate diagram layout.

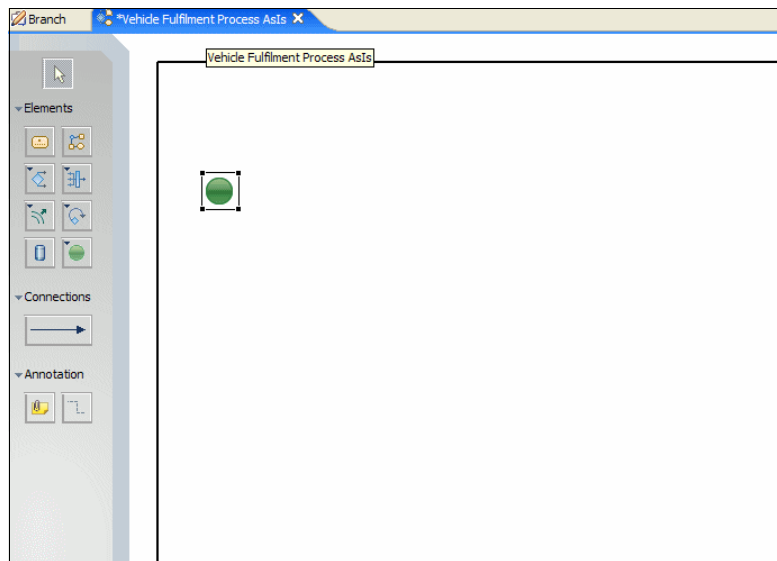



Figure 11-10 Process editor view

- d. Select the **create local task** button  on the editor palette to add a new task.

- e. In the process editor workspace, add the new task box by moving the cursor into the drawing area and clicking the left mouse button. Figure 11-11 shows the new added task.

Figure 11-11 Add a new task

- f. Rename the task to Reserve Vehicle System.
3. Add the activities as shown in Table 11-1 using the Element buttons on the process editor palette.

Element Name	Element Type
Reserve Vehicle System	Local task
Vehicle Available?	Simple decision
Vehicle Pickup	Local task

Element Name	Element Type
Assign Higher Group Vehicle	Local task
Release Vehicle In The Pool	Local task
Vehicle Return	Local task
Return Date Expired	Local task
Generate Invoice	Local task
Update Rental Agreement	Local task
Rental Agreement Fork	Fork

Note: To assign a name to a Fork element, select the Fork (found in the editor under the Join element), then modify the Name attribute in the General tab of the Attributes view. You must be in the 2-pane or 4-pane layout to see the Attributes view.

- g. Save the process by pressing Ctrl+S. The process should resemble Figure 11-12.

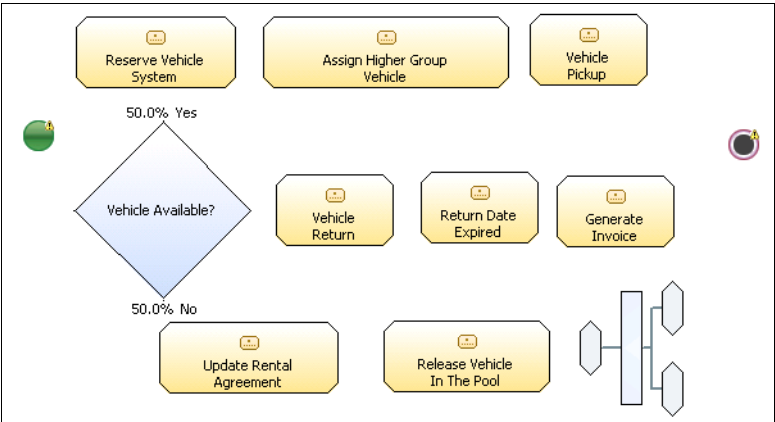


Figure 11-12 Elements within the process model

11.3.3 Creating the business items

With the basic model business activities have been created, associate the business data required for each task.

Tip: Business items may be defined prior to creating the process flow. Having the process flow defined might help with understanding the business process and make defining business items for that particular flow simpler.

For the ITSO Speedy Rentals scenario, we defined data required to complete the vehicle fulfilment process, keeping the model easy to understand. The model has been designed into four entities, as shown in the following four tables: Branch, Driver, Rental Agreement, and Vehicle.

Table 11-2 Branch details

Name	Type
Location ID	String
Location Type	String
Address	String
City	String
Country	String
ZIP Code	String

Table 11-3 Driver details

Name	Type
Driver Name	String
License ID	String
License Expiration Date	Date
Date of Birth	Date
Credit Card Number	String
Debit Card Number	String
Risk Level	String
Loyalty Club Number	String

Table 11-4 Rental agreement details

Name	Type
Reservation ID	String
Vehicle Group Requested	String
Vehicle Group Received	String
Pickup DateTime	DateTime
Planned Return DateTime	DateTime
Actual Return DateTime	DateTime
Estimated Cost	Double
Overdue Status	String
Discount	String
Rental Status	String
Actual Cost	Double
Driver Details	Driver
Branch Details	Branch
Vehicle Details	Vehicle
Vehicle Status	String

Table 11-5 Vehicle details

Name	Type
Group ID	String
Model Name	String
Manufacturer	String
Engine Size	String
Fuel Type	String
Transmission Type	String
Available	Boolean
Inventory ID	String

1. Create the four business items.
 - a. Click the **4-pane layout** button.
 - b. Expand **ITSO Speedy Rentals Data Catalog** in the project tree.
 - c. Right-click the **ITSO Speedy Rentals Business Item** and select **Rename**. This renames the business item created by default.
 - d. Enter the name **Driver**.
 - e. Right-click **ITSO Speedy Rentals Data Catalog** from the project tree view and select **New** → **Business Item**.
 - f. Enter **Branch** and click **Finish**.
 - g. Use **New** → **Business Item** to create business items for the Rental Agreement and Vehicle.

The business items are now created in the project tree view (Figure 11-13).

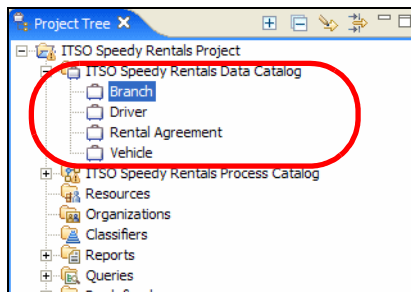


Figure 11-13 Business item project tree

2. Add attributes to the Branch business item.
 - a. Double-click **Branch** in the tree to open the business attributes editor.
 - b. To add an attribute, click **Add**. This adds a default attribute of type String (Figure 11-14).

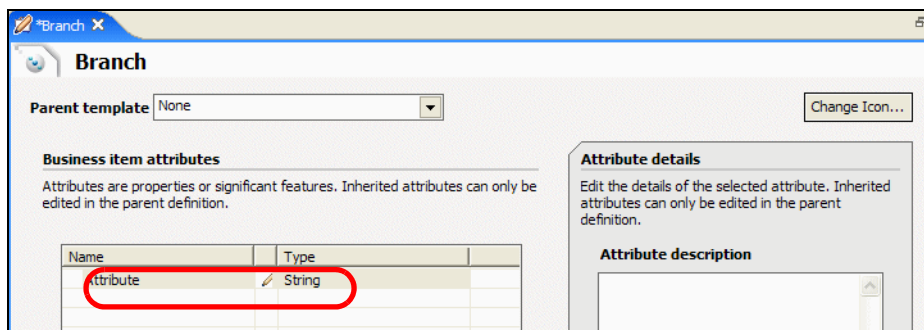


Figure 11-14 Add a new attribute

- c. Click **Attribute** and overtype this name with the new attribute name, which in this case is Location ID.
- d. To change the attribute type, highlight **String** then click ... to open the Type Selection dialog. From here you can select a basic or complex type. In this instance, we keep the attribute types as String.
- e. Add the remaining attributes for Branch, as described in Table 11-2 on page 181. Figure 11-15 shows the completed editor.
- f. Save the Branch business item and close the editor.

The screenshot shows the 'Branch' business item editor. At the top, there is a 'Parent template' dropdown set to 'None'. Below this is a section titled 'Business item attributes' with a note: 'Attributes are properties or significant features. Inherited attributes can only be edited in the parent definition.' A table lists the attributes:

Name	Type
Location ID	String
Location Type	String
Address	String
City	String
Country	String
ZIP Code	String

Figure 11-15 Completed business item Branch

3. Define attribute names and types for the Driver, Rental Agreement, and Vehicle business items, using the attributes shown in Table 11-3 on page 181 through Table 11-4 on page 182. Be sure to save your changes.

Note: Use the Type Selection dialog to change the attribute type from String where necessary. For basic types (such as Double and Boolean), select the attribute type from the Basic type list. For complex types (such as Driver, Branch, and Vehicle) select a complex type under the Complex type radio button.


11.3.4 Connecting tasks with associated data

Now that the business items have been created, document the model with more detail. For each task, define the input to that task (the business items that are required for that task to complete) and the business item outputs from the task.

Note: Our scenario uses the <task name> <attribute> naming convention (for example, Vehicle Pickup Input and Vehicle Pickup Output). When a task has multiple inputs we used <meaningful name> <attribute>.

1. Open the Process editor for Vehicle Fulfilment Process AsIs model.
 - a. Expand **ITSO Speedy Rentals Process Catalog**.
 - b. Double-click **Vehicle Fulfilment Process AsIs**.
2. Connect the elements in the process.

A connection enables you to define the relationship between activities and shows the execution flow of the business process. A connection carries associated data required as input or output from or to an activity.

- a. Click the Connector button .
- b. Make a connection from the external boundary to Reserve Vehicle System task using the left mouse button to select the “from” and “to” tasks as shown in Figure 11-16.

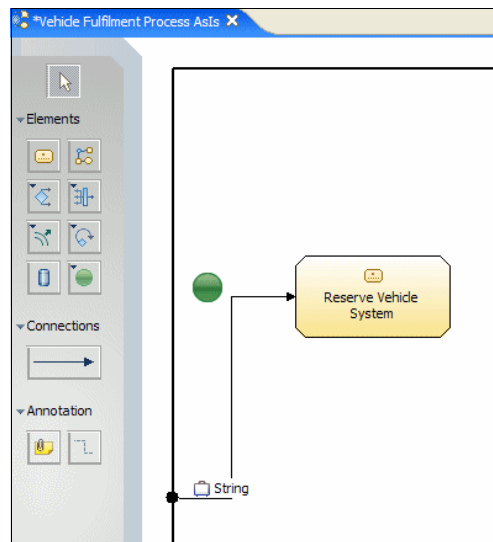


Figure 11-16 Connecting tasks

- c. Select the regular mouse cursor, then click on the connection. The Attributes view displays the connection attributes (Figure 11-17). Set the name of the connection to **Vehicle Fulfilment Start Connection** and use the **Browse** button to set the Associated data to **Rental Agreement**.

General

▼ General information

This section provides general information about this object flow.

Name

Vehicle Fulfilment Start Connection

Description

Associated data

Rental Agreement

Figure 11-17 Connection attributes

- d. Create all of the connections shown in Table 11-6, making sure to provide the correct names and associated data. In doing so, note that:
- When you create connections between the fork element and other tasks, the Data Connection dialog asks whether you would like to create a new output or use the Rental Agreement (Figure 11-18). Select the default of using **Rental Agreement** and click **OK**.

Data Connection

Select an existing input and output for the connection, or choose to create new ones.

Available outputs of Rental Agreement Fork

☐ Create a new output

☒ Output, Rental Agreement

Available inputs of Vehicle Return

☒ Create a new input

OK Cancel

Figure 11-18 Data Connection dialog

- The *Start* element is the green circle at the start of the process and the *Stop* element is the red and gray circle at the end of the process.

Table 11-6 Element connections

“From” element	“To” element	Connection name	Associated data
External Boundary	Reserve Vehicle System	Vehicle Fulfilment Start Connection	Rental Agreement
Start	Reserve Vehicle System	Start Connection	None

“From” element	“To” element	Connection name	Associated data
Reserve Vehicle System	Vehicle Available	Vehicle Available Connections	Rental Agreement
Vehicle Available (yes)	Vehicle Pickup	Vehicle Is Available Connection	Rental Agreement
Vehicle Available (no)	Assign Higher Vehicle Group	Vehicle Not Available Connection	Rental Agreement
Assign Higher Vehicle Group	Reserve Vehicle System	Assign Higher Vehicle Group Connection	Rental Agreement
Vehicle Pickup	Release Vehicle in The Pool	Release Vehicle Connection	Rental Agreement
Release Vehicle In The Pool	Fork	Vehicle Status Connection	Rental Agreement
Fork(1)	Vehicle Return	Vehicle Return Connection	Rental Agreement
Fork(2)	Return Date Expired	Vehicle Not Returned Connection	Rental Agreement
Vehicle Return	Generate Invoice	Generate Invoice Connection	Rental Agreement
Return Date Expired	Update Rental Agreement	Rental Agreement Update Connection	Rental Agreement
Update Rental Agreement	Return Date Expired	Check Vehicle Return Connection	Rental Agreement
Generate Invoice	Stop	End Connection	Rental Agreement

Note: For clarity, our scenario has maintained the attribute type. For example, we name connections as <taskname> Connection.

3. The process model should now look like Figure 11-19 on page 188. Save the process.

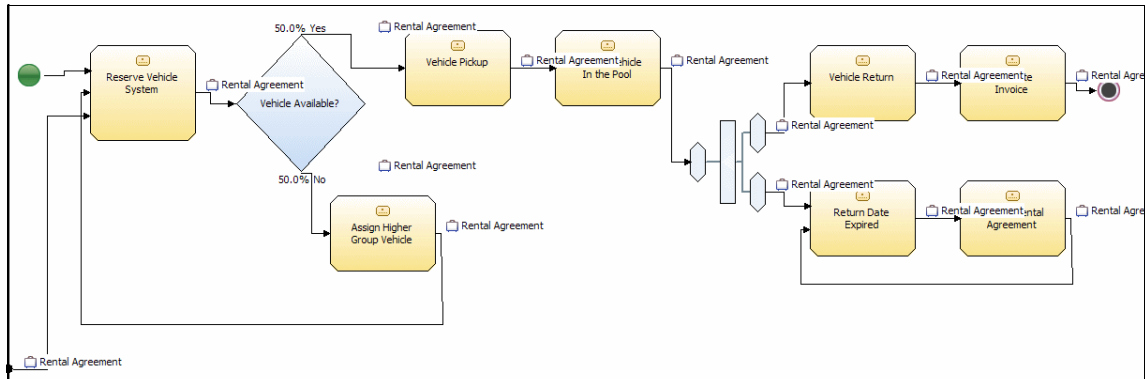


Figure 11-19 As-Is model with flow connections and associated data

4. Define the input and output for the complete process model.

Our process requires a completed Rental Agreement from a Reserve Vehicle process, so add this business item as an input into the Reserve Vehicle System task, and also into the output.

- Click in any white space in the process editor to display the process properties in the Attributes view.
 - In the Attributes view, click the **Inputs** tab. Rename the existing input attribute from Input to Rental Agreement Input.
 - Ensure that the Associated data field for the input is set to **Rental Agreement**.
 - Click the **Outputs** tab. Add a new output by clicking **Add**. Name the output to Rental Agreement Output and set the Associated data to **Rental Agreement**.
5. Rename the inputs and outputs for each of the elements in the process. To perform the renaming, highlight an element, click either **Inputs** or **Outputs** in the Attributes view, and change the name. Use the values in Table 11-7 on page 189 for the input and output names.

Tip: Some elements contain multiple inputs or outputs. You can determine which input or output relates to which connection based on how they appear connected to the task. The top connection into a task is the first input listed for the task. The connection below that is the second input (Input:2) listed for the task, and so forth.

Table 11-7 Element Input and Output attributes

Element	Input	Output
Reserve Vehicle System	Start Process Input Rental Agreement Input Vehicle Upgrade Input	Reserve Vehicle System Output
Vehicle Available	Vehicle Available Input	Vehicle Is Available Output Vehicle Is Not Available Output
Vehicle Pickup	Vehicle Pickup Input	Vehicle Pickup Output
Release Vehicle In the Pool	Release Vehicle Input	Release Vehicle Output
Assign Higher Group Vehicle	Assign Higher Group Vehicle Input	Assign Higher Group Vehicle Output
Vehicle Return	Vehicle Return Input	Vehicle Return Output
Fork	Planned Return Date Input	Actual Return Date Output New Return Date Output
Generate Invoice	Generate Invoice Input	Generate Invoice Output
Return Date Expired	Return Date Expired Input New Return Date And Time Input	Return Date Expired Output
Update Rental Agreement	Update Rental Agreement Input	Update Rental Agreement Output

Note: For clarity, this scenario example uses the <task name> <attribute> naming convention (Vehicle Pickup Input, Vehicle Pickup Output). When a task has multiple inputs, we have used <meaningful name> <attribute>.

6. (Optional) Enter a meaningful description for each task in the Description field of the element Attributes view.
7. Save the process using Ctrl+S.

11.3.5 Adding classifiers

Classifiers are used to categorize tasks and other process elements for analysis purposes. Several predefined classifiers are available. Our scenario defines two classifiers, IT System Classifier and Manual Classifier.

To define and add a classifier:

1. Create a new classifier catalog.
 - a. Select the 4-pane layout.

- b. Right-click **Classifiers** in the project tree and select **New → Classifier Catalog**.
 - c. Enter the catalog name **ITSO Speedy Rentals Classifier Catalog** and click **Finish**.
 2. Create a new classifier.
 - a. Right-click **ITSO Speedy Rentals Classifier Catalog → New → Classifier**.
 - b. Enter **Implementation Classifier** as the name and select **Finish**.
 3. Add a new classifier value.
 - a. Right-click **ITSO Speedy Rentals Classifier Catalog → New → Classifier Value**.
 - b. Enter **IT System** for the name, choose a color, and select **Finish**.
 4. Create a second classifier value named **Manual Activity**.
 5. After the classifier values are defined, they can be added to the model's activities using the **Classifiers** tab in the **Attributes** view.
 - a. In the **Vehicle Fulfilment Process AsIs** process model, click **Reserve Vehicle System** to view its attributes in the **Attributes** view.
 - b. Click the **Classifier** tab in the **Attributes** editor.
 - c. Click **Add** and select **ITSO Speedy Rentals Classifier Catalog → Implementation Classifier → IT System** to mark this as an automated activity. Click **OK**. Figure 11-20 on page 191 shows the classifier attributes for **Reserve Vehicle System**.

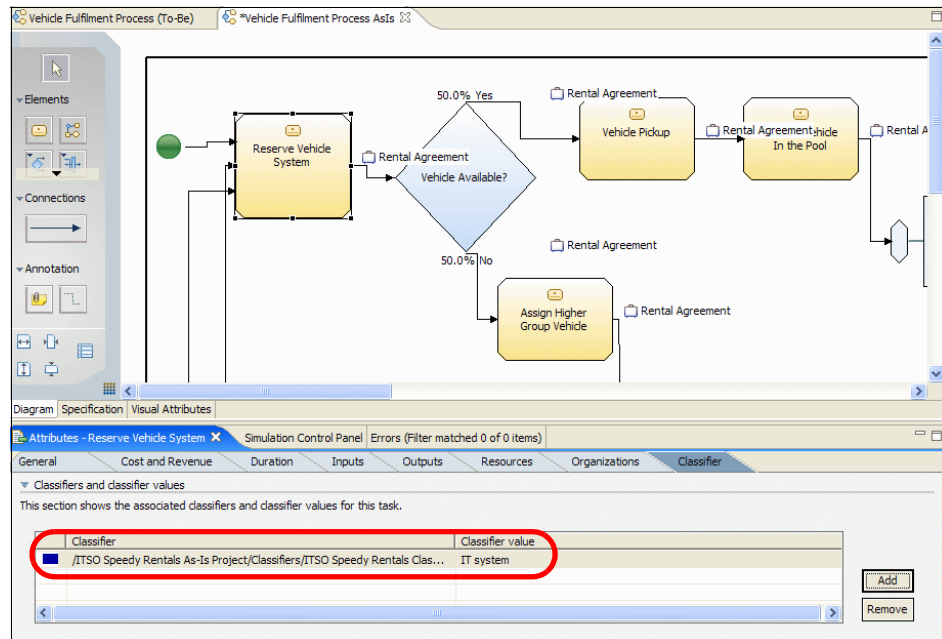


Figure 11-20 Classifier Attribute Editor

6. Add a new classifier for each activity using the values in Table 11-8.

Table 11-8 Classifier attributes

Element	Classifier
Reserve Vehicle System	IT System
Vehicle Pickup	Manual Activity
Assign Higher group Vehicle	Manual Activity
Release Vehicle In The Pool	IT System
Vehicle Return	Manual Activity
Return Date Expired	IT System
Update Rental Agreement	Manual Activity
Generate Invoice	IT System

11.3.6 Adding resources and roles

A resource catalog is created to hold resource definition templates, resource definitions, resources, and roles. This scenario has the following roles:

- ▶ Head Office Manager
- ▶ Head Office Agent
- ▶ Branch Manager
- ▶ Branch Agent
- ▶ Branch Vehicle Agent

The scenario resource is the Rental IT System.

1. Create a new resource catalog.
 - a. In the project tree, right-click **Resources** and select **New** → **Resource Catalog**.
 - b. In the Name of new resource catalog field enter your catalog name (ITSO Speedy Rentals Resource Catalog).
 - c. Click **Finish**.
2. Create a new resource definition.

A resource definition is a set of similar characteristics shared by resources of the same type. For example, we have defined a Rental IT System resource.

- a. In the project tree, right-click **ITSO Speedy Rentals Resource Catalog** and select **New** → **Resource Definition**.
 - b. Enter Rental IT System Resource Definition.
 - c. Click **Finish**.
3. Create a new resource.
 - a. In the project tree, right-click **ITSO Speedy Rentals Resource Catalog** and select **New** → **Resource**.
 - b. In the name field enter Rental IT System Resource and ensure that the Rental IT System Resource Definition is selected as the Associated resource definition.
 - c. Click **Finish**.
4. Create the new roles.

We created several roles required for a task in our ITSO Speedy Rentals Resource Catalog.

- a. In the project tree, expand **Resources**, right-click **ITSO Speedy Rentals Resource Catalog**, and select **New** → **Role**.
 - b. Enter Head Office Manager Role in the Name of new role field.

- c. Click **Finish**.
- d. Repeat these steps to create the following role names:
 - Head Office Agent Role
 - Branch Manager Role
 - Branch Agent Role
 - Branch Vehicle Agent Role

Your project tree should now have all the resources and roles defined as shown in Figure 11-21.

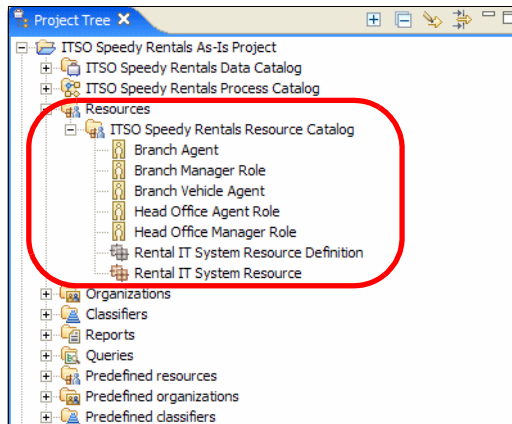


Figure 11-21 Resources project tree

11.3.7 Defining the organization

To complete the As-Is model, define the ITSO Speedy Rentals organization. This will be held in a catalog and can be used to model organization entities, locations, and structures. For the purpose of this book, we define a simplified organizational structure.

1. Create the organization catalog.
 - a. In the project tree, right-click **Organizations** and select **New** → **Organization Catalog**.
 - b. In the Name of new organization catalog field, enter the catalog name, ITSO Speedy Rentals Organization Catalog.
 - c. Click **Finish**.
2. Create an organization definition.
 - a. In the project tree, expand **Organizations**, right-click **ITSO Speedy Rentals Organization Catalog**, and select **New** → **Organization Definition**.

- b. Enter Branch Organization Definition in the name field.
 - c. Click **Finish**.
- 3. Create an organization unit.
 - a. In the project tree, expand **Organizations**, right-click **ITSO Speedy Rentals Organization Catalog**, and select **New** → **Organization Unit**.
 - b. Enter a name of Head Office Organizational Unit and ensure that the Associated organization definition is set to **Organization definition**.
 - c. Click **Finish**.
- 4. Create a location definition.
 - a. In the project tree, expand **Organizations**, right-click **ITSO Speedy Rentals Organization Catalog**, and select **New** → **Location Definition**.
 - b. Enter Branch Location Definition for the name.
 - c. Click **Finish**.
 - d. In the Branch Location Definition editor, click **Add** to add a new attribute.
 - e. Enter Branch Details for the name and change the type to Branch, from our business item catalog.
 - f. Save the changes in the editor.
- 5. Create a location.
 - a. In the project tree, expand **Organizations**, right-click **ITSO Speedy Rentals Organization Catalog**, and select **New** → **Location**.
 - b. Enter Branch Location for the name, and ensure that the Associated location definition is set to **Branch Location Definition**.
 - c. Click **Finish**.
- 6. Create an organization structure.
 - a. In the project tree, expand **Organizations**, right-click **ITSO Speedy Rentals Organization Catalog**, and click **New** → **Structure**.
 - b. Enter ITSO Speedy Rentals Organization Structure for the name.
 - c. Click **Finish**.
 - d. The ITSO Speedy Rentals Organization Structure editor opens. In the palette, click the **Create organization unit** button and place it on the canvas. Select **Head Office Organization Unit**.
 - e. Create a new location: Click the **Location** button in the palette and place it on the canvas. Select **Branch Location**.

- f. Use the **Connector** button to connect Head Office and Branch.
Figure 11-22 shows our first connection.

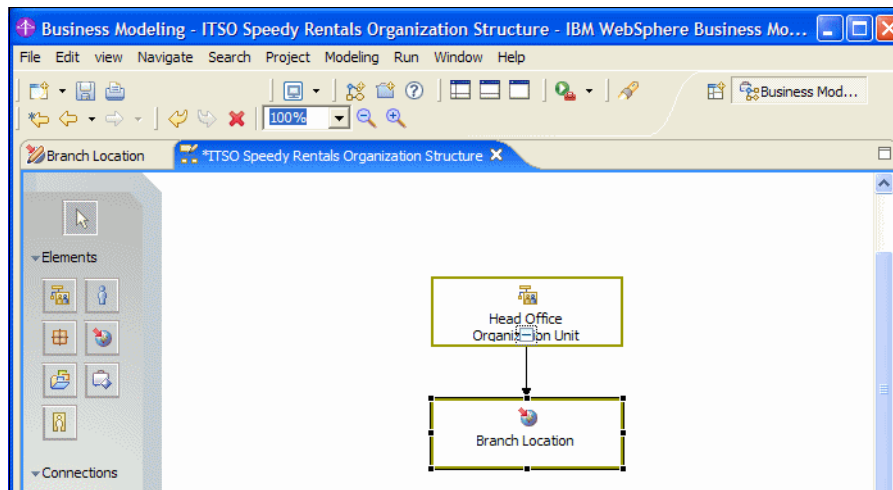


Figure 11-22 Adding a new organization structure

- g. Add roles to the structure using the **Roles** button on the palette, using the role definitions from the supplied list (Figure 11-23). Define these roles:
- Head Office Manager Role
 - Head Office Agent Role
 - Branch Manager Role
 - Branch Agent Role
 - Branch Vehicle Agent Role

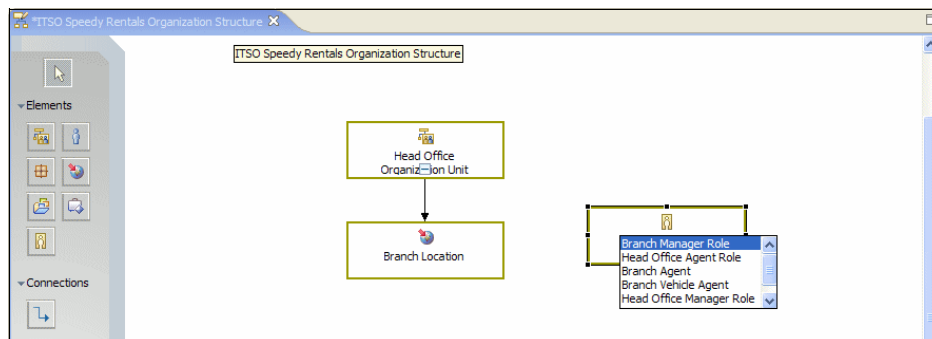


Figure 11-23 Adding roles to the organization structure

- h. Create a new resource by selecting the **Create bulk resource** button from the palette and placing it on the canvas. Select **Rental IT System Resource**.

- i. Connect your resources using connections, as shown in Figure 11-24.

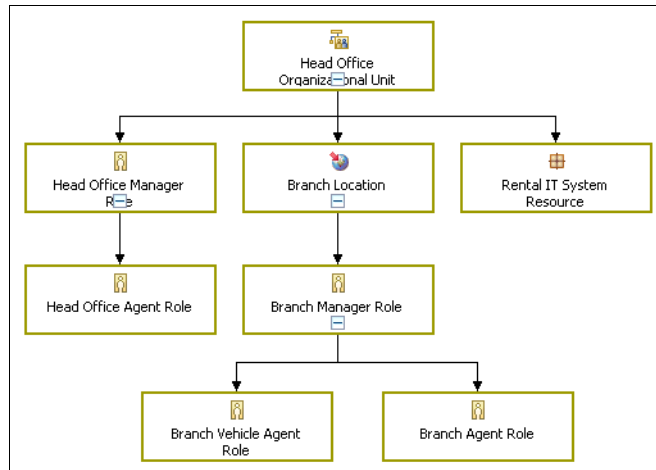


Figure 11-24 Complete organization structure

- j. Save your organization structure.

11.3.8 Adding resources and organizations to the process

For a complete model, add the resources to the As-Is model. You can assign roles to local tasks by selecting a role in the project tree and dragging it to the task in the process diagram. In addition, assign an estimated time to each task as to how long the resource is needed. This represents the task duration actively executed, rather than the time an activity might wait for an unavailable resource.

1. Assign a resource to the Reserve Vehicle System task.
 - a. Open the Vehicle Fulfilment Process AsIs process in the editor.
 - a. In the project tree, expand **Resources** → **ITSO Speedy Rentals Resource Catalog**, click **Rental IT System Resource** and drag it to the **Reserve Vehicle System** task in the process diagram. Drop it there. This creates an entry in the resource attributes.
 - b. Click the **Resources** tab in the Attributes view.
 - c. In the Bulk resource requirements section, click the first entry, **Bulk requirement 1**. Change the name to Rental IT System Bulk Requirement.
 - d. Click **0 second** in the Time required column. Click the ... icon to open the Time duration window.
 - e. Enter 2 Milliseconds as an estimated duration of the resource required by this task. (We have assigned a pseudo time of 2 milliseconds to all

automatic tasks.) In most cases, execution times of automatic tasks are inconsequential compared to the potentially long durations of manual tasks.

- f. Click **OK**.
2. Repeat these steps for the other automated IT tasks using the values in Table 11-9. For each task, drag Rental IT System Resource onto the task to create a Bulk resource requirements entry.

Table 11-9 Adding bulk resources to the model

Task in the diagram	Name	Time required
Release Vehicle in The Pool	Release Vehicle In The Pool Bulk Resource	2 milliseconds
Return Date Expired	Return Date Expired Bulk Resource	2 milliseconds
Generate Invoice	Generate Invoice Bulk Resource	2 milliseconds

3. Assign a role to the Assign Higher Group Vehicle manual task.
 - a. Under Resources in the project tree, click **Branch Agent Role** and drag it to the **Assign Higher Group Vehicle** task in the process diagram. Drop it there. This creates an entry in the resource attributes.
 - b. Click the **Resources** tab in the Attributes view.
 - c. In the Role requirements section, click the first entry, **Role requirement 1**, and change the name to Assign Higher Group Vehicle Role Requirement.
 - d. Click **0 second** in the Time required column. Click the ... icon to open the Time duration window.
 - e. Select **10 Minutes** as estimated duration it takes the branch agent to fulfill the task. (We assume that the current process is not very effective, therefore the agent needs a relatively long period of time to find and assign a car of a higher car group.)
 - f. Click **OK**.
 - g. Click the entry under Resource definition. Click the ... icon to open the Select Resource Definition window.
 - h. Click **Staff** to select it.
 - i. Click **OK**.
4. Repeat these steps for the other manual tasks, using the values in Table 11-10. Be sure to set the Resource definition to Staff for each task.

Table 11-10 Adding roles into the model

Role to drag	Task in the process	Name	Time required
Branch Vehicle Agent Role	Vehicle Pickup	Vehicle Pickup Role Requirement	10 Minutes
Branch Vehicle Agent Role	Vehicle Return	Vehicle Return Role Requirement	12 Minutes
Branch Agent Role	Update Rental Agreement	Update Rental Agreement Role Requirement	1 Hour

5. Save the changes to the process using Ctrl+S.

11.4 Simulating the As-Is current process model

With the main steps completed in creating the process model, use WebSphere Business Modeler to simulate running the process. The simulation feature enables you to check that your model is correct and flows as expected, gather performance statistic about the process, and highlight any potential areas for improvement, as well as bottlenecks.

Simulation mainly depends on, and takes into account, the following process definitions:

- ▶ Resources and roles associated with availability and costs
- ▶ Task durations specified
- ▶ Probabilities on output branches of decisions

11.4.1 Configuring simulation

This section shows how to set up and run a basic simulation; however, in this book, we do not dive deeply into process analysis and redesign.

Before you can start simulation, configure the As-Is model to add additional information.

1. Add probabilities to decision branches:
 - a. Click the **Vehicle Available?** decision in the process diagram view.
 - b. Click **Output branches** in the Attributes view.
 - c. Click the first **50%** entry in the **Probability (%)** column.
 - d. Type 85 as probability for the Yes branch. This means that the requested car group is expected to be available in 85% of the cases.

- e. Click the second **50%** entry in the **Probability (%)** column.
 - f. Type **15** as probability for the No branch. This means we expect a higher car group is assigned in 15% of the cases.
 - g. Press **Ctrl+S** to save the changes.
2. Complete the model and allow loopback in the simulation runs.
 To run a simulation, you must examine the input criteria for each task and ensure that the criteria are met. Especially consider loopbacks, where a connection returns back to a previous task in the flow (such as the connection between **Update Rental Agreement** and **Return Date Expired**). If a task has multiple inputs, the default behavior is that both inputs have to arrive before the task can start. In the case of a loopback, this means that the task never will be executed, so modify that default behavior to start the task on arrival of either input.
- a. Click **Modeling** → **Mode** → **Intermediate** to switch modeling mode to the intermediate mode. Input criteria and output criteria can be changed only in intermediate and advanced mode.
 - b. Click the **Reserve Vehicle System** task in the process editor view.
 - c. Click the **Input Logic** tab in the Attributes view.
 - d. Click **Add** to generate a second input criteria entry in the table. The second entry is generated as an OR criterion to the first entry.
 - e. Unmark the check box under **Vehicle Upgrade Input** in the first row of the table. This change defines that the Vehicle Upgrade Input is not required for the first part of the criteria.
 - f. Mark the check box under **Vehicle Upgrade Input** in the second row of the table. See Figure 11-25.

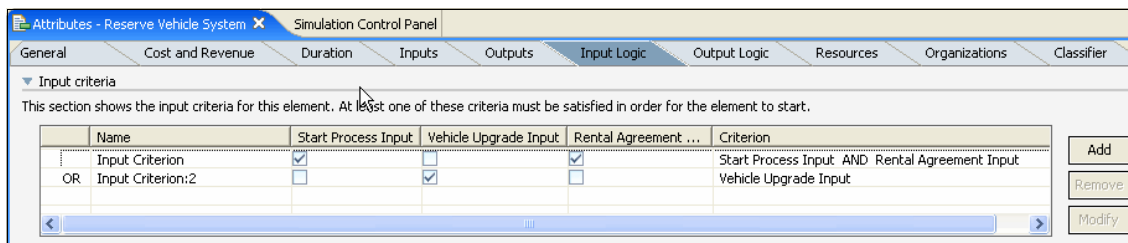


Figure 11-25 Multiple input criteria

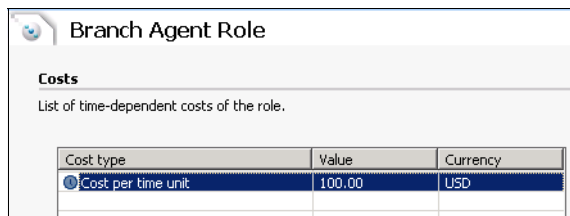
- g. Click the task **Return Date Expired** in the Process Editor view, and apply the same change as above to the second loop in the process.
- h. Click the **Input Logic** tab in the Attributes view.
- i. Click the **Add** icon to generate a second input criteria entry in the table.

- j. Unmark the check box under **Return Date Expired Input** in the first table entry.
 - k. Mark the check box under **Return Date Expired Input** in the second table entry. This changes the input criteria for the Return Date expired task so that a loopback from the Update Rental Agreement task will start the task again.
 - l. Press Ctrl+S to save the changes.
 - m. Click **Modeling** → **Mode** → **Basic** to switch from Modeling mode back to the Basic mode.
3. Add a decision to simulate the probability of the Update Rental Agreement task.

The Return Date Expired task has the function of a timer that expires when the car is not returned in time. To simulate the expiration probability and its impact on the overall process, add a decision element to the path to specify the probability that a customer does not return the car and the branch agent has to call them to update the rental agreement.

- a. Right-click the connector between the output of the Return Date Expired task and the input of the Update Rental Agreement task. Click Delete to remove the connector.
- a. Right-click the connector between the output of the Generate Invoice task and the Stop node. Click Delete to remove the connector.
- b. Select the **Simple decision** element from the palette and drag it to the process graph area of the connector that was just removed. Drop it and rename it to **Expired Decision**.
- c. Select the **Join** icon from the palette and drag it to the process graph.
- d. Select the connector from the palette and connect the output of the Return Date Expired task with the entry of the Expired Decision. Accept the defaults in the Data Connection dialog.
- e. Connect the Yes output of the Expired Decision with the input of the Update Rental Agreement tasks. Accept the defaults in the Data Connection dialog.
- f. Connect the No output of the Expired Decision with one of the inputs of the Join element. Accept the defaults in the Data Connection dialog.
- g. Connect the Generate Invoice task with one of the inputs of the Join element. Accept the defaults in the Data Connection dialog.
- h. Connect the output of the Join element with the Stop node. In the Data Connection dialog, select **Output, Rental Agreement**.
- i. Click the **Expired Decision** element in the process editor.
- j. Click the **Output branches** tab in the Attributes view.

- k. Click the first entry, **50.0**, in the probability column (Yes path) and type 20.
 - l. Click the second entry, **50.0**, in the probability column (No path) and type 80. This means we expect in 80% of the cases the customers will bring back the rented car in time before the contract expires.
 - m. Press Ctrl+S to save the changes.
4. Assign some costs to roles for simulation.
 - a. In the project tree, expand **Resources** → **ITSO Speedy Rentals Resource Catalog**, and double-click **Branch Agent Role** to open the Branch Agent Role window.
 - b. Click the **Costs** tab.
 - c. Click **Add** to add a new cost.
 - d. Click the **Cost per time unit** radio button and click **OK**.
 - e. Click the first **0.00** entry in the Value column, and type 100 as the value (Figure 11-26). This means the calculation that it will cost \$100 for every hour the role is carried out.



Branch Agent Role		
Costs		
List of time-dependent costs of the role.		
Cost type	Value	Currency
<input checked="" type="radio"/> Cost per time unit	100.00	USD

Figure 11-26 Branch Agent Role

- f. Save and close the Branch Agent Role window.
 - g. Double-click **Branch Vehicle Agent Role** to open the Branch Vehicle Agent Role window.
 - h. Click the **Costs** tab, click **Add**, select the **Cost per time unit** radio button, and click **OK**.
 - i. Click the first **0.00** entry in the Value column and type 80 as value to signify \$80 for every hour the role is carried out.
 - j. Save and close the Branch Agent Role window.
5. Assign some costs and revenue to tasks for simulation.
 - a. Click the **Vehicle Pickup** task in the process editor.
 - b. Click the **Cost and Revenue** tab in the Attributes view.
 - c. Change the first pull-down box from None to **Literal Value**.

- d. Enter 120 as the cost for every time the tasks runs (Figure 11-27) for the calculation that on average the vehicle cost is \$120. We assume that vehicle costs are due when the customer takes the car. No tasks in the process other than the Update Rental Agreement task will carry vehicle cost.

The screenshot shows a software window titled 'Attributes - Vehicle Pickup'. It has several tabs: 'General', 'Cost and Revenue' (which is selected), 'Duration', 'Inputs', 'Outputs', 'Resources', and 'Organizations'. Under the 'Cost and Revenue' tab, there is a section for 'Processing cost' with the description 'The expense incurred every time this task runs.' Below this, there is a dropdown menu set to 'Literal value', a text input field containing the number '120', a currency dropdown menu set to 'USD', and a 'Clear' button.

Figure 11-27 Processing cost

- e. Scroll down to the Revenue section in the Attributes view.
- f. Specify the money that is earned when the task completes. By default it is set to None, which indicates that no revenue is made. To change this, select **Literal** value and specify the amount of revenue made. For this task, we leave it set to None.
- g. Define the costs and revenue for the tasks shown in Table 11-11.

Table 11-11 Costs and revenues

Task	Costs for the task	Revenue for the task	Comments
Reserve Vehicle System	\$1	None	A system task; therefore has a relatively low cost.
Vehicle Return	None	\$200	A manual task with no additional cost. Average revenue of the rental contract is accounted for here.
Assign Higher Group Vehicle	\$40	None	A manual task. An upgrade adds additional vehicle costs.
Release Vehicle in the Pool	\$1	None	A system task; therefore has a relatively low cost.
Update Rental Agreement	\$30	\$50	A manual task. Contract extension adds additional vehicle cost and generates additional revenue.
Generate Invoice	\$1	None	A system task, and therefore has a relatively low cost.

- h. Save the process model.

11.4.2 Generating a simulation snapshot and running the simulation

You must create a simulation snapshot before you can run a simulation. A simulation snapshot captures the process as it is at a particular time, including all of the current default values and input data for the included tasks and activities. You can create and keep several simulation snapshots of a process. This enables you to compare and analyze simulation results of model and input variations.

1. Generate the simulation snapshot.
 - a. Right-click **Vehicle Fulfilment Process AsIs** in the project tree and click **Simulate** to generate a simulation snapshot profile and open the simulation view.
 - b. Click the **Inputs** tab in the Attributes view.
 - c. Click **Rental Agreement Input** to highlight the first entry in the table. Click **Edit** in the **Total number of tokens** section to open the Edit simulation attribute dialog.
 - d. Enter 20 in the **Edit simulation attribute** window under Literal value. This means that you will generate and run 20 cases through the process model.
 - e. Click **OK**.
 - f. Click **Edit** in the **Recurring time interval for bundle creation** section to open the Selection duration dialog.
 - g. Change the entry from 1 minute to 10 minutes. This means that you expect a customer to collect a rented car every 10 minutes on average.
 - h. Click **OK**.
 - i. Press Ctrl+S to save the changes.
2. Run the simulation.
 - a. Click the **Simulation Control Panel** tab to open the simulation control icons.

- b. Click the green start icon (marked in Figure 11-28) to run the simulation. Note that 20 cases run graphically through the Vehicle Fulfilment Process. Each case generates an entry in the Simulation Control Panel below the graph. You can examine more details about each run by clicking the Processes tab, the Tasks tab, or the Connections tab. Each entry has details about process passes taken and tasks executed.

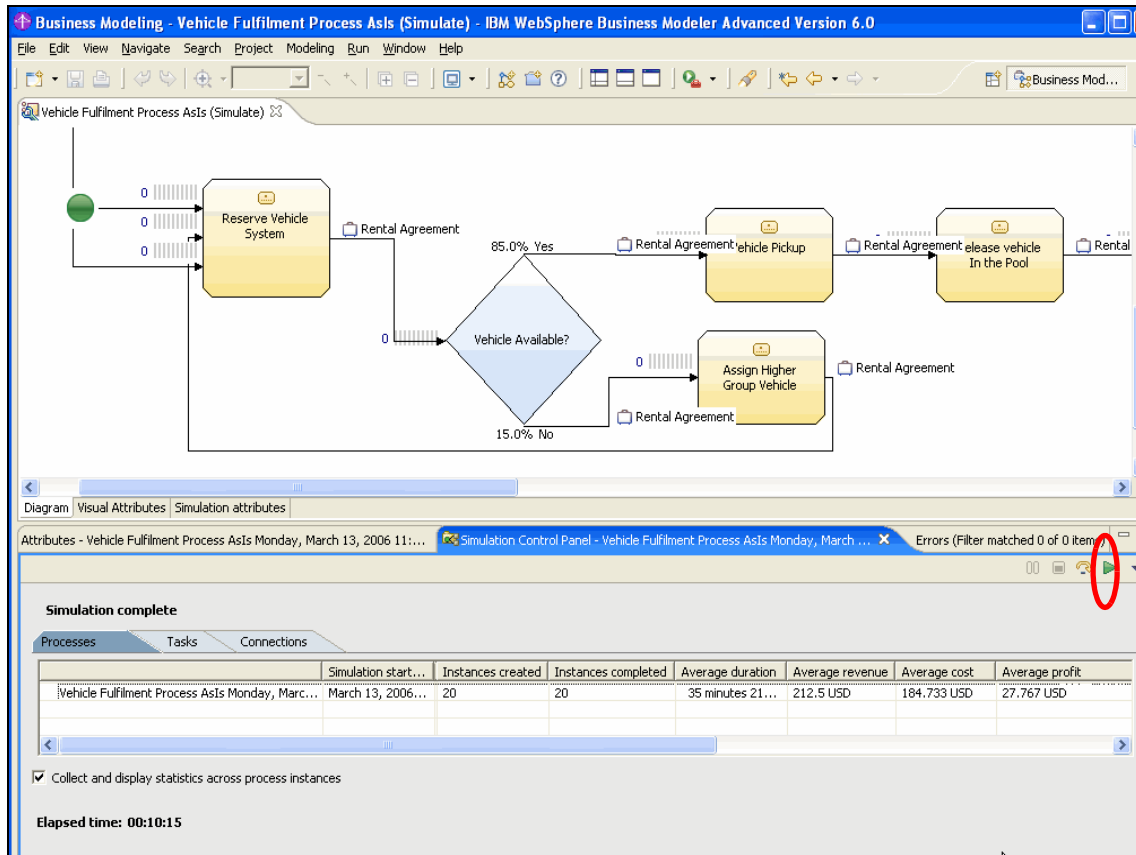


Figure 11-28 Run a simulation

- c. The Simulation Control Panel table (Figure 11-29) shows that some cases generate more costs than revenue; in these cases, the profit shows a negative value. These are typical cases that should be analyzed more deeply and better understood by business analysis. To examine more details, click the **Tasks** tab and open each case to see the tasks that have been executed.

Attributes - Vehicle Fulfilment Process AsIs Monday, March 13, 2006 11:...

Simulation Control Panel - Vehicle Fulfilment Process AsIs Monday, March ...

Err

Simulation complete

Processes

Tasks

Connections

	Process s...	Process end time	Total revenue	Total cost
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 13, 2006 11:09:04 PM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 13, 2006 11:24:04 PM CET	200 USD	201.667 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 13, 2006 11:29:04 PM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 13, 2006 11:39:04 PM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 14, 2006 12:37:04 AM CET	250 USD	252.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 14, 2006 12:47:04 AM CET	250 USD	252.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 14, 2006 12:09:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 13,...	March 14, 2006 12:29:04 AM CET	200 USD	251 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 12:29:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 12:39:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 1:37:04 AM CET	250 USD	252.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 12:59:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 1:09:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 1:19:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 1:29:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 1:39:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 3:37:04 AM CET	300 USD	352.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 1:59:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 2:09:04 AM CET	200 USD	152.333 USD
Vehicle Fulfilment Process AsIs Monday, March 13, 2006...	March 14,...	March 14, 2006 2:19:04 AM CET	200 USD	152.333 USD

☐ Collect and display statistics across process instances

Figure 11-29 Simulation Control Panel results

Note: For precise analysis we recommend increasing the number of tokens that are run through the process in simulation. However, high numbers of tokens increase the simulation time significantly.

Figure 11-29 might not exactly match your simulation results.

3. Analyze your simulation results with the dynamic analysis features.
- You can perform detailed analysis of the results of your simulations to extract information regarding scheduling, costs, output, and other statistics. You can perform the following four types of dynamic analysis on a process model:
- Aggregated**

Shows information about activities and resources used in all process instances generated during a simulation.

Process Cases	Shows statistics produced by all process cases in a simulation.
Process Instances	Provides a summary analysis to show process results for activities in a particular instance of a process that is created during a simulation run.
Process Comparison	Enables comparison of the weighted average analysis results for two simulated processes that use the same input parameters.

This scenario runs analysis of the process cases only on cost. This type of analysis categorizes the different passes that have been executed with the associated cost structure.

- a. In the project tree, expand **Vehicle Fulfilment Process AsIs**. Expand the **Vehicle Fulfilment Process AsIs Simulation snapshot** folder and the **Vehicle Fulfilment Process AsIs <day><month><year><time>** folder. This folder contains the result of your simulation with the date and timestamp showing when the simulation was run.
- b. Right-click **Simulation result <day><month><year><time>** and click **Dynamic Analysis → Process Cases Analysis → Process Cost**.
- c. The Process Cost Analysis dialog opens. Accept the defaults of all process instances and click **Finish**.
- d. The Path Signature dialog opens. Click **Yes** to generate path signatures.

- e. The Dynamic Analysis view is displayed as shown in Figure 11-30. When you select a case in the Dynamic View, the process editor is updated to illustrate that view.

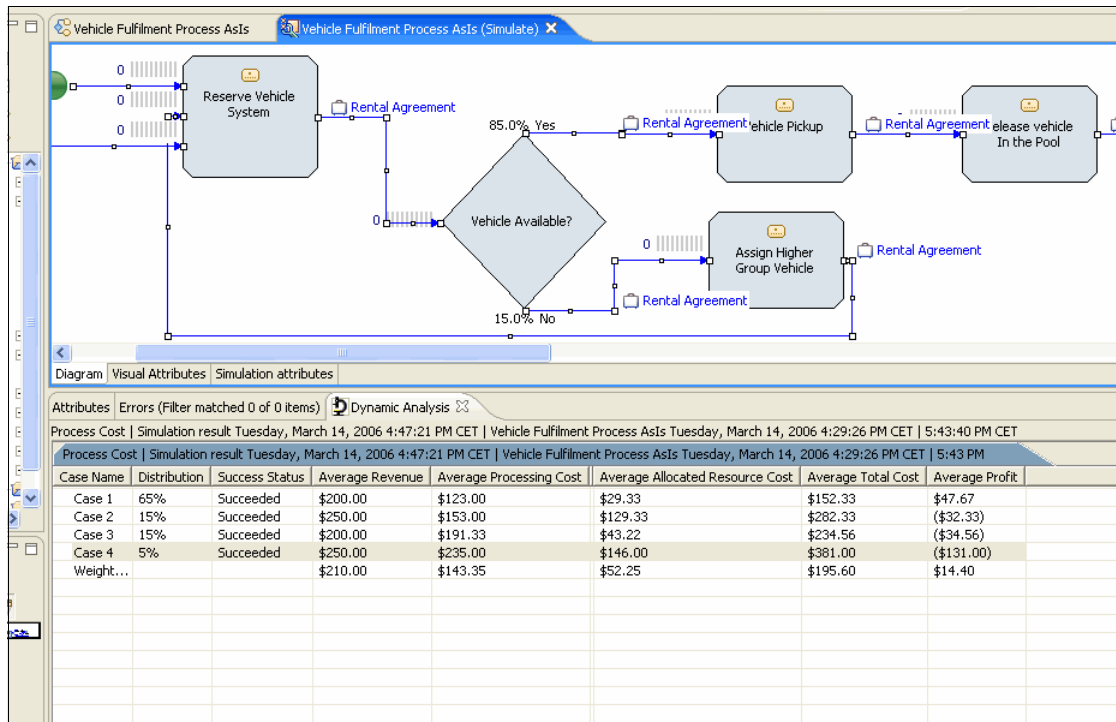


Figure 11-30 Case analysis view

Note: Your results in the Dynamic Analysis view probably will not match those in Figure 11-30, as we only ran a small simulation sample. It is possible, for example, that you will not see four cases, as shown in Figure 11-30.

- f. Click the **Case 1** entry in the table. The connectors in the process editor view turn blue and highlight the executed passes and task for this category. Specifically in our simulation results we see the following four cases. Your use cases might be different.
- **Case 1** represents the standard path where the requested car group is available and the customer returns the car according to the contract in time. On average this case generates a profit of around \$47. Unfortunately, the distribution shown in the table is just 65% of all cases. We had assumed that this probability would be higher.

- **Case 2** represents the path where customer got the requested car group but he did not return it according to the contract end date. The branch agent had to investigate and call the customer to extend the contract. Because the average allocated resource costs (branch agent) were raised by \$100, the overall profit is negative (about a \$35 loss). The distribution probability is 15%.
- **Case 3** represents the path where the customer got an upgraded car group because the requested group was not available. The upgrade means higher vehicle costs and higher average allocated resource costs (the use of a branch agent); therefore, the overall profit is negative (a loss of about \$35). The distribution probability is 15%.
- **Case 4** represents the path where customer was upgraded to a higher car group and did not return the car according to the contract end date. This is the combination of case 2 and 3. This on average results in a loss of \$131. The distribution is 5%.

Because we model the car group upgrade and the rental agreement extension tasks in a loop, each situation can happen multiple times in a rental agreement.

11.4.3 Analysis summary and recommendations for the To-Be model

Based on the results of the simulation, we can analyze the data as follows and recommend the following changes:

- ▶ Only 65% of all rental cases generate profit. In 35% percent of the rentals, the costs for the vehicles and the human resources are higher than the revenue generated so the company loses money. Our recommendation is to increase the number of default cases.
- ▶ The car group stock has to be adjusted properly for each location to reduce the number of upgrades. To achieve this, our recommendation is to monitor this closely and adapt quickly to the customer car group rental.
- ▶ The resource costs for car group upgrades should be reduced. Our recommendation is to replace the human task by an automatic service that assigns the upgrade based on rules and certain conditions.
- ▶ The number of cases where the customer does not return the car according to the contract should be reduced. Our recommendation is to monitor these cases closely at each location and make customers explicitly aware that they have to call the agency in cases of contract extensions.
- ▶ The branch agent human resource costs for rental agreement extension tasks are very high. Our recommendation is to outsource the task to a central call center that has lower human resource costs.

11.5 Creating the To-Be process model

We have defined the To-Be model as being an *implementation model*, a model that is deployed to WebSphere Process Server. Therefore in creating this model we must view it from two perspectives: business analyst and IT development. The To-Be model cannot be completed in isolation, and the business analyst and IT developer must work in conjunction to ensure a deployable model.

This section shows the various views offered in WebSphere Business Modeler for both the business analyst and enterprise architect, how they collaborate on process model development, and considerations that must be taken into account during the model development.

The roles have the following responsibilities:

- ▶ The *business analyst* role creates the business process flow; ensures that the business process is correct and that any business rules are documented.
- ▶ The *enterprise architect* role ensures that the business model is in a state suitable for export into WebSphere Integration Developer; that is, has correct execution logic suitable for export to WebSphere Process Server.

11.5.1 ITSO Speedy Rentals management team decisions

Based on the analyst summary of the As-Is business model, the ITSO Speedy Rentals management team agreed on these implementation requirements:

- ▶ The car group upgrade task will be automated and implemented by a set of flexible rules that can be updated and adjusted over time.
- ▶ The car group stock will be monitored by management on a weekly basis. The new implementation must provide detailed, near-real-time information for each location.
- ▶ The number of calls about vehicles not returned on time will be monitored by management on a weekly basis.
- ▶ Calls about rental agreement extensions will be outsourced to a call center.
- ▶ To reduce the number of complaints on the time frame it takes to return the vehicle, monitoring will be set up in each location. A new implementation based on the latest RFID technology will be set up to identify the rental vehicle when it drives onto the branch property. The new car rental fulfillment implementation will integrate the RFID subsystem and measure the time frame until the task Vehicle Return handover is completed.
- ▶ For loyal customers, a new discount will be applied. The automatic discount calculation must be based on a flexible and changeable set of rules.

11.5.2 Creating the To-Be project

The first step is to create a new project for the To-Be implementation model. Although we could create the To-Be process model in the As-Is project, we found that for clarity this was best practice for maintaining both models.

The As-Is model details only business flow, and therefore is not suitable for deployment (which will be based on execution flow for WebSphere Process Server). We create a simplified execution flow based on the As-Is model.

1. Switch to WebSphere Process Server Modeling mode.
 - a. Select **Modeling** → **Mode** → **WebSphere Process Server**. Any errors that result in the As-Is process catalog can be ignored because the As-Is model contains constructs unsuitable for WebSphere Process Server. (See 11.6, “Modeling for WebSphere Process Server” on page 219.)
 - b. Ensure that the 4-pane layout button is selected.

Note: If you wish to model a business process for implementation in WebSphere Process Server, we recommend choosing WebSphere Process Server Modeling Mode.

If a model has been created in another mode, you can switch to WebSphere Process Server mode and view the validation errors. These errors must be fixed before the model can be exported to WebSphere Integration Developer.

2. Create a new To-Be project.
 - a. Select **File** → **New Business Modeling Project**.
 - b. Enter **ITSO Speedy Rentals To-Be Project** in the New project name field and **Vehicle Fulfilment Process ToBe** in the Process name field. Click **Finish**.
3. Export the As-Is project.
 - a. Right-click **ITSO Speedy Rentals As-Is Project** and select **Export**.
 - b. Select **WebSphere Business Modeler project (.zip)** and click **Next**.
 - c. Select a directory to hold the exported .zip file, for example `c:\WMB\Exported Models\`.
 - d. Ensure that **ITSO Speedy Rentals As-Is Project** is selected as the Project to export.
 - e. Select the **Export entire project** radio button.
 - f. Enter **As-Is Project** in the Export filename field.

- g. Deselect the **Include simulation snapshots** box.
 - h. Deselect the **Include business measures** box.
 - i. Select **Finish** to export the .zip project file.
4. Import the As-Is project into the new To-Be project.
- a. Right-click **ITSO Speedy Rentals To-Be Project** and select **Import**.
 - b. Select **WebSphere Business Modeler project (.zip)** and click **Next**.
 - c. Select the **As-Is Project.zip** file to import.
 - d. Click **Finish** to import the project.
 - e. During the import into existing catalogs, WebSphere Business Modeler asks whether the imported catalogs are to be merged. Click **Yes to all** (Figure 11-31).

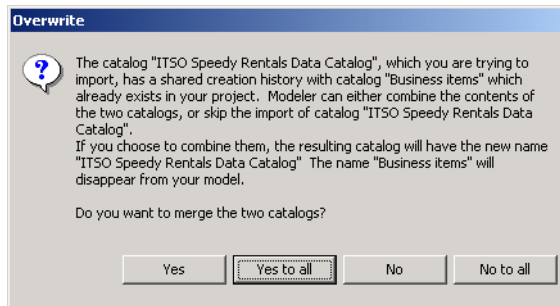


Figure 11-31 Merge catalogs

Figure 11-32 on page 212 shows our new To-Be project tree with the imported As-Is project.

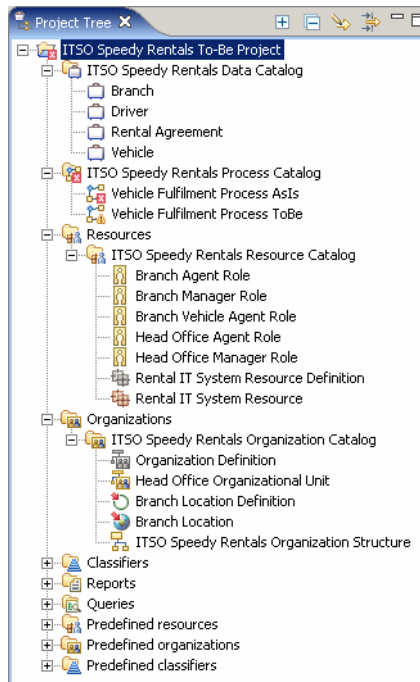


Figure 11-32 To-Be project tree

Note: Ignore the errors in the project tree. They are caused by the model containing constructs unsuitable for WebSphere Process Server. These errors will be removed later in this section.

11.5.3 Refining the business items

The As-Is business process model that was created by the business analyst has business items defined in terms of attributes required to complete a business task. The business analyst is not concerned with the way these attributes are translated into a form suitable for execution in WebSphere Process Server.

The enterprise architect must ensure that the business items created by the business analyst in the As-Is model are refined into a form suitable for execution in the To-Be implementation model.

The Business Item Catalog was already imported from our As-Is model, so make any changes required now. For example, the business analyst defined most of the business item attributes as `String`, so our implementation model should

Figure 11-33 shows the attribute editor with additional options in WebSphere Process Server mode.

Figure 11-33 Business item attribute editor

In our Branch business item, all attributes are defined as type `String`. When this business item is exported for WebSphere Process Server into WebSphere Integration Developer, the development team will know to implement these attributes as type `String`.

Chapter 11. Model with WebSphere Business Modeler 213

11.5.4 Creating the To-Be process flow

This example scenario assumes that the business analyst and enterprise architect have reviewed the simulation results and refined the As-Is business process, removing any bottlenecks, or replacing manual tasks with automated tasks.

The next steps involve creating the new To-Be business process implementation model.

1. Create the To-Be model tasks using the process editor and the following steps. The Start node and Stop nodes are added automatically.

Table 11-12 shows the tasks to be added to this process model.

Table 11-12 To-Be process model activities

Element name	Element type
While Loop: Reserve Vehicle	While loop
Vehicle Collection	Local task
Rental Agreement Status Set To Active	Local task
While Loop: Wait on Vehicle Return	While loop
Vehicle Return handover	Local task
Return Vehicle To System	Local task
Calculate Discounts	Local task
Payment	Local task

- a. In the project tree, expand **ITSO Speedy Rentals Process Catalog**, right-click **Vehicle Fulfilment Process AsIs**, and select Delete. In the Confirm Delete dialog, click **Yes**.
 - b. Double-click **Vehicle Fulfilment Process ToBe** to open the process editor.
 - c. Add the elements listed in Table 11-12, ensuring that the correct activity type is selected from the process editor palette. Figure 11-34 on page 215 shows the layout of each activity element.
 - d. Remove the Start node because it is not needed.
2. Create the connections in the process to create the process model shown in Figure 11-34 on page 215.

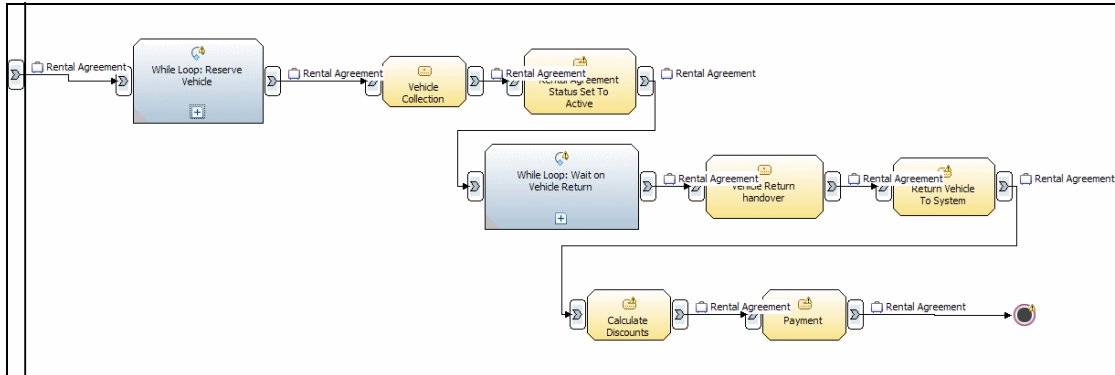



Figure 11-34 Vehicle Fulfillment process model

- a. Connect the activities using the  button, using details in Table 11-13.

Table 11-13 To-Be main process connections

“From” element	“To” element	Connection name	Associated data
External Boundary	While Loop: Reserve Vehicle	Rental Agreement Start Connection	Rental Agreement
While Loop: Reserve Vehicle	Vehicle Collection	Vehicle Pickup Connection	Rental Agreement
Vehicle Collection	Rental Agreement Status Set To Active	Activate Rental Agreement Connection	Rental Agreement
Rental Agreement Status Set To Active	While Loop: Wait on Vehicle Return	Rental Status Connection	Rental Agreement
While Loop: Wait on Vehicle Return	Vehicle Return handover	Return Vehicle Connection	Rental Agreement
Vehicle Return handover	Return Vehicle To System	Vehicle Returned Connection	Rental Agreement
Return Vehicle To System	Calculate Discounts	Calculate Discounts Connection	Rental Agreement
Calculate Discounts	Payment	Payment Connection	Rental Agreement
Payment	Stop Node	End Vehicle Fulfilment Connection	Rental Agreement

- b. Open the 4-pane view by clicking the  button. The project tree displays.
- c. In the project tree, expand **ITSO Speedy Rentals Data Catalog**.


- d. For each connection that is created, select the **Rental Agreement** business item, and drag and drop it onto the connection. The connection type will change from String to Rental Agreement.

Note: For simplicity we have used Rental Agreement business item as the input and output data item for all activity elements.

- e. Ensure that each connection is assigned the correct name, as shown in Table 11-13 on page 215.
 - f. Save the process.
3. Add tasks to the While Loop: Reserve Vehicle while loop activity.
- a. Select **While Loop: Reserve Vehicle**. Click the + symbol to expand this activity and open the while loop process editor.
 - b. Add the new elements as listed in Table 11-14 using the process editor palette to draw the model.

Table 11-14 Reserve Vehicle while loop

Element name	Element type
Local Information Repository	Local repository
Reserve Vehicle In System	Local task
Vehicle Available?	Simple decision
Assign Higher Group Vehicle	Local task
Stop Node	Stop

- c. Change the percentages of the Vehicle Available simple decision to indicate that Yes is expected 90% of the time, and No is expected 10%.
- d. Click **Local Information Repository**, and in the Attributes view click **Browse** next to Associated data. Select **Rental Agreement** and click **OK**.
- e. Connect the activities using the  button as shown in Figure 11-35 on page 217 and detailed in Table 11-15 on page 217. Ensure that the connection names and associated data are also set correctly.

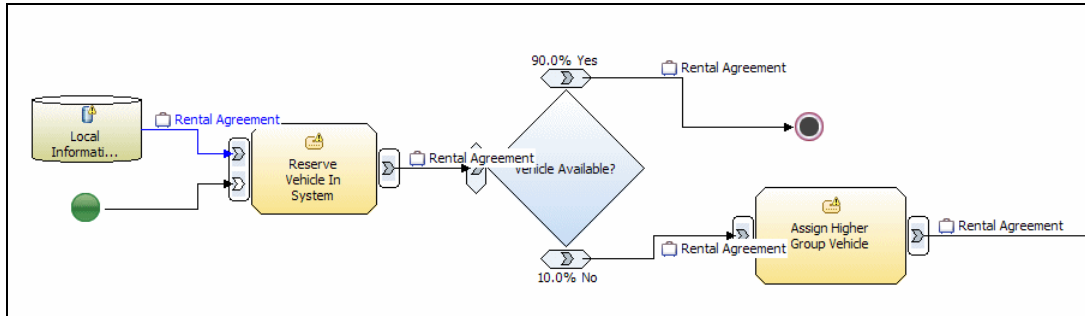


Figure 11-35 While Loop: Reserve Vehicle process


Table 11-15 While Loop: Reserve Vehicle connections

“From” element	“To” element	Connection name	Associated data
Local InformationRepository	Reserve Vehicle In System	None	Rental Agreement
Start Node	Reserve Vehicle In System	Start Connection	None
Reserve Vehicle In System	Vehicle Available?	Is Vehicle Available Connection	Rental Agreement
Vehicle Available? (yes)	Stop Node	End Connection	Rental Agreement
Vehicle Available? (no)	Assign Higher Group Vehicle	Assign a Higher group Vehicle Connection	Rental Agreement
Assign Higher Group Vehicle	External Boundary	Vehicle Assigned Connection	Rental Agreement

- f. Save the process. Ignore the errors on the Vehicle Available decision — they will be fixed later.
- g. Return to the top-level process by right-clicking in an empty space in the process editor and selecting **Parent** from the context menu.
4. Add tasks to the while loop activity While Loop: Wait on Vehicle Return.
 - a. Select **While Loop: Wait on Vehicle Return**. Click the + symbol to expand this activity and open the while loop’s process editor.
 - b. Add the new elements as listed in Table 11-16 on page 218 using the process editor palette to draw the model.

Table 11-16 Wait on Vehicle Return while loop

Element Name	Element Type
Local Information Repository	Local repository
Check If Vehicle Returned	Local task
Vehicle Overdue?	Simple decision
Update Rental Agreement	Local task
Merge	Merge

- c. Change the percentages of the Vehicle Overdue simple decision to indicate that Yes is expected 20% of the time, and No is expected 80%.
- d. Click **Local Information Repository**, and in the Attributes view click **Browse** next to Associated data. Select **Rental Agreement** and click **OK**.
- e. Connect the activities using the  button as shown in Figure 11-36 and detailed in Table 11-17. Ensure that the connection names and associated data are also set correctly.

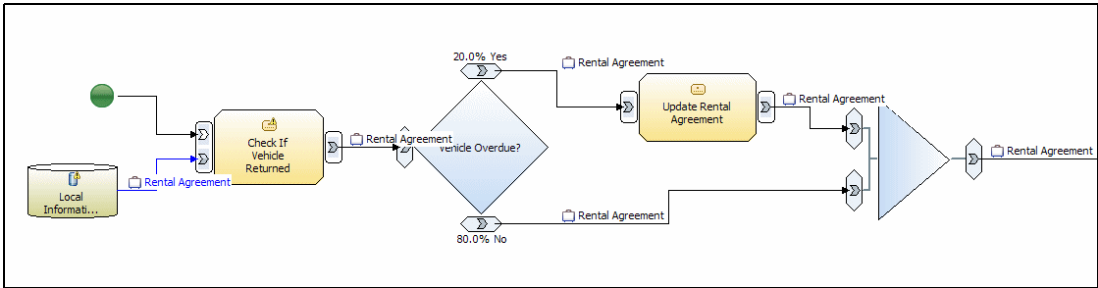


Figure 11-36 While Loop: Wait On Vehicle Return process

Table 11-17 While Loop: Wait on Vehicle Return connections

“From” element	“To” element	Connection name	Associated data
Local Information Repository	Check If Vehicle Returned	None	Rental Agreement
Start Node	Check If Vehicle Returned	Rental Status Connection	None
Check If Vehicle Returned	Vehicle Overdue?	Is Vehicle Overdue Connection	Rental Agreement
Vehicle Overdue? (yes)	Update Rental Agreement	Vehicle Is Overdue Connection	Rental Agreement

"From" element	"To" element	Connection name	Associated data
Vehicle Overdue? (no)	Merge	Unchanged Rental Agreement Connection	Rental Agreement
Update Rental Agreement	Merge	Updated Rental Agreement Connection	Rental Agreement
Merge	External Boundary	End Connection	Rental Agreement

- f. Save the process. Ignore the errors on the Vehicle Overdue decision. We will fix those later.
- g. Return to the top-level process by right-clicking in an empty space in the process editor and selecting **Parent** from the context menu.

11.6 Modeling for WebSphere Process Server

Now that the base To-Be process model is completed, add additional detail that is required for export to WebSphere Process Server.

WebSphere Business Monitor offers business-level modeling that is not platform specific, therefore not all modeling elements supported in WebSphere Business Monitor can be transformed into WebSphere Process Server constructs.

Note: When your model is first saved in WebSphere Process Server mode, several WS-BPEL transformation validation errors and warnings are shown in the Error View with a ZBL error code. If there are any errors in the model, then incorrect code will be exported or will not be able to export at all.

The following section details the steps required for modeling in WebSphere Process Server mode.

11.6.1 WebSphere Process Server mode considerations and unsupported elements

In the WebSphere Process Server model, we found during implementation of this scenario that the following best-practice approach produced the fewest errors and warnings:

- Be sure that for each element input criterion, there is at least one output criterion.

- ▶ Local task elements may have at most only one output criterion associated to an input criterion. For multiple output criteria, use a decision element to model exclusive branching.
- ▶ A model must not contain any downstream tasks connected back to upstream tasks (for example, in our As-Is model, a connection from the Update Rental Agreement local task's output back to the Return Date Expired local task). In our To-Be model this flow has been replaced with loop constructs.

WebSphere Business Modeler contains some model elements that cannot be transformed into WS-BPEL either because there is no equivalent WS-BPEL or because semantics are different. The following elements are not supported in WebSphere Process Server mode and therefore cannot be selected from the palette:

- ▶ Notification broadcaster
- ▶ Notification receiver
- ▶ Observer
- ▶ Timer
- ▶ Global repository
- ▶ Do-while loop
- ▶ For loop

11.6.2 Defining element attributes

WebSphere Process Server requires correct definitions for the following element attributes:

- ▶ Inputs and outputs
- ▶ Input branches and output branches
- ▶ Classifiers, resources, and roles
- ▶ Input logic and output logic
- ▶ Advanced input logic and advanced output logic
- ▶ Loop conditions

Inputs and outputs

All input and output values must be set. In our scenario we name all input and output for each local task, ensure that for each input there is an associated output, ensure that each input and output has a unique name, and that all input

and output have been assigned associated data. (For simplicity, this scenario uses the Rental Agreement business item.)

1. For each element in the process (including the elements in the while loop), run these steps using the values in Table 11-18 through Table 11-20 on page 222:
 - a. Click the element in the process model, for example While Loop: Reserve Vehicle.
 - b. In the Attributes view, click the **Inputs** tab and enter the input names.
 - c. In the Attributes view, click the **Outputs** tab and enter the output names.

Note: Default names are assigned to each input and output. We rename them here with meaningful names.

Table 11-18 To-Be process model inputs and outputs

Element name	Input name	Output name
While Loop: Reserve Vehicle	Reserve Vehicle Input	Reserve Vehicle Output
Vehicle Collection	Vehicle Collection Input	Vehicle Collection Output
Rental Agreement Status Set To Active	Rental Status Active Input	Rental Status Active Output
While Loop: Wait on Vehicle Return	Wait on Return Input	Wait on Return Output
Vehicle Return handover	Vehicle Handover Input	Vehicle Handover Output
Return Vehicle To System	Vehicle Return Input	Vehicle Return Output
Calculate Discounts	Calculate Discount Input	Calculate Discount Output
Payment	Payment Input	Payment Output

Table 11-19 While Loop: Reserve Vehicle inputs and outputs

Element name	Input name	Output name
Local Information Repository	None	None
Reserve Vehicle In System	Repository Input Reserve Vehicle In System Input	Reserve Vehicle In System Output

Element name	Input name	Output name
Vehicle Available?	Vehicle Available Input	Vehicle Available Check Output:Yes Vehicle Available Check Output:No
Assign Higher Group Vehicle	Assign Higher Group Vehicle Input	Assign Higher Group Vehicle Output
Start Node	None	None
Stop Node	None	None

Table 11-20 While Loop: Wait on Vehicle Return inputs and outputs

Element name	Input name	Output name
Local Information Repository	None	None
Check If Vehicle Returned	Rental Overdue Status Input Start Input	Rental Overdue Status Output
Vehicle Overdue?	Rental Status Input	Vehicle Is Overdue Vehicle Not Overdue
Update Rental Agreement	Update Rental Agreement Input	Update Rental Agreement Output
Merge	Updated Rental Agreement Input Unchanged Rental Agreement Input	Rental Agreement Output
Start Node	None	None

2. Change the process input attributes name:
 - a. In the process editor, click anywhere in the white space to display the process attributes in the Attributes view.
 - b. Click the **Inputs** tab. Rename the default name of Input to Start Vehicle Fulfilment Process Input.
3. Save the process.

Input branches and output branches

Our model contains two simple decisions and a merge element, all of which need input and output branches defined.

1. For both of the simple decisions (located in the while loops), perform the following changes using values in Table 11-21:
 - a. Click the element in the process model, for example **Vehicle Available?**.
 - b. In the Attributes view, click the **Input branches** tab and enter the input branch names.
 - c. In the Attributes view, click the **Output branches** tab and enter the probability under the Probability column.

Table 11-21 Decision elements

Element name	Input branch name	Output branch probability
Vehicle Available?	Vehicle Available Input Branch	Yes:90% No:10%
Vehicle Overdue?	Vehicle Overdue Input Branch	Yes:20% No:80%

2. Select the merge element in the While Loop: Wait on Vehicle Return while loop and perform the following steps, using values in Table 11-22:
 - a. In the Attributes view, click the **Input branches** tab and enter the input branch names.
 - b. In the Attributes view, click the **Output branches** tab and enter the output branch name.

Table 11-22 Merge element

Element name	Input branch name	Output branch name
Merge	Update Rental Agreement Input Branch	Rental Agreement Output Branch
	Unchanged Rental Agreement Input Branch	

3. Save the process.

Classifiers, resources, and roles

As with the As-Is model, add our defined classifiers to specify whether a task is a human task or automatic, then assign a role or resource to each of the activities.

1. Assign the classifiers, resources, and roles to each element in the process, using the information in Table 11-23 through Table 11-25 on page 225. For each element, follow these steps:
 - a. Click the element in the process model.
 - b. In the Attributes view, click the **Classifier** tab. Click **Add**, select the appropriate classifier, and click **OK**.
 - c. To define a resource (for IT System resources only), in the Attributes view, click the **Resources** tab. Click **Add** in the Bulk resource requirements section. Set the name in the Name column, and set the resource type in the Bulk resource column.
 - d. To define a role (for manual activities only), in the Attributes view, click the **Resources** tab. Click **Add** in the Role requirements section. Set the name in the Name column, and set the role type in the Role column. Additionally for every role, set the Resource definition to a value of **Staff**.

Table 11-23 To-Be process model classifiers, resources, and roles

Element name	Classifier	Role/resource name	Role/resource type
Vehicle Collection	Manual Activity	Branch Vehicle Agent Role Requirement	Branch Vehicle Agent Role
Rental Agreement Status Set To Active	IT System	Rental IT System Bulk Requirement	Rental IT System Resource
Vehicle Return handover	Manual Activity	Branch Vehicle Agent Role Requirement	Branch Vehicle Agent Role
Return Vehicle To System	IT System	Rental IT System Bulk Requirement	Rental IT System Resource
Calculate Discounts	IT System	Rental IT System Bulk Requirement	Rental IT System Resource
Payment	IT System	Rental IT System Bulk Requirement	Rental IT System Resource

Table 11-24 While Loop: Reserve Vehicle classifiers, resources, and roles

Element name	Classifier	Role/resource name	Role/resource type
Reserve Vehicle In System	IT System	Rental IT System Bulk Requirement	Rental IT System Resource
Assign Higher Group Vehicle	IT System	Rental IT System Bulk Requirement	Rental IT System Resource

Table 11-25 While Loop: Wait on Vehicle Return classifiers, resources, and roles

Element name	Classifier	Role/resource name	Role/resource type
Check If Vehicle Returned	IT System	Rental IT System Bulk Requirement	Rental IT System Resource
Update Rental Agreement	Manual Activity	Branch Agent Role Requirement	Branch Agent Role

2. Save the process.

Input logic and output logic

These attributes enable you to detail the input and output criteria and specify any preconditions. Our scenario does not apply any preconditions, so we have used the default settings for both input logic and output logic. Figure 11-37 shows the Input Logic view for Vehicle Collection.

The screenshot shows the 'Attributes - Vehicle Collection' window with the 'Input Logic' tab selected. The 'Input criteria' section contains a table with the following data:

Name	Vehicle Collection Input	Criterion
Input Criterion	<input checked="" type="checkbox"/>	Vehicle Collection Input

Below the table are buttons for 'Add', 'Remove', and 'Modify'. The 'Preconditions' section below it has a text area for preconditions, an 'Add' button, a 'Remove' button, and fields for 'Name', 'Description', and 'Expression' with 'Clear' and 'Edit...' buttons at the bottom right.

Figure 11-37 Input Logic attribute view

Advanced input logic and advanced output logic

These attributes enable you to specify other attributes for input and output criterion. You can put constraints on the input and output criterion, such as specifying a condition that the Rental Agreement input must be active. Our scenario uses the default settings for both advanced input and output logic tabs because no constraints are defined. Figure 11-38 shows an Advanced Input Logic view.

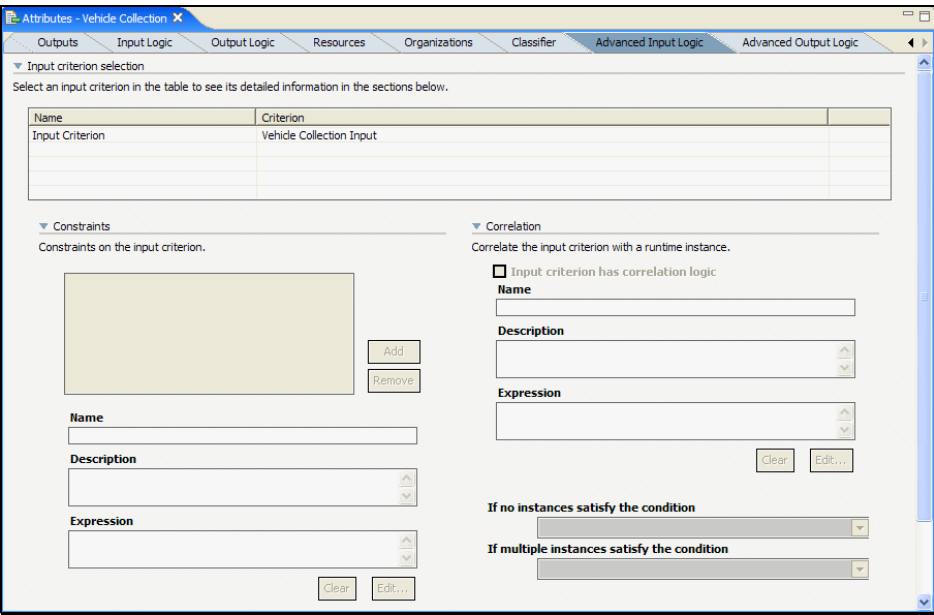


Figure 11-38 Advanced Input Logic attribute view

Loop conditions

Loop condition elements used in the model have this additional tab. This setting is used to define the entry and exit logic for the loop. WebSphere Business Modeler provides an Expression Builder for creating loop condition logic. For more details, refer to 11.6.4, “Adding condition logic” on page 232.

11.6.3 Technical implementation specification

This section describes technical attributes, which enable you to specify implementation details.

Open the Technical Attributes view

In some WebSphere Business Modeler installations the Technical Attributes view is not turned on by default. If this view is not visible, follow these steps to open it:

1. In the main menu, click **Window** → **Show View** → **Other** to open the Show View dialog.
2. Expand **Business modeler views**, click **Technical Attributes View** and click **OK**.

Working with technical attributes

The use of technical attributes enables the enterprise architect to capture any technical information required for implementation within the model, thus reducing the amount of development time required for implementation. The technical attributes for a process, task, or service are used when the model is exported to WebSphere Process Server.

Process level technical attributes

To examine the technical attributes for a process, follow these steps:

1. Click the **Technical Specification** tab in the process editor (Figure 11-39).

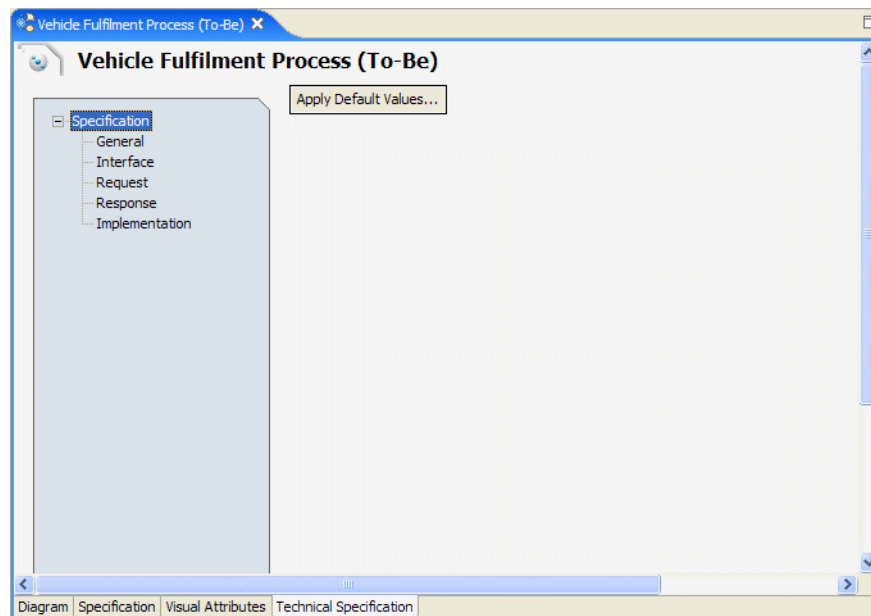


Figure 11-39 Technical Specification: default settings view

2. Apply technical specification default settings.
 - a. Click the **Apply Default Values** button to automatically generate the required values for your implementation model.
 - b. In the Apply Default Values dialog, click **Yes**.
3. Examine some of the default technical attributes.
 - a. Click the **General** tab. This tab contains details for defining WS-BPEL information (Figure 11-40).

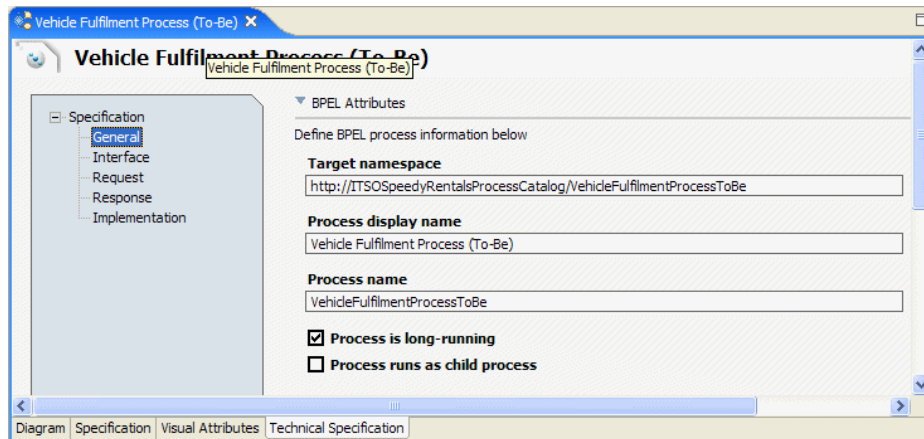


Figure 11-40 Technical Specification: General view

- b. Click the **Interface** tab. This tab contains details for specifying the WSDL interface elements. Our scenario uses the default values, so it does not require further definition.

- c. Click the **Request** tab. This tab contains details for defining WSDL or WS-BPEL information (Figure 11-41). Under Operation type, click **One way operation**.

Vehicle Fulfilment Process (To-Be)

Specification
General
Interface
Request
Response
Implementation

Input criterion section

Select an input criterion in the table to see the WSDL and BPEL information in the sections below

Input criterion name	Corresponding output criterion name	Valid operation type	Selected operation type
Input Criterion	Output Criterion	Request / response, one way	One way

Operation type

Select the type of operation

☐ Request / Response operation
☒ One way operation

WSDL Attributes

Define the WSDL interface information

Operation name
InputCriterion

WSDL Message Details

Define the WSDL message that represents the in

Message name
InputCriterion

Part name
InputCriterion

BPEL Attributes

Define the BPEL receive activity information

Activity display name
Vehicle Fulfilment Process (To-Be) Receive

Activity name
VehicleFulfilmentProcessToBe_InputCriterion

Diagram | Specification | Visual Attributes | Technical Specification

Figure 11-41 Technical Specification: Request view

- d. Click the **Response** tab. This tab contains details for defining WSDL and WS-BPEL information (Figure 11-42).

Vehicle Fulfilment Process (To-Be)

Specification
General
Interface
Request
Response
Implementation

Output criterion section

Select an output criterion in the table to see the WSDL and BPEL information in the sections below

Output criterion name	Corresponding input criterion name	Response type
Output Criterion	Input Criterion	No response

WSDL Attributes

Define the WSDL interface information

WSDL Message Details

Define the WSDL message that represents the output of a

Message name
Part name

BPEL Attributes

Define the BPEL reply activity information

Activity display name

Activity name

Diagram | Specification | Visual Attributes | Technical Specification

Figure 11-42 Technical Specification: Response view

- e. Click the **Implementation** tab. This tab has details for the element's SCA (Service Component Architecture) information. This information is used to implement the element as an SCA component in WebSphere Integration Developer. The default values set has been used (Figure 11-43).

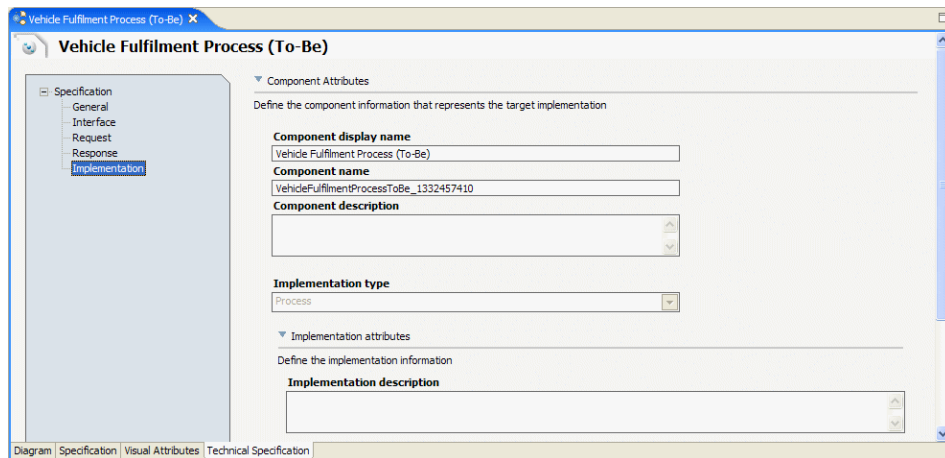


Figure 11-43 Technical Specification: Implementation view

Task level technical attributes

The next steps define how tasks are transformed to BPEL and SCA implementation artifacts.

Click the **Diagram** tab in the process editor, and perform the following steps:

1. Click the **Vehicle Collection** task.
2. Click the **Technical Attributes View** tab in the Attributes window to display the technical attribute specifications.

3. Click the **Implementation** tab and set the Implementation type to **Human Task** as shown in Figure 11-44. In the SCA model, this task is now exported as human task implementation.

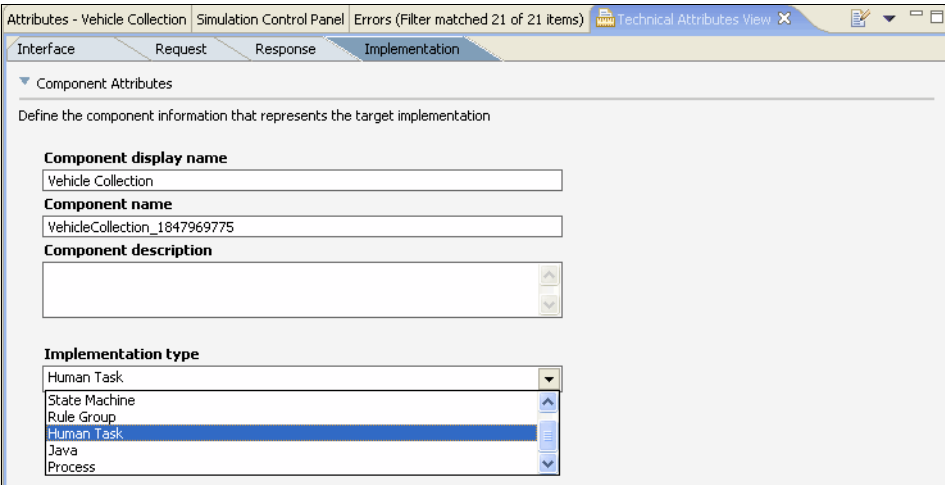


Figure 11-44 Technical Implementation Type setting on tasks

Note: The generated default Component name has a unique number appended to the task name as shown in Figure 11-44. For consistency, we recommend changing the default set for the Component name according to the naming conventions that were set up by the project implementation team.

4. Repeat these steps to set the implementation type property for the tasks shown in Table 11-26.

Note: Tasks that have no implementation type specified will be exported with no implementation, so service implementations must be defined in WebSphere Integration Developer.

Table 11-26 Element implementation type settings

Element name	Implementation type
Assign Higher Group Vehicle	Rule Group
Check If Vehicle Returned	Process
Update Rental Agreement	Human Task

Element name	Implementation type
Vehicle Return handover	Human Task
Calculate Discounts	Rule Group

5. Save the process.

11.6.4 Adding condition logic

Our To-Be model uses two condition elements: decision and loop. This section shows how to apply condition logic to our model for each type of condition element.

Loop conditions

You can define entry and exit logic for loops by using the Expression Builder tool to define conditions.

Our scenario defines condition logic for our two loops: Reserve Vehicle and Wait On Vehicle Return. Perform the following steps:

1. Click the **Reserve Vehicle** element in the model.
2. In the Attributes view, click the **Loop condition** tab.
3. Click **Edit** to open the Expression Builder.
4. In the Expression Builder, perform the following steps:
 - a. Set First term to **Modeling artifact**.
 - b. Expand **ITSO Speedy Rentals Process Catalog** → **Vehicle Fulfilment Process ToBe** → **While Loop: Reserve Vehicle** → **Reserve Vehicle Input** → **Vehicle Details** → **Available**. This will be the data item used to determine whether the vehicle is available for rental.
 - c. Select the **is equal to** operator.
 - d. Select **Boolean** from the Second term menu.
 - e. Select **false** from the Boolean value menu.
 - f. Click **Apply** and then **OK** to save the changes.
5. Repeat these steps for the Wait On Vehicle Return condition loop, using Expression Building to build the following condition (Figure 11-45 on page 233):
 - a. Select **Wait On Return Input** → **Rental Status**. This will be the data item that is used to determine whether the vehicle is overdue.
 - b. Select the **is equal to** operator.

- c. Select **Text** from the Second term menu.
- d. Enter out in the Text value area.
- e. Select **Apply**, then **OK** to save the changes.

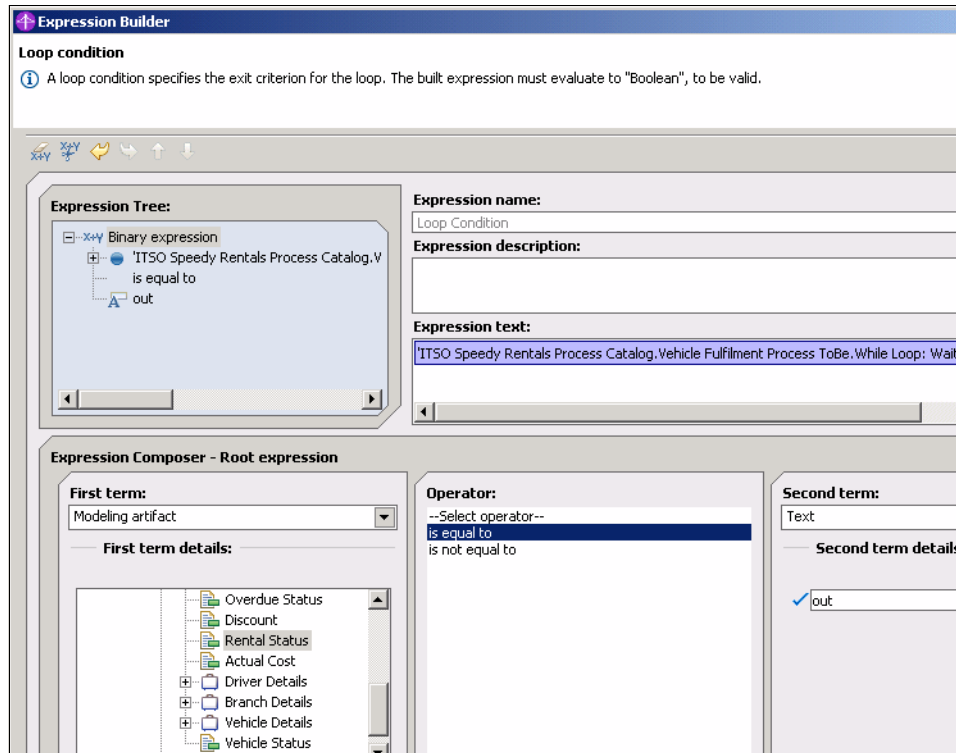


Figure 11-45 Wait on Vehicle Return condition loop

6. Save the process.

Simple decision conditions

Decision elements are used in the model for Vehicle Available? and Vehicle Overdue?. The conditions for these decisions are set on the output branches of the decisions; that is, a binary decision is made on specified criteria. These criteria are set using the Expression Builder.

Perform the following steps to define a condition for the Vehicle Available? simple decision:

1. Click **Vehicle Available?** in the process editor.
2. In the Attributes view click the **Output branches** tab to display the two output branches (that is, the binary decision result, in our case Yes and No outputs).

3. Click **Yes** then **Edit** to start the Expression Builder.
4. Use the Yes condition if the Available attribute from the Vehicle business item is true. Configure this as follows:
 - a. Select **Modeling artifact**, then expand the process catalog **ITSO Speedy Rentals Process Catalog** → **Vehicle Fulfilment Process ToBe** → **While Loop: Reserve Vehicle** → **Vehicle Available?** → **Vehicle Available Input** → **Vehicle Details**, and select **Available**. This data item will be used to determine whether the vehicle is available for rental.
 - b. Select the **is equal to** operator.
 - c. Select **Boolean** from the Second term menu.
 - d. Select **true** from the Boolean value menu.
 - e. Select **Apply** and then **OK**.
5. In the Output branches tab of the Attributes view, click **No**. The text in the Expression field has been completed automatically to check that the Available attribute is not equal to Yes.

Repeat the steps above for the Vehicle Overdue? simple decision element:

1. Click **Vehicle Overdue?** in the process editor.
2. In the Attributes view, click the **Output branches** tab to show the two output branches, Yes and No.
3. Click **Yes** then **Edit** to start the Expression Builder.
4. Use the Yes condition if the Available attribute from the Vehicle business item is true. Configure this as follows:
 - a. Select **Modeling artifact**, then expand the process catalog **ITSO Speedy Rentals Process Catalog** → **Vehicle Fulfilment Process ToBe** → **While Loop** → **Vehicle Overdue?** → **Rental Status Input**, and select **Overdue Status**. This data item will be used to determine whether the vehicle is available for rental.
 - b. Select the **is equal to** operator.
 - c. Select **Text** from the Second term menu.
 - d. Enter Yes as the Second term details.
 - e. Select **Apply** and then **OK**.
5. In the Output branches tab of the Attributes view, click **No**. The text in the Expression field has been completed automatically to check that the Available attribute is not equal to Yes.
6. Save the process. There should be no errors (only warnings) in the process model.

11.7 Creating the business measures model

A business measures model enables you to define metrics to measure a process in action, real-time. Define a business measures model in WebSphere Business Modeler and export it to WebSphere Business Monitor.

We use the business measures model to make the following measurements:

- ▶ The times that a higher group vehicle is assigned. This affects the overall cost and measures how successfully ITSO Speedy Rentals manages its inventory.
- ▶ The due-date violation (customer is late returning a rental). This affects cost, highlights customer behavior, and measures communication with customers.
- ▶ Special case contracts (business situations) where a customer generates three or more due-date violations for a rental.
- ▶ Vehicle handover time, which measures customer satisfaction.

This section describes how to define a business measures model for our process using the following steps:

- ▶ Creating business measures
- ▶ Creating an event for a business situation
- ▶ Dimensional analysis
- ▶ Stopwatches
- ▶ Counters
- ▶ Key performance indicators (KPIs)
- ▶ Aggregate metrics
- ▶ Synchronizing the business measures model

11.7.1 Creating business measures

This section describes how to create the business measures in the project tree. We populate these business measures with data in the following sections.

1. Make business measures visible in the project tree (Figure 11-46).
 - a. Click the **Filters** icon in the project tree.
 - b. Deselect the **Predefined business measures events** and **Business measures events** filters.
 - c. Click **OK**.

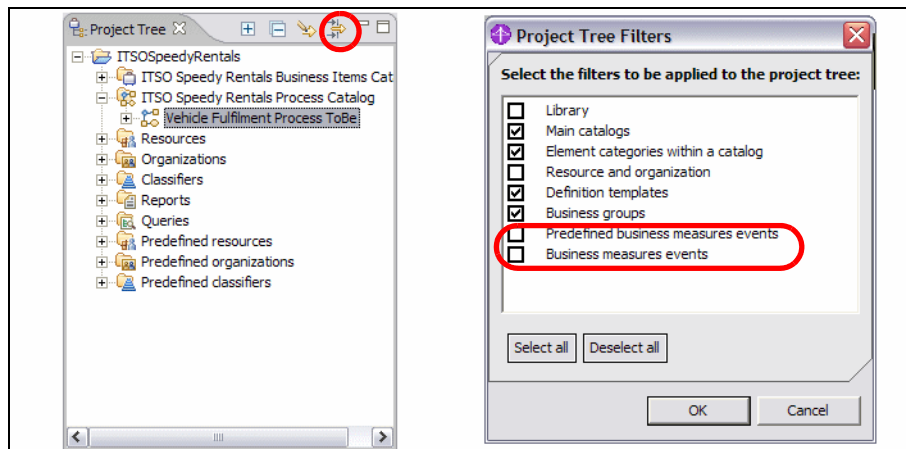


Figure 11-46 Project tree filters

2. In the project tree, expand **ITSO Speedy Rentals Process Catalog**, right-click **Vehicle Fulfilment Process ToBe**, and select **Create Business Measures**.

This option is available only if you have not previously created business measures for this process. Otherwise, you must select **Synchronize** to synchronize the process with the business measures that already exist.

3. The Select Process Elements dialog opens (Figure 11-47), displaying the elements of the process that are available for monitoring. Leave all elements checked and click **OK**.

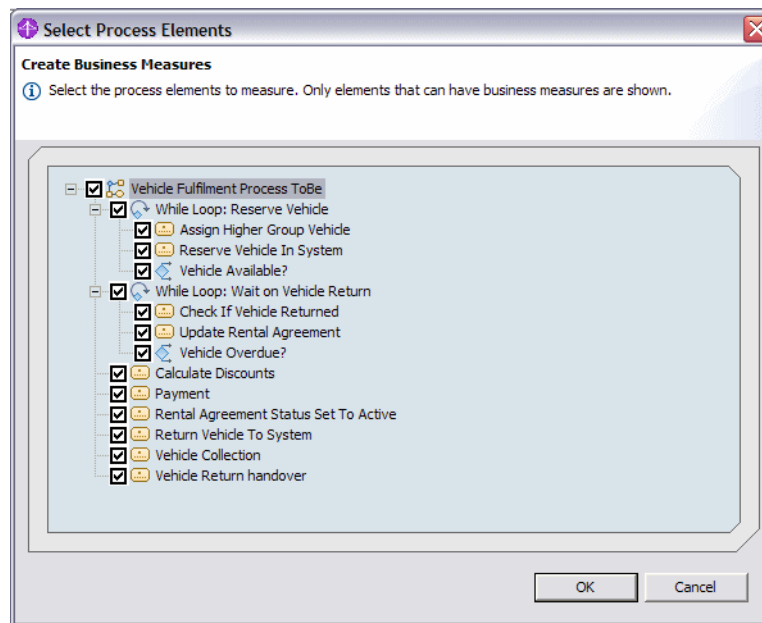


Figure 11-47 Create Business Measures dialog box

4. A new business measures node is created in the project tree under the process node (Figure 11-48). The business measures will open in an editor.

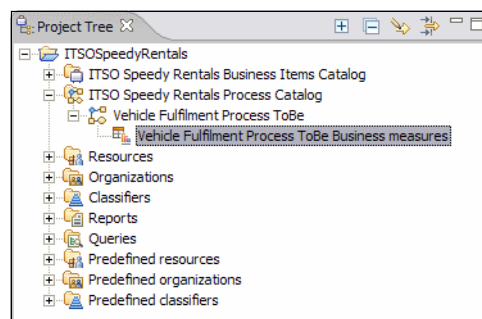


Figure 11-48 Business measures created

5. The business measures editor opens on the KPIs and Aggregate Metrics tab (Figure 11-49).

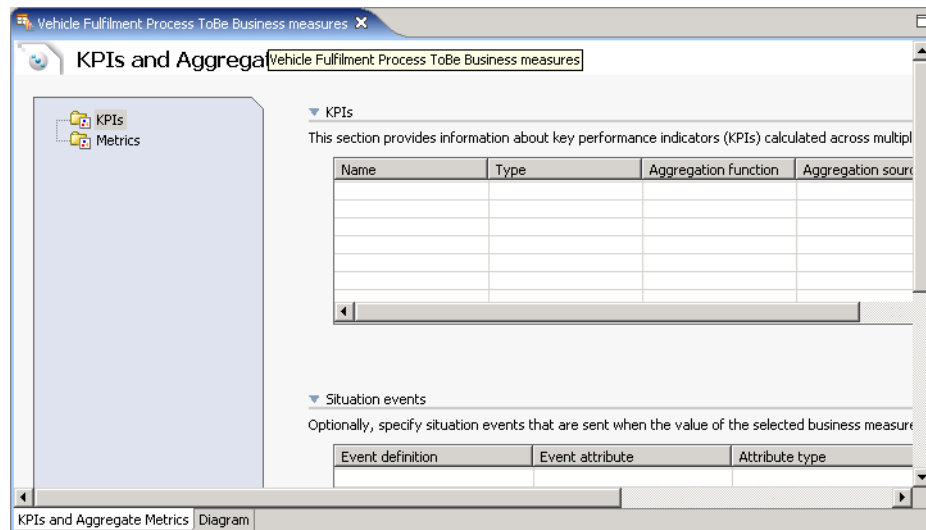


Figure 11-49 Business measures: KPIs and Aggregate Metrics

6. Click the **Diagram** tab to see a graphical representation of the business measures model. The Attributes view displays attributes of the business measures model.

You can specify the maximum duration that is expected for the top-level process by changing the value in the Maximum duration field. If the process takes longer to run than the maximum duration you specify, the Is Delayed predefined metric is set to true. You can monitor this metric in WebSphere Business Monitor.

Additionally, you can specify the date and time at which a top-level process becomes active in a runtime environment by changing the value of the Valid from field. We use the default values for each of these settings.

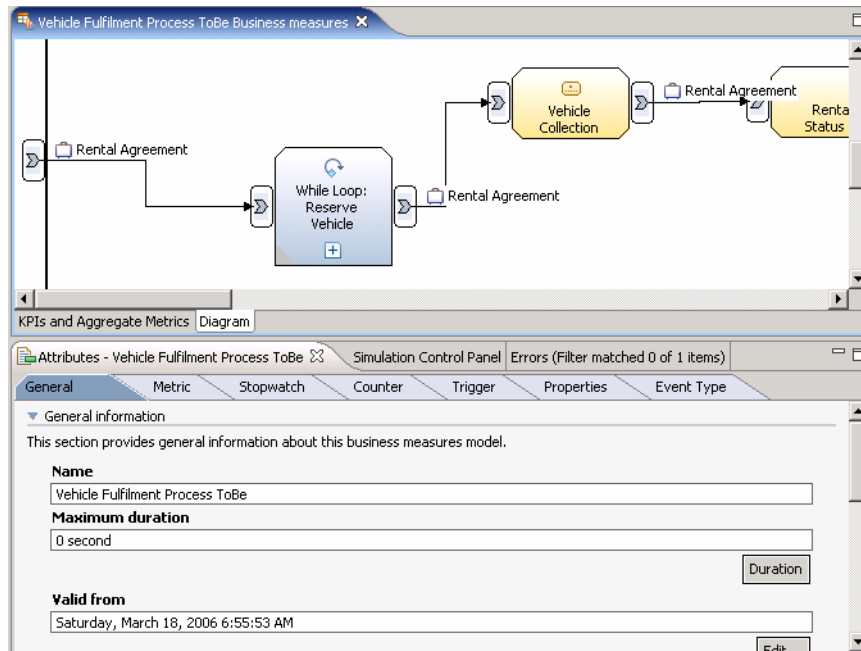


Figure 11-50 Business measures: diagram and attributes

11.7.2 Creating an event for a business situation

Before defining a business situation, first define the event that will be created by WebSphere Business Monitor whenever this business situation is detected. To create a new event, perform the following steps:

1. In the Project Tree view, right-click **Events** and select **New → Event Definition**. The Create a new event definition dialog appears.
2. Set the Name of the new event definition field to **Recurring Date violation** and click **Finish**.
3. The event definition opens in the editor. You use this editor to define event type attributes. Four default event type attributes will be defined already. Click **Add** to add a new attribute.
4. Set the Name of the attribute to **Rental ID**. Leave the Type as **String** (Figure 11-51 on page 240).

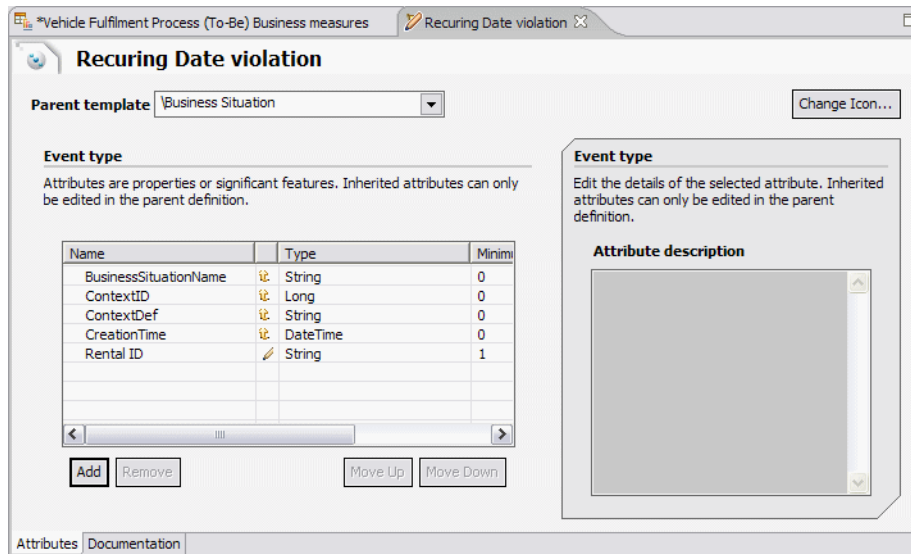


Figure 11-51 Event definition attributes

5. Save and close the event definition editor.

11.7.3 Dimensional analysis

Dimensional analysis organizes historical data into levels of details that you can drill down through to extract significant information. To be able to perform dimensional analysis in WebSphere Business Monitor, you must specify some information in the business measures model for each instance measure (metric, stopwatch, and counter).

A process can be described in terms of *quantitative* data, which takes on many values and participates in calculations, and in terms of *dimensions*, which are entry points for manipulating and analyzing the data in meaningful ways.

Note: *Dimensions* are ways to segment data into groups so that the analysis can be performed on a subset of the historical data.

Typically, any measure with non-numeric values is a level of a dimension, and you analyze other measures against dimensions.

For example, using dimensional analysis can help answer question such as:

- ▶ Do more due-violated contracts come from the USA than from Canada?
- ▶ Are customers returning cars slower in locations at the airports?

- Are rentals for class C cars tending to upgrade to a higher vehicle class?

We can analyze numerical data (counters, stopwatches) against other data that we capture (country, city, customer classification).

Adding a dimension

A dimension can be multi-level, such as a location, which breaks down into country and city.

To add the location as a dimension, perform these steps:

1. Click the **Diagram** tab in the business measures model and click in an empty area to see the attributes of the process in the Attributes view.
2. In the Attributes view, click the **General** tab and click **Add** in the Dimensions section.
3. Rename the dimension to Location (Figure 11-52).

The screenshot shows the 'Attributes - Vehicle Fulfillment Process ToBe' window. The 'General' tab is active. The 'Dimensions' section is expanded, showing a table with one entry, 'Location', which is highlighted with a red circle. The 'Add' button is visible next to the table.

Dimensions
Location

Figure 11-52 Creating a dimension

4. Define two additional dimensions: Location type and Requested vehicle class.

Create a new trigger

Dimensions used in dimensional analysis are defined using metrics. In our scenario, all of the metrics used as dimensions obtain their value from the process input. So create a new trigger for process input to be used in these metrics value assignments. To define a trigger, perform the following steps:

1. In the Attributes view for the process, click the **Trigger** tab.
2. In the Reusable triggers section, click **Add** to add a new trigger.
3. Change the default value in the Name field to Rental started.
4. Click in the Source category field to select **Inputs and outputs**.
5. Click the button in the **Source** field (circled in Figure 11-53).

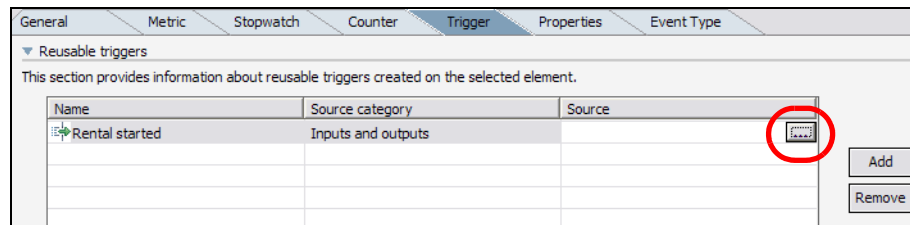


Figure 11-53 Adding new trigger

6. The Select Pin Set dialog appears. Select **Vehicle Fulfilment Process ToBe_Input Criterion** (Figure 11-54) and click **OK**.

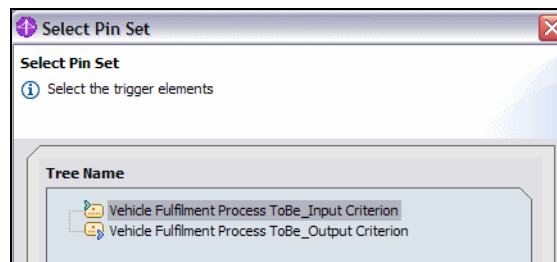


Figure 11-54 Select Pin Set dialog

Add metrics

The structure of the dimension is defined by using some of the business items of the process.

1. In the Attributes view for the process, click the **Metric** tab.
2. Click **Add** in the Metrics section. Name the metric Country, and change the Type to **String**.

3. Repeat this process to add the metrics shown in Table 11-27.

Table 11-27 Metrics

Metrics name	Metrics type
City	String
Location type	String
Requested vehicle class	String
Late return percentage	Double
Class upgrade percentage	Double
Loyalty club membership percentage	Double

4. Select the **Country** metric to highlight it.
5. In the Value section, click **Add** to add a trigger to this metric. Click in the Trigger column and click ... to open the Select Trigger dialog.
6. In the Select Trigger dialog, click **Select existing trigger**, select **Rental started**, and click **OK** (Figure 11-55).

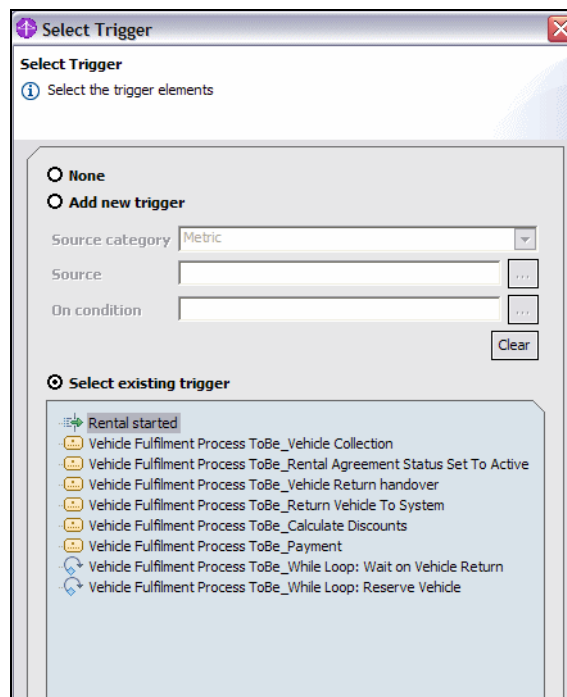


Figure 11-55 Select Trigger dialog

7. Click **Empty Calculation** in the Calculation column and click ... to open the Expression Builder.
8. In the Expression Builder, set the First term to **Modeling artifact**. Expand the modeling artifact **Vehicle Fulfilment Process ToBe** → **Input Criterion** → **Start Vehicle Fulfilment Process Input** → **Branch Details** and select **Country** (Figure 11-56). Click **Apply** and **OK**.

This generates the calculation:

Vehicle Fulfilment Process ToBe.Input Criterion.Start Vehicle Fulfilment Process Input.Branch Details.Country

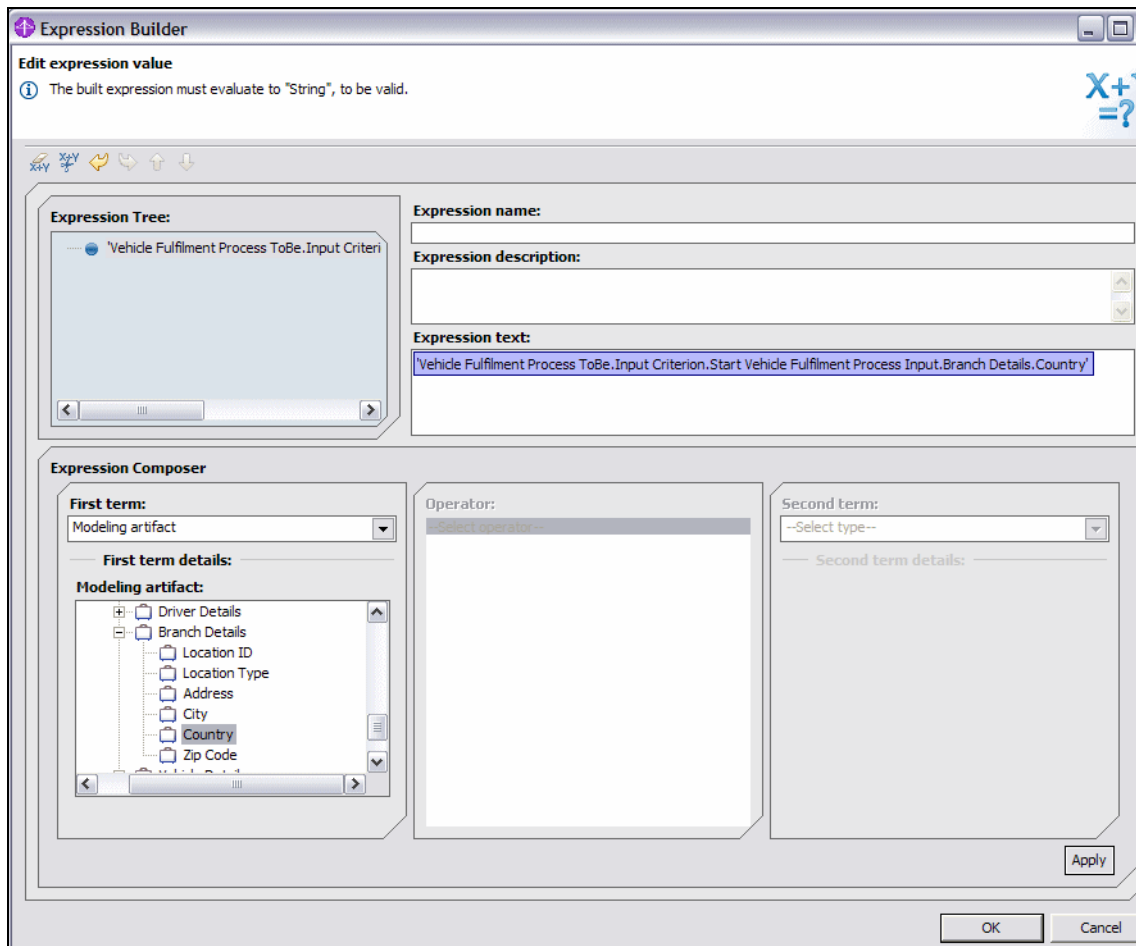


Figure 11-56 Expression Builder

9. Repeat these sets to define additional Rental Started triggers for the metrics in Table 11-28 on page 245.

Table 11-28 Create dimension metrics

Metric	Calculation
City	Vehicle Fulfilment Process ToBe.Input Criterion.Start Vehicle Fulfilment Process Input.Branch Details.City
Location type	Vehicle Fulfilment Process ToBe.Input Criterion.Start Vehicle Fulfilment Process Input.Branch Details.Location Type
Requested vehicle class	Vehicle Fulfilment Process ToBe.Input Criterion.Start Vehicle Fulfilment Process Input.Vehicle Group Requested

Set the quantitative metrics

The remaining metrics (Late return percentage, Class upgrade percentage, and Loyalty club membership percentage) are used as quantitative data in dimensional analysis.

Perform the following steps:

1. Before the start of the Vehicle Collection activity, check the rental agreement to see whether Vehicle Group Received differs from Requested Vehicle Class. If it does differ, this implies that a vehicle upgrade was given, so set the Class upgrade percentage metric for that instance to 100. A value of 0 reflects a rental that did not get upgraded. Implement this as follows:
 - a. In the Attributes view for the process, click **Metric**.
 - b. Select the **Class upgrade percentage** metric, then click **Add** in the Value section.
 - c. Click in the Trigger column and click ... to open the Select Trigger dialog.
 - d. In the Select Trigger dialog, select **Add new trigger**.
 - e. In the Source category, select **Inputs and outputs**.
 - f. Click the button near the Source field to open the Select Pin Set dialog.
 - g. In the Select Pin Set dialog, expand **Vehicle Fulfilment Process ToBe_Vehicle Collection**. Select **Vehicle Fulfilment Process ToBe_Vehicle Collection_Input Criterion** and click **OK**.
 - h. Click the button near the On condition. This opens the Expression Builder.
 - i. In the Expression Builder, set the First Term to **Modeling artifact**. Expand **Vehicle Fulfilment Process ToBe → Vehicle Fulfilment Process ToBe_Vehicle Collection → Input Criterion → Vehicle Collection Input** and select **Vehicle Group Received**.
 - j. In the Operator section, click **is not equal to**.

- k. In the Second term, select **Modeling artifact**. Expand **Vehicle Fulfilment Process ToBe** and select **Requested vehicle class**.
 - l. Click **Apply**, then **OK**.
 - m. Click **OK** in the Select Trigger dialog.
 - n. Click in the Calculation column in the Attributes view and click ... to open the Expression Builder.
 - o. In the First term, select **Number**. Enter 100 as the Number value. Click **Apply** and **OK**.
2. Follow a similar approach to calculate the Loyalty club membership percentage metric. At a high level, perform the following steps, referring to the previous step for detailed instructions.
 - a. Add a new trigger value for **Loyalty club membership percentage**.
 - b. Set the trigger to be a new trigger, using the following criteria:
 - i. Set Source category to **Inputs and outputs**.
 - ii. Set Source to Vehicle Fulfilment Process ToBe_Calculate Discounts_Input Criterion.
 - iii. Set the On condition to an expression that checks whether Vehicle Fulfilment Process ToBe.Vehicle Fulfilment Process ToBe_Calculate Discounts.Input Criterion.Calculate Discount Input.Driver Details.Loyalty Club Number is not equal to an empty String (select Text as the second term and leave the Text value empty).
 - c. Set the calculation to a Number of 100.
3. The Late return percentage metric should be set to the value of 100 if the Vehicle Overdue? element is true. Define this with a reusable trigger called "Vehicle not returned" as follows:
 - a. In the Diagram view expand **While Loop: Wait on Vehicle Return**.
 - b. Click **Update Rental Agreement** to highlight it. In the Attributes view, click the **Trigger** tab and click **Add**.
 - c. Name the trigger Vehicle not returned, ensure that the Source category is set to State change, and set the source to **Activity State Started**.
 - d. In the Diagram view, return to the parent process and click anywhere in the white space to show the properties for the process.
 - e. In the Attributes view for the process, click **Metric**.
 - f. Select the **Late return percentage** metric. Click **Add** in the Value section.
 - g. Click the new trigger to open the Select Trigger dialog. Click **Select existing trigger**, expand **Vehicle Fulfilment Process ToBe_While Loop: Wait on Vehicle Return** → **Vehicle Fulfilment Process ToBe_While**

Loop: Wait on Vehicle Return_Update Rental Agreement, select **Vehicle not returned**, and click **OK** (Figure 11-57).

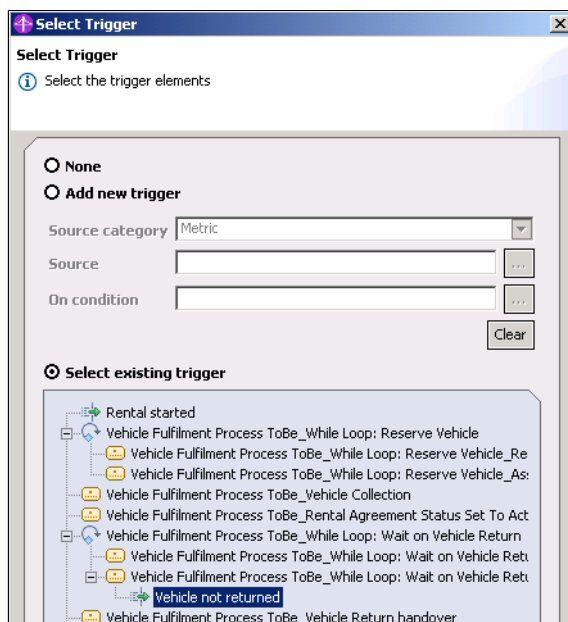


Figure 11-57 Selecting the Vehicle not returned trigger

- h. Open the Expression Builder for the Calculation field. Set the First term field to **Number**, set the Number value to 100, and click **Apply**, then **OK**.
4. Save your changes to the process.

Set the dimensional analysis properties for dimension metrics

1. In the Attributes view for the process, click the **Metric** tab and select the **Country** metric.
2. In the Dimensional analysis and database schema settings section for the Country metric, perform the following steps (Figure 11-58 on page 248):
 - a. Set the Usage in WebSphere Business Monitor field to **Aggregation group in dimensional analysis**.
 - b. Select **Create index to optimize database performance for sorting based on this measure in WebSphere Business Monitor**.
 - c. Set the Maximum string length to **20**.
 - d. Check **Set as part of dimension key**.
 - e. Set Dimension to **Location**.

- f. Set Aggregation group level to **0**.

▼ Dimensional analysis and database schema settings

This section provides details about how the selected business measure will be represented in the dimensional analysis view of WebSphere Busi

Usage in WebSphere Business Monitor

Aggregation group in dimensional analysis ▼

☒ Create index to optimize database performance for sorting based on this measure in WebSphere Business Monitor

Maximum string length

20

▲▼

☐ Allocate additional space in database to accommodate Unicode string for globalization

☒ Set as part of dimension key

Dimension

Location ▼

Aggregation group level

0

▲▼

Figure 11-58 Dimensional analysis and database schema settings

3. Repeat these steps to set the dimensional analysis properties for the City, Location type, and Requested vehicle class metrics, using the values shown in Table 11-29.

Table 11-29 Set the dimensional analysis properties for dimension metrics

Metric	Dimension	Aggregation group level
City	Location	1
Location type	Location type	0
Requested vehicle class	Requested vehicle class	0

Set dimensional analysis properties for quantitative metrics

1. In the Attributes view for the process, click the **Metric** tab and select the **Late return percentage** metric.
2. In the Dimensional analysis and database schema settings section for the Late return percentage metric, perform the following steps as shown in Figure 11-59 on page 249:
 - a. Set the Usage in WebSphere Business Monitor field to **Quantitative data in dimensional analysis**.
 - b. Click **Add** to add a new aggregation measure. Set the Aggregation measure name to Average late return percentage. Ensure that the Aggregation function is set to **Average**.

Dimensional analysis and database schema settings

This section provides details about how the selected business measure will be represented in the dimensional analysis view of WebSphere Business Monitor

Usage in WebSphere Business Monitor: Quantitative data in dimensional analysis

☐ Create index to optimize database performance for sorting based on this measure in WebSphere Business Monitor

Maximum string length: 0

☐ Allocate additional space in database to accommodate Unicode string for globalization

☐ Set as part of dimension key

Dimension:

Aggregation group level: 0

Aggregation measure name	Aggregation function
Average late return percentage	Average

Figure 11-59 Dimensional analysis and database schema settings

- Repeat these steps to set the dimensional analysis properties for the Class upgrade percentage and Loyalty club membership percentage metrics, using the values shown in Table 11-30.

Table 11-30 Set the dimensional analysis properties for dimension metrics

Metric	Aggregation measure name	Aggregation function
Class upgrade percentage	Average class upgrade percentage	Average
Loyalty club membership percentage	Average loyalty club membership percentage	Average

11.7.4 Stopwatches

Implement the following stopwatches:

- ▶ Vehicle return duration

Watches the handover time that a customer needs to return a rental car. Handover starts when the Check if Vehicle Returned activity begins (if the vehicle has been returned), and ends when the Vehicle Return handover activity completes.
- ▶ Vehicle collection duration

Watches how long the Vehicle Collection activity takes to complete.
- ▶ Update agreement duration

Watches how long the Update Rental Agreement activity takes to complete.

Creating the stopwatches

To create the Vehicle return duration stopwatch, follow these steps:

1. In the Diagram view, click in the white space to show the attributes for the process in the Attributes view, then click the **Stopwatch** tab.
2. In the Stopwatches section, click **Add** to define a new stopwatch. Name the stopwatch `Vehicle return duration`.
3. With this stopwatch highlighted, click **Add** in the Triggered actions section. Define the trigger to start the stopwatch when the Check If Vehicle Returned activity starts.
 - a. Click ... in the Trigger column of the new trigger to open the Select Trigger dialog.
 - b. In the Select Trigger dialog, select **Add new trigger**. Define the new trigger as follows:
 - i. Set Source category to **State change**.
 - ii. In the Source field click ... to open the Select State Transition Event dialog. Expand **Vehicle Fulfilment Process ToBe_While Loop Wait on Vehicle Return → Vehicle Fulfilment Process ToBe_While Loop Wait on Vehicle Return_Check If Vehicle Returned**, select **Activity State Started**, and click **OK**.
 - c. Click **OK** in the Select Trigger dialog to add the trigger.
 - d. Set the Resulting action column to **Start**.
4. Add a second trigger for the Vehicle return duration stopwatch to reset the stopwatch if the vehicle is not returned:
 - a. Click **Add**.
 - b. Click ... in the Trigger column of the new trigger to open the Select Trigger dialog.
 - c. In the Select Trigger dialog, choose **Select existing trigger** and expand **Vehicle Fulfilment Process ToBe_While Loop: Wait on Vehicle Return → Vehicle Fulfilment Process ToBe_While Loop: Wait on Vehicle Return_Update Rental Agreement**, select **Vehicle not returned**, and click **OK**.
 - d. Set the Resulting action column to **Reset**.
5. Add a third trigger for the Vehicle return duration stopwatch to stop the stopwatch when the Vehicle Return handover activity completes.
 - a. Click **Add**.
 - b. Click ... in the Trigger column of the new trigger to open the Select Trigger dialog.

- c. In the Select Trigger dialog, select **Add new trigger**. Define the new trigger as follows:
 - i. Set Source category to **State change**.
 - ii. In the Source field click ... to open the Select State Transition Event dialog. Expand **Vehicle Fulfilment Process ToBe_Vehicle Return handover**, select **Activity State Completed**, and click **OK**.
 - d. Click **OK** in the Select Trigger dialog to add the trigger.
 - e. Set the Resulting action column to **Stop**.
6. Figure 11-60 shows the completed Vehicle return duration stopwatch.

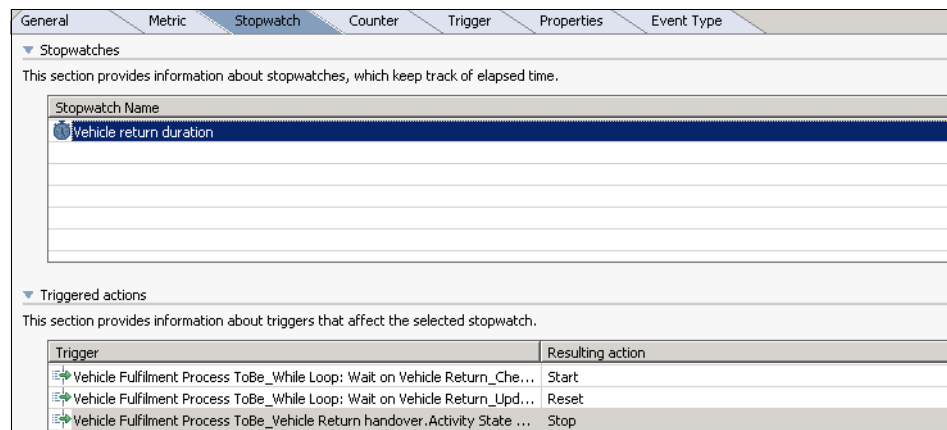


Figure 11-60 Vehicle return duration

7. Create the other two stopwatches:
- a. Create a stopwatch named **Vehicle collection duration** and assign a new state change trigger to it using values shown in Table 11-31.

Table 11-31 Vehicle collection duration trigger settings

Trigger	Resulting action
Vehicle Fulfilment Process ToBe_Vehicle Collection.Activity State Started	Start
Vehicle Fulfilment Process ToBe_Vehicle Collection.Activity State Completed	Stop

- b. Create a stopwatch named **Update agreement duration** and assign a new state change trigger to it using values shown in Table 11-32 on page 252.

Table 11-32 Update agreement duration trigger settings

Trigger	Resulting action
Vehicle Fulfilment Process ToBe_While Loop: Wait on Vehicle Return_Update Rental Agreement.Activity State Started	Start
Vehicle Fulfilment Process ToBe_Wait on Vehicle Return_Update Rental Agreement.Activity State Completed	Stop

8. Save the changes.

Setting the dimensional analysis properties

Define dimensional analysis properties for each of the stopwatches:

1. In the Stopwatch tab, select the **Vehicle return duration** stopwatch.
2. In the Dimensional analysis and database schema settings section:
 - a. Set Usage in WebSphere Business Monitor to **Quantitative data in dimensional analysis**.
 - b. Click **Add** to add a new aggregation measure. Set the name to Average vehicle return duration and set the Aggregation function to **Average**.
3. Repeat these steps for the other two stopwatches, using the settings in Table 11-33 to determine the aggregation measure settings.

Table 11-33 Dimensional analysis properties

Stopwatch	Aggregation measure name	Aggregation function
Vehicle collection duration	Average vehicle collection duration	Average
Update agreement duration	Average update agreement duration	Average

4. Save the changes.

11.7.5 Counters

Counters are specialized business measures elements that keep track of the number of occurrences of a situation or event. We implement a counter called Number of times overdue, which increments by one every time a customer violates the rental agreement by not returning the vehicle on time.

Defining this counter involves the following steps:

- ▶ Define a new metric.
- ▶ Define the counter.

- ▶ Define a situation event for the counter.
- ▶ Set the dimensional analysis properties for the counter.

Defining a new metric

Our counter requires a Reservation ID metric. To define this metric:

1. In the Diagram view, click in the white space to show the attributes for the process in the Attributes view, then click the **Metric** tab.
2. In the Metrics section, click **Add**. Name the metric **Reservation ID** and set the Type to **String**.
3. With the Reservation ID metric highlighted, click **Add** in the Value section.
4. Click ... in the Trigger column to open the Select Trigger dialog, and perform the following steps:
 - a. Select **Add new trigger**.
 - b. Set Source category to **Inputs and outputs**.
 - c. Click ... next to the Source field to open the Select Pin Set dialog. Expand **Vehicle Fulfilment Process ToBe_While Loop: Reserve Vehicle** → **Vehicle Fulfilment Process ToBe_While Loop: Reserve Vehicle_Reserve Vehicle In System**, select **Vehicle Fulfilment Process ToBe_While Loop: Reserve Vehicle_Reserve Vehicle In System_Output Criterion**, and click **OK**.
 - d. Click **OK**.
5. Click ... in the Calculation column to open the Expression Builder. Follow these steps:
 - a. Set the First term to **Modeling artifact**.
 - b. In the First term details, expand **Vehicle Fulfilment Process ToBe** → **Vehicle Fulfilment Process ToBe_While Loop: Reserve Vehicle** → **Vehicle Fulfilment Process ToBe_While Loop: Reserve Vehicle_Reserve Vehicle In System** → **Output Criterion** → **Reserve Vehicle In System Output** and select **Reservation ID**.
 - c. Click **Apply**, then **OK**.
6. In the Dimensional analysis and database schema settings section, set Usage in WebSphere Business Monitor to **Active data about running process**.

Creating a counter

To define a counter:

1. In the Diagram view, click in the white space to show the attributes for the process in the Attributes view, then click the **Counter** tab.

2. In the Counters section, click **Add**. Name the counter Number of times overdue.
3. With this counter selected, click **Add** in the Triggered actions section. Define the trigger as follows:
 - a. Click ... in the Trigger column of the new trigger to open the Select Trigger dialog.
 - b. In the Select Trigger dialog, choose **Select existing trigger**.
 - c. Expand **Vehicle Fulfilment Process ToBe_While Loop: Wait on Vehicle Return → Vehicle Fulfilment Process ToBe_While Loop: Wait on Vehicle Return_Update Rental Agreement**. Choose **Vehicle not returned**.
 - d. Click **OK**.
4. Set the Resulting action field for the trigger to **Add one**.

Adding a situation event

At monitoring time, if the Number of times overdue counter reaches a count of 3 or more, WebSphere Business Monitor should raise a situation event. Create a situation event called Recurring date violation as follows:

1. In the Counters tab, highlight the Number of times overdue counter. In the Situation events section, click **Add**. The Event dialog opens.
2. Select **Recurring Date violation**, and click **OK** (Figure 11-61).

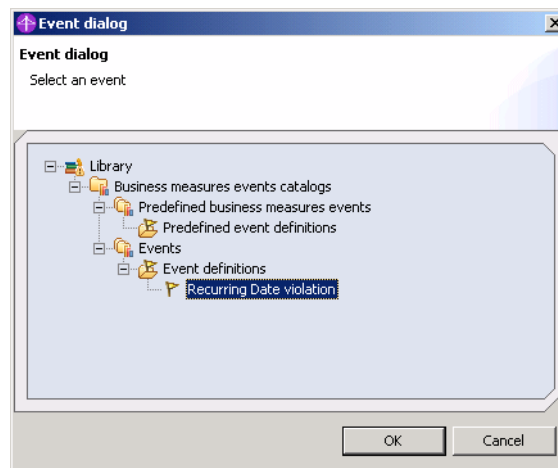


Figure 11-61 Event dialog

3. Expand **Recurring Date violation** to display the situation event properties. Double-click in the Attribute calculation name field for the Business SituationName attribute and click ... to open the Expression Builder.

4. In the Expression Builder, set the First term to **Text**. Set the Text value to MaxOverdueCountExceeded and click **Apply**, then **OK**.
5. Double-click in the Attribute calculation name field for the Rental ID attribute and click ... to open the Expression Builder.
6. In the Expression Builder, set First term to **Modeling artifact**, expand Vehicle Fulfilment Process ToBe, and select the **Reservation ID** metric (Figure 11-62). Click **Apply**, then **OK**.

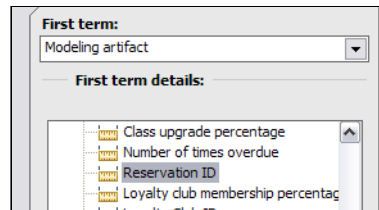


Figure 11-62 Expression Builder

7. Highlight the **Recurring Date violation** situation event, and click **Edit** next to the On Condition field. This opens the Expression Builder. Define the following expression:
 - a. Set First term to **Modeling artifact**.
 - b. In First term details, expand **Vehicle Fulfilment Process ToBe** and select **Number of times overdue**.
 - c. Set the Operator to **is greater than or equal to**.
 - d. Set Second term to **Number**.
 - e. In Number value, enter 3.
 - f. Click **Apply**, then **OK**.
8. The situation events section should look similar to Figure 11-63.

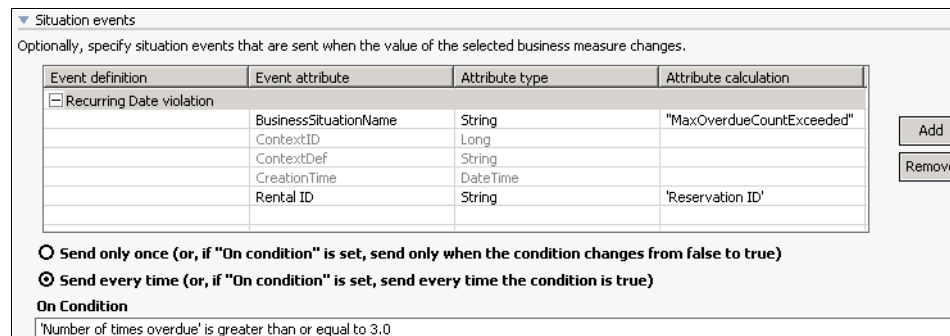


Figure 11-63 Situation events

Setting the dimensional analysis properties

Define dimensional analysis properties for the counter as follows:

1. Select the **Number of times overdue** counter.
2. In the Dimensional analysis and database schema settings section:
 - a. Set Usage in WebSphere Business Monitor to **Quantitative data in dimensional analysis**.
 - b. Click **Add** to add a new aggregation measure. Set Aggregation measure name to Average number of times overdue and Aggregation function to **Average**.
3. Save the changes.

11.7.6 Key performance indicators (KPIs)

Now define the following KPIs:

- ▶ Percentage of rentals with late returns
How often customers violate their contract due dates.
- ▶ Average number of late returns per rental
The average number of times a customer violates contract by returning the car late.
- ▶ Average handover time
The average amount of time a customer takes to hand over the car. This measures the efficiency of customer service.

To create these KPIs, follow these steps:

1. In the main editor, click the **KPIs and Aggregate Metrics** tab.
2. Click the **KPIs** link on the left side of the editor, then click **Add** in the KPIs section to add a new KPI.
3. Set the following properties for the KPI:
 - a. Set the Name field to Percentage of rentals with late returns.
 - b. Set the Type to **Double**.
 - c. Set Aggregation Function to **Average**.
 - d. Click in the Aggregation source field. Click ... to open the Source Business Measure selection dialog. Select **Late return percentage** and click **OK**.
 - e. Uncheck the Use Target field.
 - f. Click in the Lower limit field. Click ... to open the Default value dialog. Click **Literal value**, enter 0, and click **OK** (Figure 11-64 on page 257).

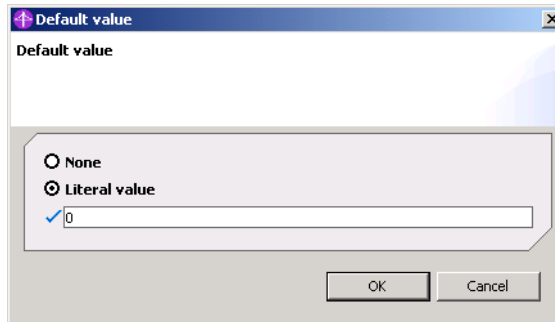


Figure 11-64 Default value dialog

- g. Click in the Upper limit field. Click ... to open the Default value dialog. Click **Literal value**, enter 25, and click **OK**. The Percentage of rentals with late returns is now defined (Figure 11-65).

KPIs

This section provides information about key performance indicators (KPIs) calculated across multiple runs of the process.

Name	Type	Aggregation function	Aggregation source	Use Target	Lower limit	Upper limit
Percentage of rentals with late returns	Double	Average	Late return percentage	<input type="checkbox"/>	0.0	25.0

Add Remove

Figure 11-65 Percentage of late returns KPI

4. Repeat these steps to define a second KPI called Average number of late returns per rental. Use the following attributes:
 - Name: Average number of late returns per rental
 - Type: **Double**
 - Aggregation function: **Average**
 - Aggregation source: **Number of times overdue**
 - Use Target: Uncheck
 - Lower limit: 0
 - Upper limit: 0.5
5. Repeat these steps to define a third KPI called Average handover time. Use the following attributes:
 - Name: Average handover time
 - Type: **Duration**
 - Aggregation function: **Average**
 - Aggregation source: **Vehicle return duration**

6. Figure 11-66 shows the completed KPIs. Save the changes.

Figure 11-66 All KPIs

Aggregate metrics provide calculations across multiple executions of the process for finding average, maximum, minimum, or total number of occurrences.

1. In the main editor, click the **KPIs and Aggregate Metrics** tab.
2. Click the **Metrics** link on the left side of the editor, then click **Add** in the Metrics section to add a new aggregate metric.
3. Set the following properties for the aggregate metric:
 - a. Set the name to Highest number of date violations.
 - b. Set Type to **Integer**.
 - c. Set Aggregation function to **Maximum**.
 - d. Set Aggregation source to **Number of times overdue**.
4. Figure 11-67 shows the completed aggregate metric. Save your changes.

Figure 11-67 Aggregate metric

11.7.8 Synchronizing the business measures model

The definition of the business measures model is complete. Before you can export the process to WebSphere Integration Developer and the business measures model to WebSphere Business Monitor, synchronize the business measures model with the process model:

1. In the Project Tree view, right-click the process model **Vehicle Fulfilment Process ToBe** and select **Synchronize**.
2. This opens the Synchronize business measures dialog. In the section Business measures for these process elements will be synchronized, click **Vehicle Fulfilment Process ToBe**, then click **OK** (Figure 11-68).

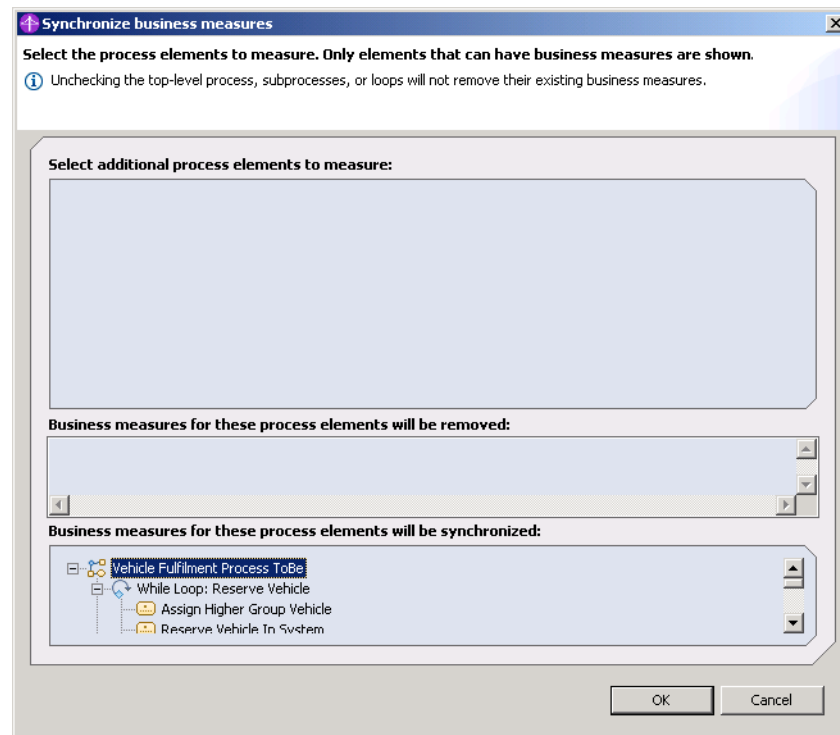


Figure 11-68 Synchronize business measures dialog

The process now is ready for export.

11.8 Exporting the To-Be model to WebSphere Integration Developer and WebSphere Business Monitor

With the To-Be model and the business measures model complete, generate and export model artifacts for handover to IT development.

When the project containing the business measures model is exported, the process model is automatically exported as well.

Note: The WebSphere Business Modeler Export wizard offers the following two options to export and generate the BPEL artifacts:

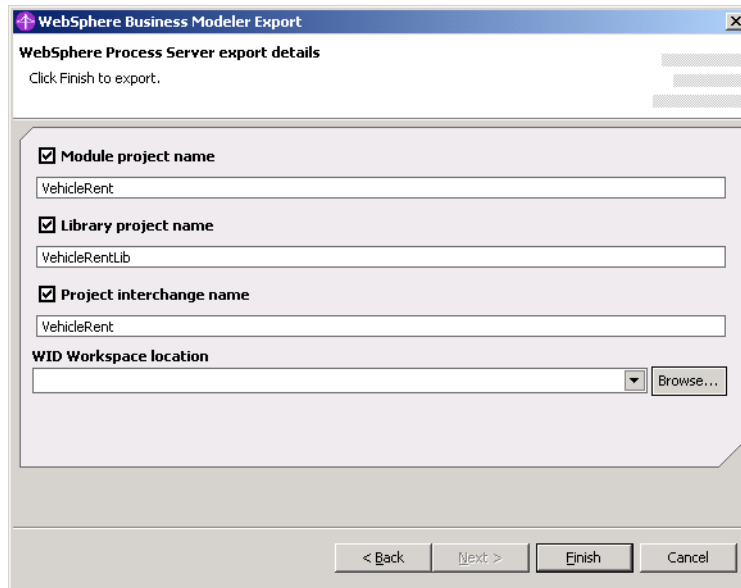
- ▶ WebSphere Business Monitor and development tools
- ▶ WebSphere Process Server

However, if you want to monitor the process in WebSphere Business Monitor, you have to use the **WebSphere Business Monitor and development tools** option to keep the business measure model in sync with your WebSphere Process Server implementation.

To export the model, perform the following steps:

1. Check the process model for validation errors and make sure that only warnings (and no errors) are indicated.
2. In the project tree, right-click and select **Export**. The Export wizard opens.
3. Select **WebSphere Business Monitor and development tool**. Click **Next**.
4. In the Destination and source screen, perform the following steps:
 - a. Click **Browse** to specify the directory where you want to put the files.
 - b. Select **ITSO Speedy Rentals To-Be Project** from the Project list.
 - c. Optionally, specify a number for the project version. This number is used by WebSphere Business Monitor so that you can have incremental versions of the same project. We take the default value here.
 - d. Check **Export user defined event types to event catalog** to export the event definitions we have created in the project tree. Only event definitions that are referenced from business measures in this project are exported.
 - e. Click **Overwrite files** to allow overwrite of any previous versions of files that you might have exported. Click **Next**.

5. In the WebSphere Process Server export details screen (Figure 11-69), follow these steps:
 - a. Specify the WebSphere Integration Developer module by checking **Module project name** and entering VehicleRent as the module name.
 - b. Specify the WebSphere Integration Developer library by checking **Library project name** and entering VehicleRentLib as the name of the library.
 - c. Specify the Project interface name: Check **Project interchange name** and enter VehicleRent.
 - d. We do not use the option to browse to the workspace that is used by WebSphere Integration Developer.
 - e. Click **Finish**. A window opens when the export process is complete.



The image shows a screenshot of the 'WebSphere Business Modeler Export' dialog box. The title bar reads 'WebSphere Business Modeler Export'. The main area is titled 'WebSphere Process Server export details' and contains the instruction 'Click Finish to export.' Below this, there are three checked checkboxes: 'Module project name', 'Library project name', and 'Project interchange name'. Each checkbox has a corresponding text input field. The 'Module project name' field contains 'VehicleRent'. The 'Library project name' field contains 'VehicleRentLib'. The 'Project interchange name' field contains 'VehicleRent'. Below these fields is a section labeled 'WID Workspace location' with a text input field and a 'Browse...' button. At the bottom of the dialog, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Figure 11-69 WebSphere Process Server export details

6. If there were any errors or warnings during the export process, click **Details** to read them. Otherwise, click **OK**.
7. The WebSphere Business Monitor and development tools wizard generates the following files for you:
 - Monitor.zip file
 - Contains the business measure model for WebSphere Business Monitor.

- WebSphere Process Server\VehicleRental.zip file
The Project Interchange File to import into WebSphere Integration Developer.
- EventCatalog\EventCatalog.xml
Contains the situation event definitions.



Assemble with WebSphere Integration Developer

This chapter describes how the role of the integration developer can take resources exported from WebSphere Business Modeler and use them to create and test a WS-BPEL business process implementation.

The integration developer undertakes the following steps:

- ▶ Import generated artifacts from WebSphere Business Modeler that have been created by the business analyst
- ▶ Import service artifacts from the service developer
- ▶ Customize business process logic
- ▶ Implement several component types that will comprise our application, including business rules and human tasks
- ▶ Create maps to translate between process interfaces and datatypes, and service interfaces and datatypes
- ▶ Assemble the components into the composite application
- ▶ Configure the test server and unit-test the process
- ▶ Export application artifacts for deployment onto WebSphere Process Server

This chapter assumes that you have WebSphere Integration Developer V6.0.1 installed.

12.1 Import necessary artifacts

This section describes how to import the artifacts you need to assemble the WS-BPEL business process:

- ▶ Import artifacts from the To-Be process built in WebSphere Business Modeler
- ▶ Import pre-built Web service artifacts that will be used by the WS-BPEL business process.

12.1.1 Import artifacts from WebSphere Business Modeler

This section imports programming artifacts that have been generated by the business analyst and enterprise architect with WebSphere Business Modeler. These artifacts will form the abstract basis for the composite application that we build in this chapter.

We assume that WebSphere Integration Developer has been installed already along with the WebSphere Process Server unit test environment (UTE) feature.

Perform the following steps:

1. Open a new workspace in WebSphere Integration Developer V6.0.1.
2. Import the WebSphere Business Modeler artifacts:
 - a. Click **File** → **Import** → **Project Interchange** and click **Next**.
 - b. Click **Browse** next to the From zip file field and locate **VehicleRent.zip**. If you have completed 12.3.4, “Export deployable artifacts for WebSphere Process Server” on page 331, this file will be in the \WebSphere Process Server directory where you exported your WebSphere Business Modeler artifacts. If you have not completed this section, you can import VehicleRent.zip from the \Modeler\Export\WebSphere Process Server directory in the additional material that is supplied with this book.
 - c. Select both the **VehicleRent** and **VehicleRentLib** projects (Figure 12-1 on page 265), then click **Finish**.

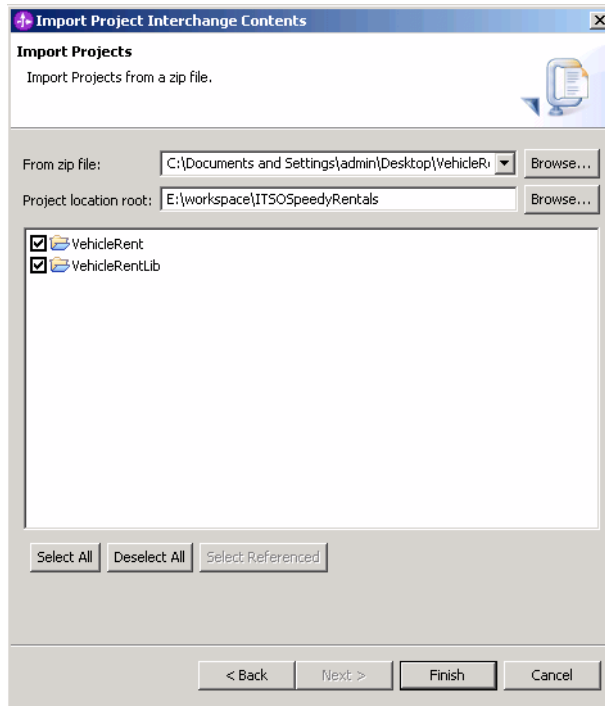


Figure 12-1 Selecting projects to import

3. Make sure you are in the Business Integration perspective, which shows the projects in Figure 12-2 on the left side of your workspace. Allow the workspace build to complete before proceeding. An indicator at the bottom right indicates progress of this task.

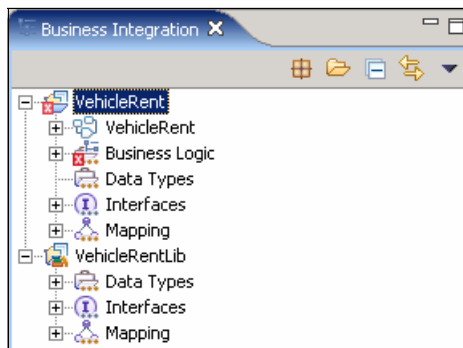


Figure 12-2 Business integration perspective

These artifacts, which form the basis of the composite application that we build in this chapter, are organized into the following projects:

- ▶ A module named VehicleRent
Contains the assembly diagram, WS-BPEL process, business rule groups, human tasks, and interfaces.
- ▶ A library named VehicleRentLib
Contains an additional interface plus the data type definitions for our business objects.

The VehicleRent module has a dependency on the VehicleRentLib library, because this library contains common elements that VehicleRent requires.

Note: At this point a red “X” appears on the Business Logic folder under VehicleRent, indicating a problem: WebSphere Integration Developer has validated the project and found no business rule implementations for your business rule components. We take care of that shortly.

4. Open the process model to see what has been exported from WebSphere Business Modeler, and to see what services the process will need:
 - a. In the Business Integration view, expand **Vehicle Rent** → **Business Logic** → **Processes**. Double-click **VehicleFulfillmentProcessToBe**.
 - b. WebSphere Integration Developer opens the process. Figure 12-3 on page 267 shows the process in the process editor.

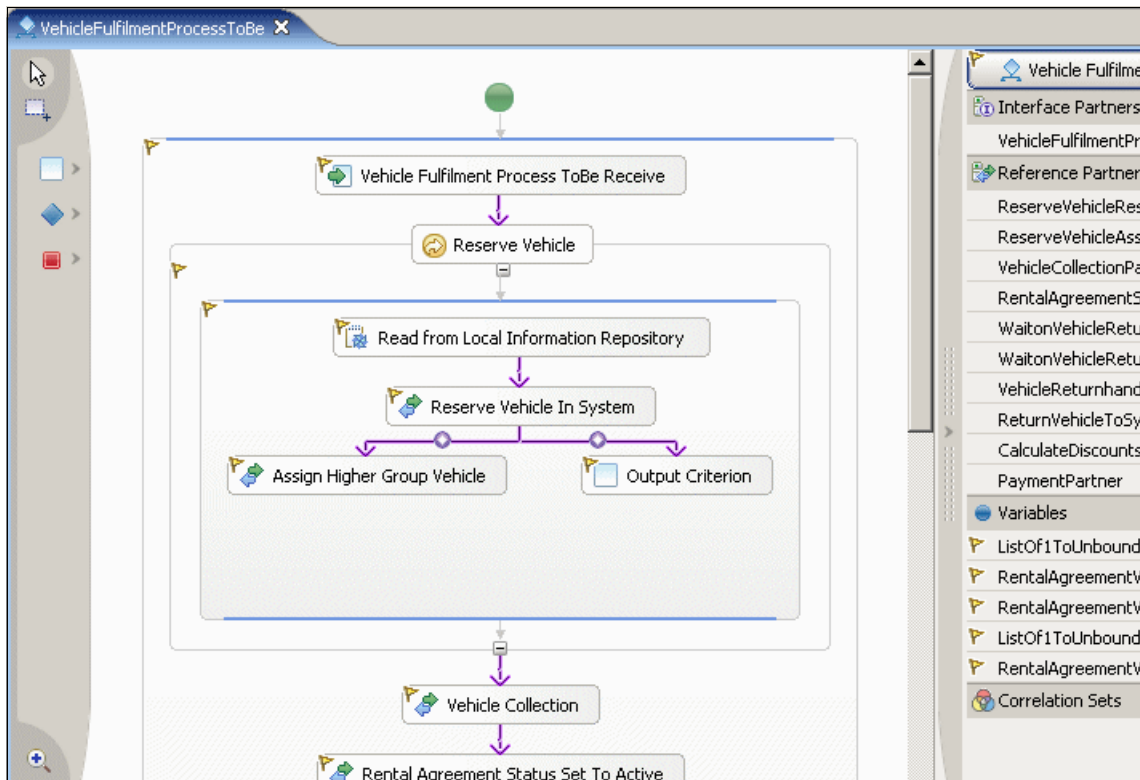


Figure 12-3 Opening VehicleFulfilmentProcessToBe process

The process is logically complete in terms of process logic, services, and data types. This highlights that WebSphere Business Modeler excels at business process modeling for analysis and redesign, as well as at creating a high-fidelity process model for execution.

However, the process at this point is somewhat abstract. For example, each *Invoke* element on the process diagram has a corresponding *Reference Partner* in WS-BPEL terms. So far this tells the process what interface type to invoke, but lacks the binding information to tell the process how to invoke our real services.

At this point, in addition to the process model, the VehicleRent module and VehicleRentLib contain several other artifacts, including the service interfaces and the data types for our business objects. There are also business logic components in the Business Logic folder of the VehicleRent module that are typed (such as Java, human task, business rule, and so forth) but have empty implementations. These will be implemented later in the chapter.

5. Close the process editor for now. The next steps import the service artifacts that will provide services to our vehicle rental business process.

12.1.2 Import service artifacts

Our business process requires importing several existing services so we can invoke them in the business process. One of our services was developed in J2EE technology by a Java developer. Most often, this service would be developed using a development environment such as Rational Application Developer or Rational Software Architect. This J2EE-based service, called *PoolManager*, deals with transacting vehicle reservation information to a persistent database. It contains the following three functions (or operations):

- ▶ Reserving a vehicle for a customer (reserveVehicle)
- ▶ Releasing a vehicle to the customer once rented (releaseVehicle)
- ▶ Returning a vehicle to the vehicle pool after the customer has turned it in (returnVehicle)

The database behind the PoolManager service is a Cloudscape database that contains two tables, VEHICLE (which stores vehicle types) and POOL (which stores the status of our vehicle inventory). This file-based database is included in the sample code that is supplied with this book.

Two other prebuilt artifacts, which we will use later, are each a type of Payment service. One is based on the use of a standard JCA file system adapter, and the other is based on a simple POJO (Plain Old Java Object) Web service; they fundamentally perform the same service but are implemented as two different types of components.

These two implementations of the Payment service are imported to show later how to use a Selector component to invoke components dynamically at runtime based on a date/time configuration. This is useful if you have different service implementations that have to run dynamically based on time of day or other custom criteria. This is the reason that Selector components are valuable: They make your implementation much more flexible to change, truly service-oriented, and dynamic.

To import the service artifacts for the J2EE Web service:

1. Import the PoolManager Web service artifacts into the workspace.
 - a. In WebSphere Integration Developer, switch to the J2EE perspective. Click **Window** → **Open Perspective** → **Other**. In the Select Perspective window, click **Show all**, select **J2EE**, and click **OK**.
 - b. From the menu, click **File** → **Import** → **EAR File** and click **Next**.

- c. Browse to the location of ITSOSpeedyServices.ear (Figure 12-4), which you can find in the \WID\Services directory of the additional material supplied with this book. Click **Finish**.

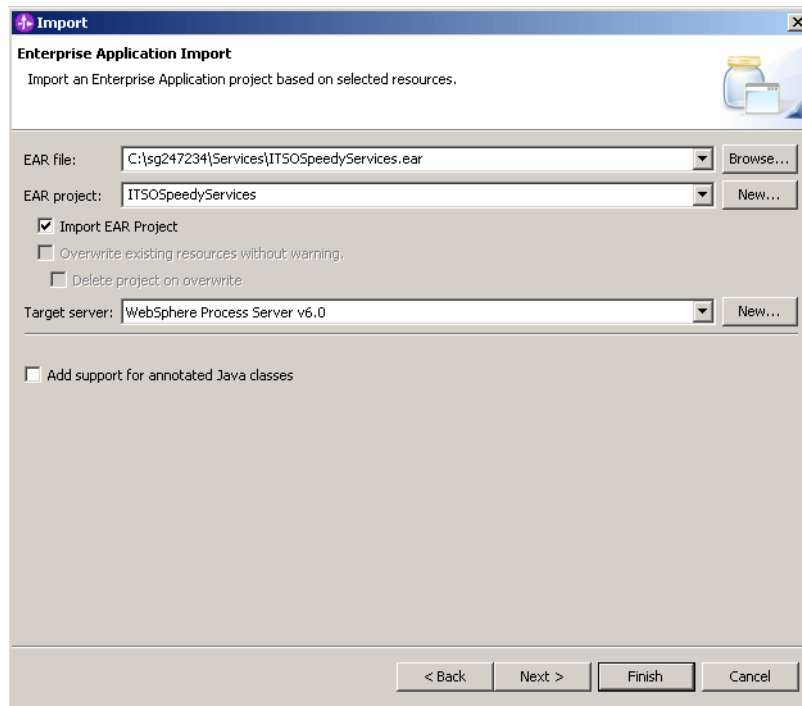


Figure 12-4 Importing J2EE EAR for PoolManager service

Note: You might wonder why we are importing the code for this Web service. Usually you would not need to import the actual code for a Web service; instead you would just use its WSDL file to find out how to invoke it. However, in this scenario we will deploy the composite (process) application and the J2EE Web services application onto the same server for testing and deployment. Therefore, keep in mind that this part of the scenario is not real-world, but it helps make our sample more convenient and instructive because we can test all on one server.

2. Make the PoolManager Web service elements available from our shared library, VehicleRentLib:
 - a. Copy the WSDL interface for the PoolManager Web service that we just imported, into our library. In the Project Explorer view, expand **EJB Projects** → **ITSOSpeedyServicesEJB** → **ejbModule** → **META-INF** →

wsdl. The wsdl folder contains PoolManager.wsdl, as shown in Figure 12-5. Right-click **PoolManager.wsdl** and click **Copy**.

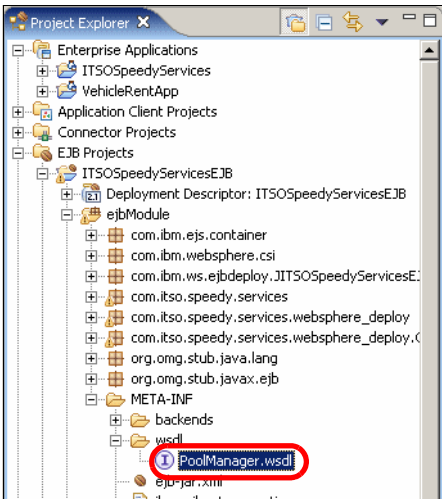


Figure 12-5 Copying WSDL interface for PoolManager Web service

- b. Switch to the Business Integration view. Expand the **VehicleRentLib** library, and expand **Interfaces**. Right-click **Interfaces** and click **Paste**. You should see the PoolManager interface appear, along with new data types and a Web Service Port, as shown in Figure 12-6.

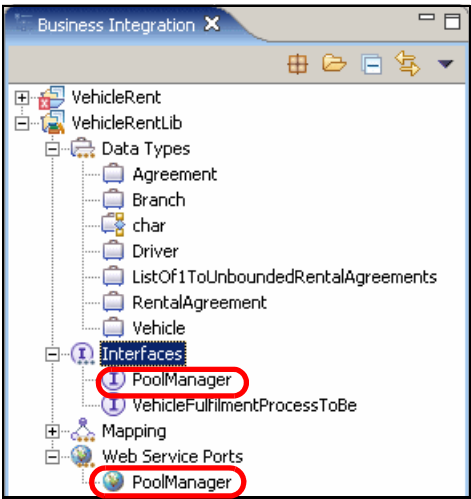


Figure 12-6 New artifacts appear under VehicleRentLib library after copying WSDL

3. Create a separate module for importing this Web service and for making it available to the main process module.
 - a. Right-click the white space in the Business Integration view and click **New** → **Other**. In the New wizard, click **Module** and click **Next**. Type **ReserveSvc**s as the name of the new module. Click **Finish**.

Note: In WebSphere Integration Developer, the primary top-level structural artifacts are modules and libraries (and mediation modules when working with WebSphere Enterprise Service Bus). Modules are fairly analogous to projects and have dependencies to libraries for elements that are shared across modules (such as common datatypes, interfaces, or data maps). Modules can also reference each other, which is useful.

We are separating our PoolManager Web service into its own module so that it can be managed and maintained separately from the process that uses it (in our case the VehicleRent module). We will place other services and component types into their own modules later in the chapter to enable us to manage the VehicleRent process module separate from the services and components it depends on, thus making changes to either the process or the services independent of each other. This separation of concerns will significantly enhance maintainability of our service-oriented architecture as it proceeds through its lifecycle.

- b. Double-click the **ReserveSvc**s module you just created to display the Module Dependencies editor. Click **Add**, select **VehicleRentLib**, and click **OK** to add a dependency on the VehicleRentLib library to the ReserveSvc's module. Press Ctrl+S to save this change and close the Module Dependencies editor.
 - c. In the Business Integration view, expand the **ReserveSvc**s module and double-click the **ReserveSvc**s assembly diagram (Figure 12-7). The assembly diagram opens in the assembly diagram editor.

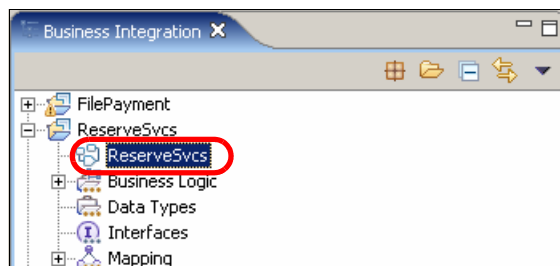


Figure 12-7 Selecting ReserveSvc's assembly diagram

- d. In the Business Integration view, expand **VehicleRentLib**, then **Web Service Ports**. Click **PoolManager**, then drag it to the assembly diagram and drop it onto the free-form surface. In the Component Creation dialog, click **Import with Web Service Binding** and click **OK** (Figure 12-8).

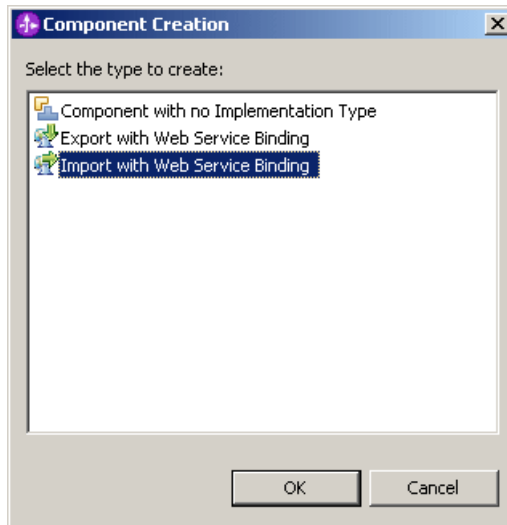


Figure 12-8 Importing PoolManager service with Web Service Binding

- e. We have now created a Web service import (default name: Import1). Right-click this **Import1** element in the assembly diagram, select **Rename**, and rename it to ImportReserveSvc.

- f. Make the services of this module available to other modules by adding an Export. From the palette on the left side of the assembly diagram, click the **Import** drawer arrow, which shows three elements, and click the **Export** element as shown in Figure 12-9. Add the Export element to the assembly diagram canvas.

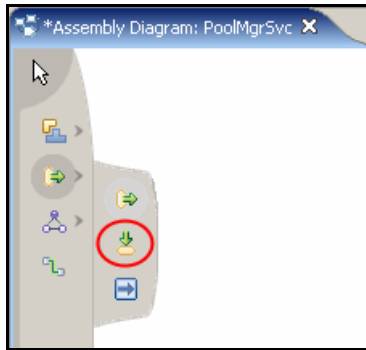


Figure 12-9 Adding an Export element to the PoolMgrSvc assembly diagram

- g. An element named Export1 will now show on the free-form surface. Select this **Export1** element, then hover the mouse over the right border of this element until a connector appears. Select this connector and connect it to the Interface icon on the left border of the ImportReserveSvc element. In the Add wire dialog select the check box for **Always create without prompt** and click **OK**.

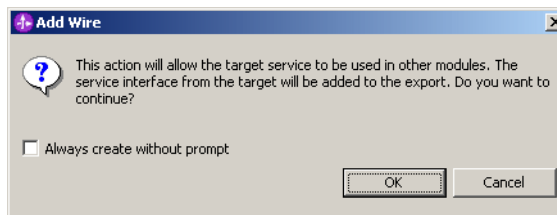


Figure 12-10 Add wire between export and import elements

- h. The assembly diagram now shows the Export1 element connected to the ImportReserveSvc Import element, and an Interface element now appears on Export1 (Figure 12-11). Save your work, then close the assembly diagram editor.

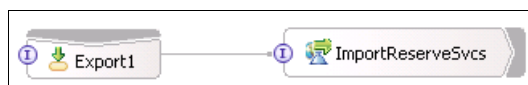


Figure 12-11 ReserveSvc assembly diagram

4. Import File Adapter and POJO service artifacts
 - a. In the Business Integration view, expand **VehicleRent** and **Interfaces**. Right-click **Payment**, and choose **Move**. A dialog appears asking where to move the Interface. Choose the **VehicleRentLib** library and click **OK**. The artifacts we will import depend on this interface, and it has to be in a shared artifact, thus we are moving it to the VehicleRentLib library that all of our modules depend on.
 - b. Click **File** → **Import** → **Project Interchange** and import PaymentSvcs.zip from the \WID\Services directory of the additional material supplied with this book. Choose to **Select All** projects and click **Finish**. Allow WebSphere Integration Developer to finish rebuilding the workspace.

Work is complete in this section. To summarize:

- ▶ Copied the EAR file for the PoolManager Web service that we will deploy along with our composite application.
- ▶ Copied the WSDL file for PoolManager to our shared library VehicleRentLib so that its interface can be shared among multiple modules.
- ▶ Created a separate module called ReserveSvcs for the Web service import.
- ▶ In the assembly diagram for ReserveSvcs, added an Import (with Web Services Binding) to call the Web service, and an Export to make the ReserveSvcs module callable by other modules.
- ▶ Imported the finished artifacts for both the file adapter-based and POJO Web service-based implementations of the Payment Web service.

12.2 Integrate and assemble the composite application

This section creates some of the other important artifacts that will make up the composite application. Tasks include:

- ▶ Organize artifacts for reuse between modules.
- ▶ Customize the VehicleRentFulfillmentProcessToBe process.
- ▶ Implement the CheckIfVehicleReturned subprocess.
- ▶ Implement the component types that the process will use, including business rules, human tasks, and a selector.
- ▶ Create interface maps and data maps that enable you to call the PoolManager from the VehicleRentFulfillmentProcessToBe process.
- ▶ Assemble all of these elements, together with the previously imported artifacts, into the composite application, which will then be ready to test and deploy into the unit test environment.

12.2.1 Organize artifacts for reuse between modules

Before proceeding further, you should accomplish a few important code reorganizing tasks. First, create a new module to store the service implementation artifacts and the subprocess that you will create shortly. Also, move all of the interfaces from the main **VehicleRent** module to the **VehicleRentLib** library so that the interfaces can be referenced by all modules.

This is the rationale behind these changes: Storing only process-related information in the process module and only service-related information in the service modules means greatly increased flexibility and manageability of your system over time. For example, if the process changes, you need only make the changes to the process module without breaking dependencies on any services. Conversely, if a service implementation changes, you can manage it by changing it in the appropriate service module without affecting the process module.

The other change, moving all interfaces to the **VehicleRentLib** library, is made because in general a library is the appropriate place to share common artifacts such as business objects, data maps, and interfaces across multiple modules.

Start making these changes:

1. In the Business Integration view, right-click the white space and click **New** → **Module**. Give the new module a name of **VehRentSvcs** and click **Finish**.
2. Double-click the **VehRentSvcs** module. This displays the dependencies editor. Under Libraries, click **Add**, select the **VehicleRentLib**, and click **OK**. Press Ctrl+S to save, then close the editor.
3. Expand the **VehicleRent** module, then expand **Interfaces**. With your mouse, highlight and select all of the interfaces in **VehicleRent**, right-click them, and select **Move**. A dialog asks where to move them; designate the **VehicleRentLib** library and click **OK**. Wait a few moments while the workspace rebuilds.
4. Expand the **VehicleRent** module, then **Business Logic**, then **Processes**. Also expand **Human Tasks** and **Rule Groups**. While holding down the Ctrl key, select the process beginning with **CheckIfVehicleReturned**, all of the human task components, and all of the rule groups.

With all of these six elements selected, right-click them and select **Move**. In the dialog box that appears, choose to move them to the **VehRentSvcs** module and click **OK**. Wait for the workspace to rebuild. You will see several errors now in the Problems view due to unresolved or unimplemented components, but you will resolve them soon.

Note: We copied the subprocess out of the main module both for usefulness and to illustrate a point. The usefulness is that sometimes you might need to reuse subprocesses; therefore it might make sense to manage processes and subprocesses in separate modules. This illustrates the point that processes are just another type of SCA component and can be reused and imported between projects as with any other type of SCA component such as a human task or business rule.

5. Expand the **VehRentSvc** module, then **Business Logic**, then **Processes**, and double-click the process beginning with **CheckIfVehicleReturned**. Click **Partner** (in the variables view on the right side), then on the **Properties** tab and the **Details** tab. Set the interface by clicking **Browse**, selecting the **CheckIfVehicleReturned** interface, and clicking **OK**. Save the changes and close the process editor.

12.2.2 Customize logic in main business process

Now change the Java code in the two Read From Local Information Repository Java snippets in the main business process. The Java snippets are changed so that they correctly assign the value of needed process variables. A real-world implementation might use some of the other process variables that were created in the modeling phase with different scopes to differentiate between local variables and process variables. Also, something more sophisticated might have been done with the Read From Local Information Repository to, for example, read the RentalAgreement out of a relational database.

To make the sample scenario easier, use a simple Java snippet to assign the variables the way you need them:

1. In the Business Integration view, expand the **VehicleRent** module, then **Business Logic**, then **Processes**, and double-click **VehicleRentFulfillmentProcessToBe** to open the WS-BPEL process.

Take a moment to familiarize yourself with the contents of the process diagram. The process will look logically similar to the way it was implemented in WebSphere Business Modeler.

2. In the first While loop, click the Java snippet element entitled **Read From Local Information Repository**, then click the **Properties** tab at the bottom and select the **Details** subtab. Comment out the last line of this Java snippet and add a line to replace its function: Put a double-slash at the beginning of the last line of code, and below it add:

```
RentalAgreementVariable2 = RentalAgreementVariable;
```

3. In the second while loop, click the Java snippet element entitled **Read From Local Information Repository**, then click the **Properties** tab and the **Details** subtab as before. Comment out the last line and add the following line:

```
RentalAgreementVariable3 = RentalAgreementVariable;
```
4. Press Ctrl+S to save the WS-BPEL process, then close the process editor.

12.2.3 Implement subprocess

This section implements a subprocess that the main process uses. The subprocess will wait for two types of events:

► Vehicle return event

This simulates the initiative by ITSO Speedy Rentals to improve customer service and efficiency by tagging their rental cars with RFID tags that send events automatically when the vehicles near the rental location.

► Timeout event

This fires when a vehicle exceeds its planned return date. When this fires, the branch vehicle agent will contact the customer to determine a new planned return date and update the customer's rental agreement. This improves ITSO Speedy Rentals' vehicle inventory control.

Note: The VehicleRentProcessToBe relies on a subprocess beginning with the name CheckIfVehicleReturned. (The exact name will depend on the unique name generated by WebSphere Business Modeler.)

The CheckIfVehicleReturned subprocess is used as a reentry point into the main process. After the customer has rented a vehicle, the main process is suspended until the vehicle is returned or goes overdue. The CheckIfVehicleReturned subprocess is the mechanism by which the events of a returning vehicle or an overdue vehicle can be handled and the main VehicleRentProcessToBe can be notified to resume.

1. Expand the **VehRentLib** library, then expand **Interfaces**. Double-click the **CheckIfVehicleReturned** interface. Add an operation to this interface that the subprocess will need:
 - a. There are two icons at the top of the Interface editor; click the one that says **Add One-Way Operation**. An operation appears: name it TurnIn.

- b. A third icon appears at the top called **Add Input**: click this button. Keep the default name of input1, but change the type by clicking on **String**, then from the pull-down menu selecting **RentalAgreement** (Figure 12-12).

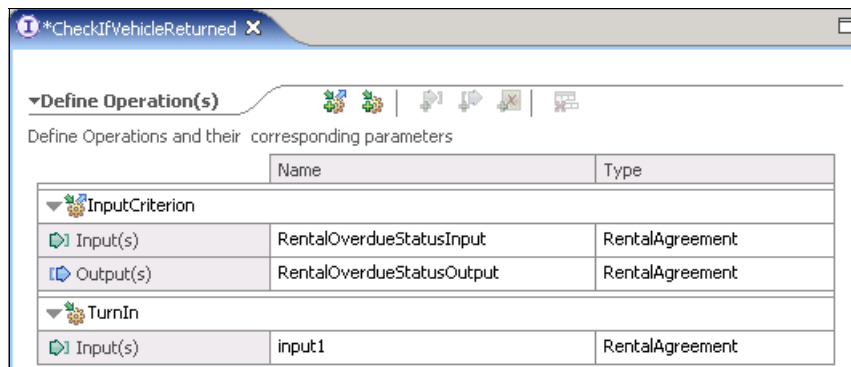


Figure 12-12 Addition of the TurnIn operation

- c. Press Ctrl+S to save the interface then close the interface editor.
2. Expand the **VehRentSvcs** module, then **Business Logic**, then **Processes**. Double-click the **CheckIfVehicleReturned** process to open it in the Business Process editor. Your initial process will look similar to Figure 12-13 with the palette of process elements on the left organized into slide-out drawers notated by arrows, the free-form surface in the middle, and the variables view on the right.

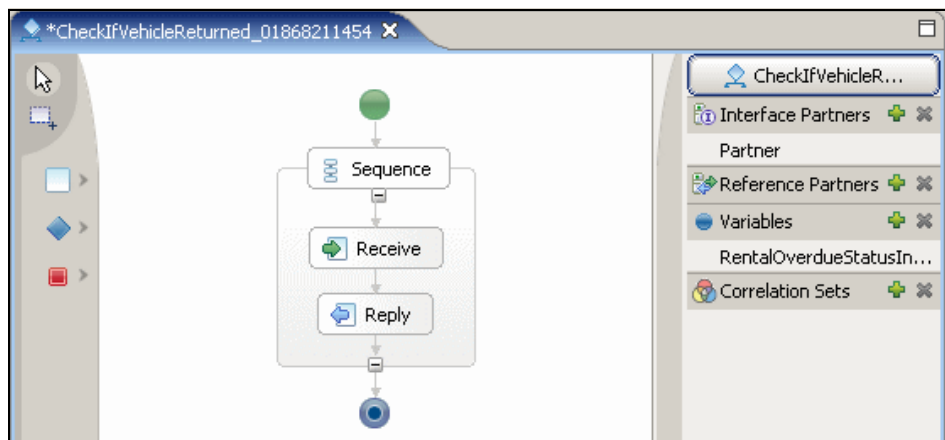


Figure 12-13 CheckIfVehicleReturned subprocess: initial state before implementation

3. You currently have a very basic sequential process with a Receive and a Reply. The first thing to do is add something to handle the data that is received into this subprocess and create two paths: one for a returning vehicle, the other for an overdue vehicle (Figure 12-14).
 - a. From the palette, select the second drawer of components, clicking the arrow to expand to reveal several components. Click the green arrow called **Receive Choice** and click to drop the ReceiveChoice component between Receive and Reply. A default name, ReceiveChoice, is highlighted; change the name to **Check TimeOut**.
 - b. An arrow from **Check TimeOut** points to a box named **Receive**. Select this box, click the **Properties** tab, then the **Description** subtab, and rename it to **Turn In**, which represents the returning vehicle event.
 - c. Click **Check TimeOut**, then hover over it until a clock icon appears. Click this **Add Timeout** icon, which creates a box named Timeout. This will be where the overdue vehicle event is handled.
 - d. From the palette, select the first drawer of components, and click the **Snippet** component. Add it to the process beneath the Turn In box, and rename the snippet **Set Overdue No**.
 - e. Create a second snippet, this time under the Timeout box, and name this snippet **Set Overdue Yes**.

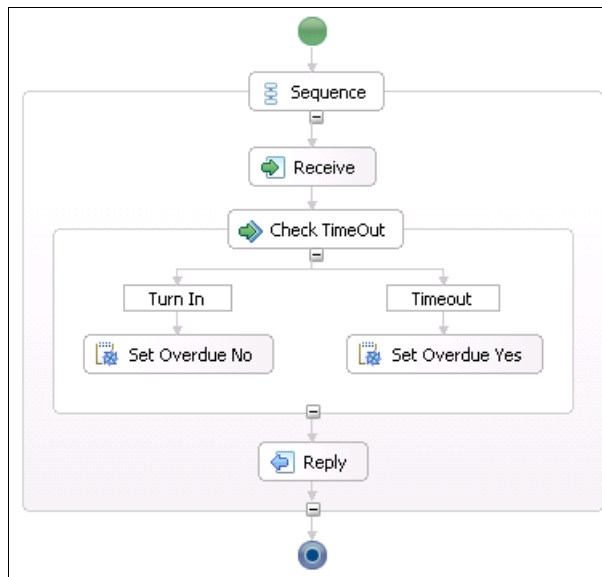


Figure 12-14 Additional activities added to the subprocess

4. Now add some detail to the components that you added to the subprocess:
 - a. Click the **Turn In** box, then click the **Properties** tab on the bottom and the **Details** subtab. For Partner, click **Browse**, choose the partner named **Partner**, and click **OK**. This fills in CheckIfVehicleReturned as our interface. For operation, select **TurnIn** from the pull-down box. For the variable, click ... and select **RentalOverdueStatusInputVariable**. Click **OK** (Figure 12-15).

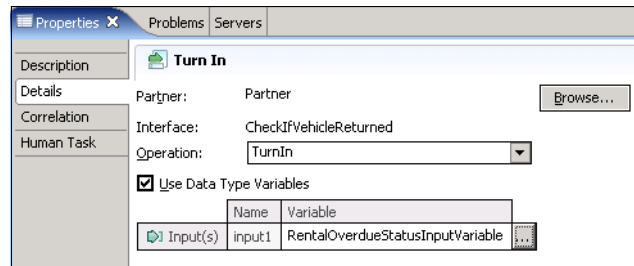


Figure 12-15 Setting the TurnIn details

- b. Now use a powerful capability called the Visual Snippet Editor, which enables you to visually create the logic needed for several elements in the process diagram:
 - i. Click the **Set Overdue No** box, then the **Properties** tab, then the **Details** subtab to take you into the Visual Snippet editor. Double-click the **Properties** tab to maximize the view.
 - ii. From the palette on the left, click the first element type called **Expression** and click the free-form surface to add it. Click inside the element and in the context menu expand **RentalOverdueStatusInputVariable**, click **OverdueStatus**, and click the white space to end the expression.
 - iii. Create another Expression element. Click inside the element, select **String**, and enter the String value of No.
 - iv. Connect the two expressions: Hover over the No expression until the yellow connector appears on the right, click it, and connect it to the OverdueStatus expression. We have visually configured logic to set the value of the OverdueStatus field in our input variable.
 - v. Repeat this process to create two more expressions, this time to set the RentalStatus of the RentalOverdueStatusInputVariable variable to a string of in (Figure 12-16 on page 281).

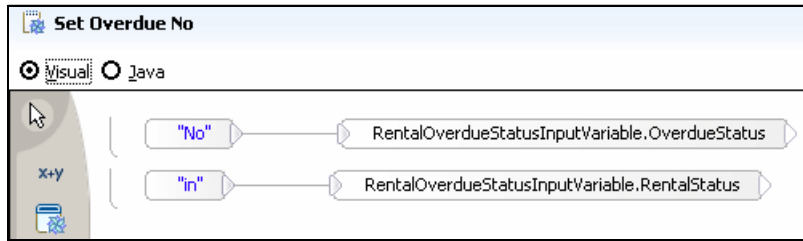


Figure 12-16 Visual Snippet Editor for Set Overdue No

- vi. Double-click the **Properties** tab to return the Visual Snippet Editor to its previous size and save the changes.
- c. Click the **Set Overdue Yes** box, then the **Properties** tab, then the **Details** subtab. Double-click the **Properties** tab to maximize the Visual Snippet editor. Click and drop four expressions as in the previous step, but this time the resulting constructs will be OverdueStatus connected to Yes and RentalStatus connected to out (Figure 12-17). When this is complete, double-click the **Properties** tab to reset the view.

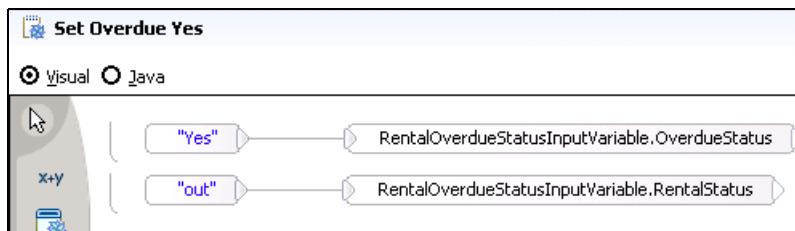


Figure 12-17 Visual Snippet Editor for Set Overdue In

- d. Configure the remaining element, Timeout, to determine when a timeout occurs so that this branch will work. Click **Timeout**, then **Properties**, then **Details**. From the pull-down menu, select **Java** as the Expression Language and **Visual** as the Expression type. If you see a dialog asking you to confirm the switch to Visual, click **Yes**.
 - i. A default expression appears in the window. You will use a different one, so drag a box around all elements in the diagram and press the Delete key.
 - ii. Use a predefined snippet to be imported into the workspace. Click **File** → **Import** → **File System** and click **Next**. Browse to the \WID\Resources subdirectory of the additional material supplied with this book and click **OK**. Check **DateTimeToCalendar.activity**, and place it into the VehRentSvc\date folder. Click **Finish** to complete the import.

Note: Sometimes useful functions such as this one can be defined as Java snippets by Java developers, to be reused by integration developers to perform their integration tasks. This snippet takes as input a `DateTime` value representing a date/time and a `String` value representing the time zone, and returns a `Calendar` object as an output. This is useful because the `Timeout` element in the process editor requires a value of type `Calendar`, thus an integration developer can reuse this very helpful snippet.

- iii. From the palette on the left of the Visual Snippet editor, click the third icon from the top, which is **Java**. In the dialog that appears, type `DateTimeToCalendar` for Type, and in the Select a Visual Snippet box select **`dateTimeToCalendar(Date,String)`**. Click **OK**. Drop the element onto the surface. The element has two input nodes on its left side, with red marks by them: these are two inputs that you as the integration developer must provide. One is the planned return date, the other is the time zone.
- iv. Click and drop an expression onto the free-form surface. Click it and in the context menu expand **`RentalOverdueStatusInputVariable`**, click **`PlannedReturnDateTime`**, then click white space to end the expression. Now connect this expression using its yellow connector on the right side to the first input node of the `dateTimeToCalendar` snippet.
- v. Click and drop another expression. Click it, click **`String`**, enter a value of `GMT`, press Enter, and click white space to end. Connect this expression to the second input node of the `dateTimeToCalendar` snippet.
- vi. From the palette, click the last icon, which is **Return**, and drop it onto the surface. Now connect the yellow connector on the right side of the `dateTimeToCalendar` snippet to the Return element.

This completes the logic for this expression, which provides a date value, converted to `Calendar` type, that our `Timeout` logic will use to determine the timeout condition. It should resemble Figure 12-18 on page 283.

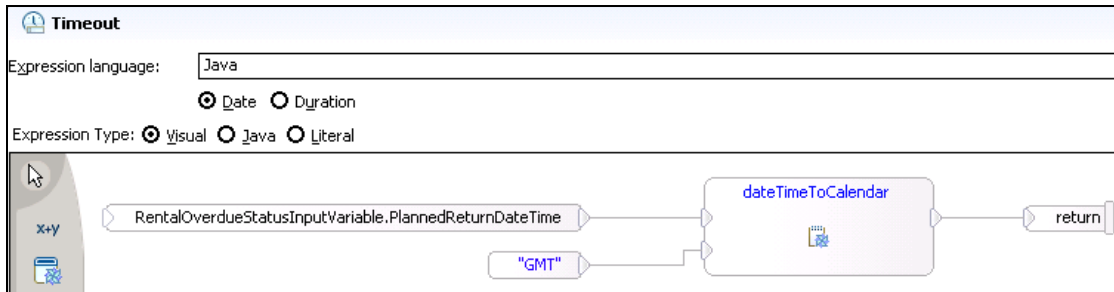


Figure 12-18 Visual Snippet Editor for Timeout

- vii. Double-click the **Properties** tab to reset the view, and save the changes.
5. This subprocess is designed to handle events (returning vehicle, vehicle timeout), and it will restart the main process that has been paused waiting for one of these events. Create a correlation set (a set of unique identifiers that provides a way for the process engine to know which process instance to restart and resume) that this subprocess will use for this purpose.
 - a. On the right side of the business process editor, click the + sign next to Correlation Sets to add a correlation set. Name it `ReservationID`. Click the **Properties** tab, then the **Details** subtab, then click **Add** to add a property.
 - b. In the Select a Property dialog, click **New**. Name the property `ReservationID`. Browse for type, select **String**, and click **OK**.
 - c. To the right of Aliases, click **New** to create an alias. For interface select **CheckIfVehicleReturned**, and for operation select **InputCriterion**. In the Part view, expand **InputCriterion** → **RentalOverdueStatusInput**, select **ReservationID**, and click **OK** (Figure 12-19).

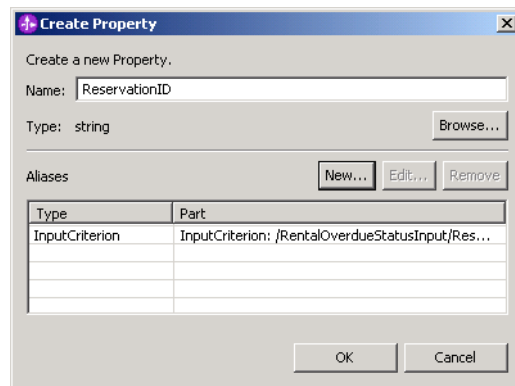


Figure 12-19 Create Property dialog

- d. Click **New** again to create another alias. For interface select **CheckIfVehicleReturned**, and for operation select **TurnIn**. In the Part view expand **TurnInParameters** → **input1**, select **ReservationID**, and click **OK**. Click **OK**, then **OK** again to complete the creation of the CorrelationSet property. This property is the unique identifier of the correlation set.
- e. Click **Receive** in the process editor, then in the **Properties** tab select the **Correlation** subtab. Click **Add**, which fills in a value of ReservationID for the correlation set. Change the value of Initiation to **Yes** (Figure 12-20).

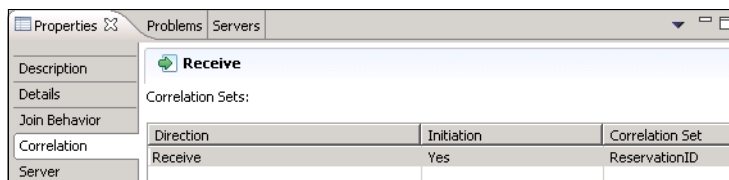


Figure 12-20 Correlation settings for the Receive activity

- f. Click the **TurnIn** condition under the Check Timeout element. In the **Properties** tab, select the **Correlation** subtab. Click **Add**, which fills in values. Accept all of the values, including a value for Initiation of No.
- g. Press Ctrl+S to save the process. Your completed subprocess will look like Figure 12-21, with no errors or warnings on the process diagram. Close the process editor.

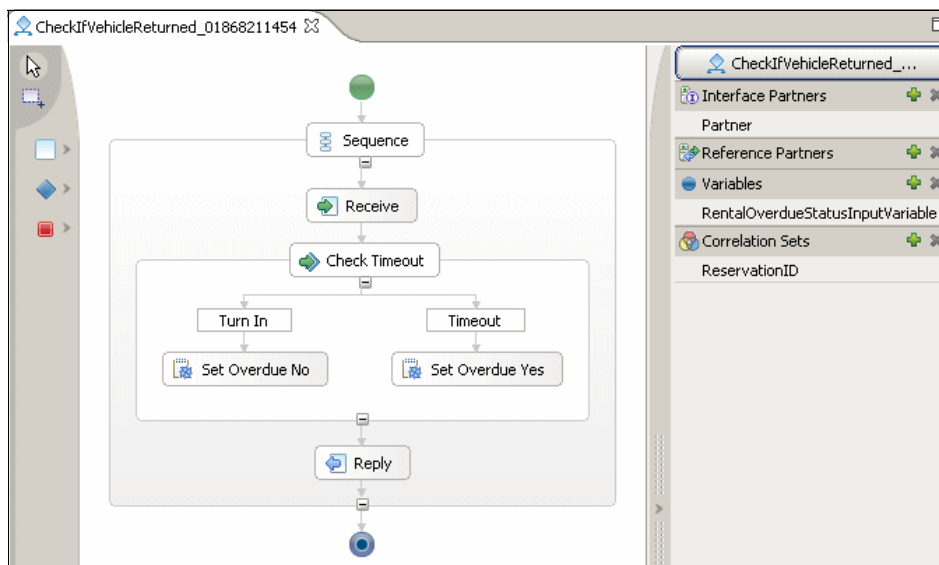


Figure 12-21 Completed CheckIfVehicleReturned subprocess

12.2.4 Implement components

This section implements several of the components needed by our vehicle rental process. You will create business rules to handle vehicle assignment and discount calculation, and human tasks that will enable ITSOSpeedyRental employees to complete rental completion, agreement update, and vehicle return tasks. You will also create a selector component that enables you to dynamically choose between different payment services at runtime.

Implement business rules

Business rules are a natural complement to an automated business process management system. They provide the capability to externalize the organization's business rules outside of the IT system, enabling the rules to be reconfigured dynamically, thus affording the organization greater speed of change. In this section you create two different types of rules in WebSphere Integration Developer: a decision table and a ruleset.

1. Create a decision table in this business rule. The purpose of this business rule is to reassign a vehicle class if the initially requested vehicle class is unavailable. The vehicle classes go alphabetically from lower to higher quality, so C vehicles are higher quality than B and A. Therefore the rule will assign a higher-class vehicle when a customer's risk level is low or medium, and if no higher-class vehicles are available, the rule will assign a lower-class vehicle.
 - a. In the Business Integration view, expand the **VehRentSvcs** module, then **Business Logic**, then **Rule Groups**. Double-click the business rule group starting with the name **AssignHigherGroupVehicle**. This opens the rule group editor.
 - b. Click **InputCriterion**, then next to Default Destination click **Enter Destination** and choose **New Decision Table** from the list.
 - c. In the New Decision Table dialog, perform the following steps:
 - i. Name the decision table `AssignHigherCarClassRule` and click **Next**.
 - ii. Ensure that Select a layout is set to **Row**. Set Number of row conditions to **2** and Number of actions to **2**. Keep Row for Layout. Click **Finish**, and a blank decision table appears.
- Note:** In a decision table, conditions are evaluated according to condition values. If a set of condition values evaluates to true, then the appropriate action the values are mapped to will be performed.
- d. In the first row of the decision table, click inside any of the white cells, then right-click and choose **Add Condition Value**. In the second row of the decision table, click inside any of the white cells, then right-click and

choose **Add Condition Value**. This makes it possible to accommodate the right number of values for our conditions.

- e. Fill out the first condition: Click the column with **Enter Term** in the first row, and select **AssignHigherGroupVehicleInput.VehicleGroupRequested**. In the other three columns in this row, select **String** from the pull-down list and the values **A**, **B**, and **C** respectively. These are the three car class types at ITS0 Speedy Rentals that may be requested (Figure 12-22).

The screenshot shows a 'Table' window with a 'Conditions' tab. The first row is being edited. The first column contains the text 'AssignHigherGroupVehicleInput.VehicleGroupRequested'. The next three columns are labeled 'A', 'B', and 'C'. Each of these columns has a dropdown menu set to 'String'. The last column is labeled 'Enter Term'. The 'Actions' tab is visible at the bottom right.

Figure 12-22 Specifying the vehicle groups in the decision table

- f. In the second row, click **Enter Term** and expand **AssignHigherGroupVehicleInput**, then **DriverDetails**, then choose **RiskLevel**. In the first three columns in this row, enter String values of high, medium, and low. This fills up all nine columns in the row (Figure 12-23). These are the three driver risk ratings for categorizing the level of riskiness of the customers, and which will be used to drive the determination of vehicle assignment in this business rule.

The screenshot shows the same 'Table' window. The second row is now being edited. The first column contains the text 'AssignHigherGroupVehicleInput.DriverDetails.RiskLevel'. The next three columns are labeled 'high', 'medium', and 'low'. Each of these columns has a dropdown menu set to 'String'. The last column is labeled 'Enter Term'. The 'Actions' tab is visible at the bottom right.

Figure 12-23 Specifying the risk level in the decision table

- g. In the third row, click **Enter Term** and choose **AssignHigherGroupVehicleOutput**. For all remaining column values, select **AssignHigherGroupVehicleInput**. This action creates the output of the rule using the contents of the input to the rule.
- h. In the last row, click **Enter Term**, expand **AssignHigherGroupVehicleOutput** and click **VehicleGroupRequested**. In the remaining nine columns, enter string values of B, B, B, A, C, C, B, B, B.

- i. In the last row again, in each of the nine column values just entered, perform these steps:
 - i. Click inside the cell containing each value (such as B).
 - ii. Locate the last icon on the right above the table that says **converts an expression of a decision table cell into a template**, and click this button. This turns the literal value into a template value so that later, business users will be able to change these values in the Business Rule Manager Web console.
 - iii. Repeat for all nine column values.
- j. Press Ctrl+S to save the rule. The completed rule should look similar to Figure 12-24. Close the decision table editor, then press Ctrl+S to save the AssignHigherGroupVehicle rule group.

Decision Table
Detailed properties for this decision table.

Interface

	Name
Input(s)	AssignHigherGroupVehicleInput
Output(s)	AssignHigherGroupVehicleOutput

Table

Conditions				
AssignHigherGroupVehicleInput.VehicleGroupRequested	"A"			
AssignHigherGroupVehicleInput.DriverDetails.RiskLevel	"high"	"medium"	"low"	
AssignHigherGroupVehicleOutput	AssignHigherGroupVehicleInput	AssignHigherGroupVehicleInput	AssignHigherGroupVehicleInput	AssignHigherGroupVehicleInput
AssignHigherGroupVehicleOutput.VehicleGroupRequested	"B"	"B"	"B"	

Figure 12-24 Completed AssignHigherCarClassRule decision table

2. Create a Ruleset: In the Business Integration view, double-click the Rule Group beginning with the name **CalculateDiscounts** to open the rule group editor.
 - a. Click **InputCriterion**, then next to Default Destination click **Enter Destination** and choose **New Ruleset** from the pull-down list. Keep the default folder, name the ruleset PriceRuleSet, and click **Finish**.
 - b. In the Ruleset editor, locate the Variables section to add several variables that our ruleset will need to use. Click the + sign to add four variables. Give them the names and types specified in Table 12-1 on page 288.

Table 12-1 Variables for PriceRuleSet

Name	Type
dailyCharge	double
rentalDays	double
rentalHours	double
discountRate	int

- c. Add some action rules (pieces of logic that should execute at a certain time):
 - i. In the Rules section, click the second button from the left to **Add Action Rule**. Give the rule a name of `Init` and Presentation text of `Init`. This action rule will serve to initialize our rule variables.
 - ii. In the Action section, click **Action**, choose **discountRate**, then the `=` operation, then **Number**, enter `0`, and press Enter.
 - iii. Press Enter four times to create four new actions. Repeat the previous step, using the following settings for these actions:


```
dailyCharge=50
rentalHours=0
rentalDays=0
CalculateDiscountOutput=CalculateDiscountInput
```
 - iv. Click the **Add Action Rule** button again to add another action rule. Name it `CalculateRentalDuration` and enter Calculate rental hours and days in the Presentation section.
 - v. In the Action section, highlight the **Action** field, click Delete to clear it, and enter the following code on a single line and press Enter:


```
rentalHours=((CalculateDiscountInput.ActualReturnDateTime.getTime()-CalculateDiscountInput.PickupDateTime.getTime())/3600000)
```
 - vi. Highlight the **Action** field, click Delete to clear it, and enter the following line of code and click Enter:


```
rentalHours =
((CalculateDiscountInput.ActualReturnDateTime.getTime() -
CalculateDiscountInput.PickupDateTime.getTime())/3600000)
```

Add the following line of code:

```
rentalDays = (rentalHours/24)
```

This completed rule calculates the number of rental days that a returned car has been out since it was initially rented.

vii. Click the **Add Action Rule** button once more to add an action rule. Name it `CalculateFinalPrice` and enter `Calculate final price` in the Presentation section.

viii. Enter the following code into the Action field on a single line:

```
CalculateDiscountOutput.ActualCost = (dailyCharge *
rentalDays) - (dailyCharge * rentalDays * discountRate/100)
```

As you can tell, this rule computes the final rental charge. At this point, the Rules section should resemble Figure 12-25.

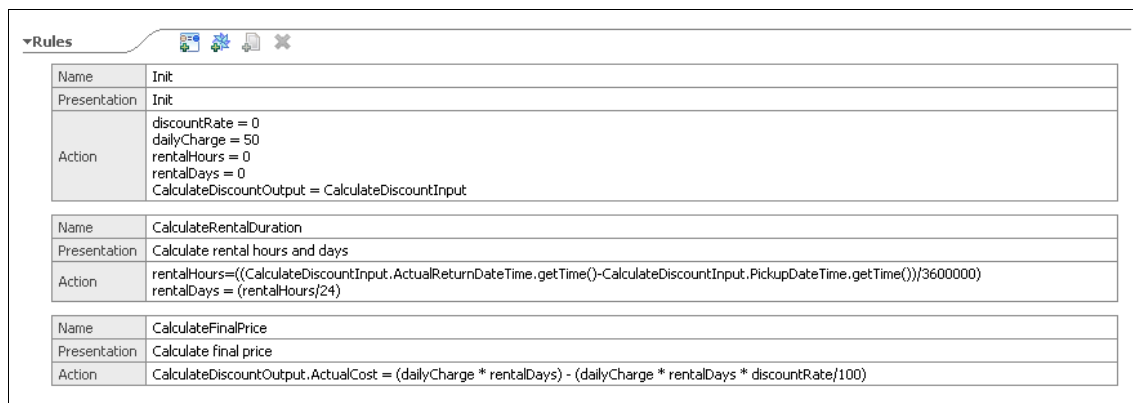


Figure 12-25 Completed action rules in `PriceRuleSet`

d. Create some rule templates: This is a powerful feature that enables you to create a model by which other rules can be created.

- i. In the Templates section, click the first button **Add If-Then Template** and name the template `Assign daily charge per vehicle type`.
- ii. Create two parameters in the Parameters section using the green + sign. Name one `VehicleType` with type `string`, and name the other `DailyCharge` with type `double`.
- iii. In the Presentation section, enter `If vehicle type is Class`. Use the expression builder to choose the parameter **VehicleType**, and enter `then daily charge is`. Pick the parameter **DailyCharge**, and enter `USD`.

Hint: To display the expression builder after typing text, simply click the white space after the text you just typed.

- iv. In the If section, use the expression builder to build this expression:
`CalculateDiscountInput.VehicleGroupRequested==VehicleType`

v. In the Then section, build the following expression:

dailyCharge=DailyCharge

This completes the first rule template, whose function is to create rules that set rates by vehicle type (Figure 12-26).

Templates				
Name	Assign daily charge per vehicle type			
Presentation	If vehicle type is Class {VehicleType}, then daily charge is {DailyCharge} USD			
Parameters	Name	Type	Constraint	
	VehicleType	string	None	
	DailyCharge	double	None	
If	CalculateDiscountInput.VehicleGroupRequested == VehicleType			
Then	dailyCharge = DailyCharge			

Figure 12-26 Template to assign a daily charge based on vehicle type

- vi. In the Templates section, click the **Add If-Then Template** button, and name the template Assign discount rate per loyalty type.
- vii. Add two parameters, one called LoyaltyType of type string, and the other called DiscountRate of type int.
- viii. In the Presentation section, enter If driver’s loyalty type is, insert the LoyaltyType parameter, enter then apply, insert the DiscountRate parameter, and enter % discount.

Note: The presentation section is where the text is entered for the Business Rules user interface. By creating a user interface that a business analyst can interact with, the rule developer enables a beneficial separation of roles. The rule developer maintains the rules, and the business analyst can flexibly change business rules as business needs require.

- ix. In the If section, type the following expression on a single line:
CalculateDiscountInput.DriverDetails.LoyaltyClubNumber.startsw
ith(LoyaltyType)

x. In the Then section, build the following expression:

discountRate=DiscountRate

This completes the second rule template (Figure 12-27 on page 291), which assigns a different discount rate for loyalty club customers of various types.

Name	Assign discount rate per loyalty type		
Presentation	If drivers loyalty type is 'LoyaltyType' then apply 'DiscountRate' % discount		
Parameters	Name	Type	Constraint
	LoyaltyType	string	None
	DiscountRate	int	None
If	CalculateDiscountInput.DriverDetails.LoyaltyClubNumber.startsWith(LoyaltyType)		
Then	discountRate = DiscountRate		

Figure 12-27 Template to assign discount rate based on loyalty type

- e. The last task in creating this rule set is to use the newly created rule templates to build some rules:
 - i. In the Rules section, click the third button called **Add template rule**. Two choices pop up, asking which template to use. Select **Assign daily charge per vehicle type**. Type the name `DailyChargeForClassA`. There are two parameter boxes to fill: use A for the first and 50 for the second.

Note: Clearly, Rule Templates are a handy feature. We use the rule template to set parameterized rules, in this case helping to create a rule setting the rental rate for Class A vehicles at \$50.

- i. Create two more template rules of the same type. Name the first `DailyChargeForClassB`, setting B and 70 in the parameter boxes. Name the second `DailyChargeForClassC`, setting C and 90 for the parameters.
 - ii. Create two template rules of the type **Assign discount per loyalty type**. Name the first `AssignDiscountRateForTypeACustomerRule`, setting A and 40 in the parameter boxes. Name the second `AssignDiscountRateForTypeBCustomerRule`, setting B and 20 in the parameter boxes.

This completes all of the template rules, and this part of the Rules section should resemble Figure 12-28 on page 292.

Name	DailyChargeForClassA
Template	Assign daily charge per vehicle type
Presentation	If vehicle type is Class A then daily charge is 50 USD
Name	DailyChargeForClassB
Template	Assign daily charge per vehicle type
Presentation	If vehicle type is Class B then daily charge is 70 USD
Name	DailyChargeForClassC
Template	Assign daily charge per vehicle type
Presentation	If vehicle type is Class C then daily charge is 90 USD
Name	AssignDiscountRateForTypeACustomerRule
Template	Assign discount rate per loyalty type
Presentation	If driver's loyalty type is A then apply 40 % discount
Name	AssignDiscountRateForTypeBCustomerRule
Template	Assign discount rate per loyalty type
Presentation	If driver's loyalty type is B then apply 20 % discount

Figure 12-28 Completed template rules in PriceRuleSet

- f. The PriceRuleSet is complete. Press Ctrl+S to save the ruleset and close the ruleset editor. Press Ctrl+S to save the CalculateDiscounts rule group.

Implement human tasks

Human workflow is an indispensable component of any business process management system. The combination of automated tasks and human tasks fully leverages a company's two major assets: its information technology and its human resources. This section implements the human task components by assigning tasks to roles.

1. In the Business Integration view, expand the **VehRentSvcs** module → **Business Logic** → **Human Tasks**. You will work with three human tasks. Open the **UpdateRentalAgreement** task by double-clicking it.
2. For this scenario, work with the Receiver setting, mainly by using two different *Verbs*. Verbs are the way that the human task manager determines how to assign human tasks. In the UpdateRentalAgreement human task, click **Potential Owner** under Receiver Settings. Look at the settings in the **Properties** tab (Figure 12-29 on page 293).

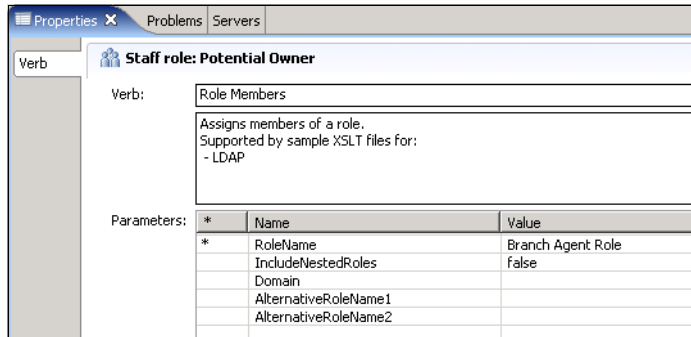


Figure 12-29 Potential owner properties for UpdateRentalAgreement

3. In the Properties tab, the verb is set to Role Members, and the Role Name is set to Branch Agent Role. These settings were configured during the business modeling phase (in WebSphere Business Modeler) and were exported as human task types for use by WebSphere Integration Developer.

The verb is supported by sample XSLT files for LDAP. This means that the Human Task Manager uses different XSLT files for different user registry types. In a production environment, typically an LDAP user registry would be used. In a development environment, it might be useful to test with a custom registry, such as a file-based registry, which we intend to use in this sample scenario in the development phase.

However, as stated, there is a limitation in that the UserRegistry XSL provided inside the Unit Test Environment does not implement the Role Members verb. Remedy this by using a modified XSL file and a file-based user registry in order to test using this configuration in our development environment. This enables testing and experimenting with the human task functionality. When we export our artifacts out of WebSphere Integration Developer for deployment, we will change the verb to Everybody to simplify the testing in WebSphere Process Server, which will not be configured with the same file registry that we are testing with here.

For now, there is nothing else to configure on the human task components, so close the VehicleCollection human task for now. There will be further instructions in 12.3.1, “Setting up the test environment” on page 307, which explains how to configure the artifacts needed for the file-based registry so that we can test the human task components using the Role Members verb.

Implement a selector

Selectors provide a way to dynamically invoke different services at runtime based on defined criteria, such as time of day. This provides an enhanced degree of flexibility to composite applications, allowing service binding decisions to not be

finalized at development time, but instead to be calculated at runtime. The selector we use in our scenario is tied to the Payment service. We will institute a selector that can choose between three different implementations of the Payment service: one based on Java, one based on a Web service invocation, and one based on a JCA filesystem adapter.

1. The first task is to create a Java implementation for the Payment service. In the Business Integration view, expand the **VehRentSvc**s module and double-click the **VehRentSvc**s assembly diagram. A blank surface appears because no components have been defined, although eventually all of our services will be defined on this diagram.
2. In the palette on the left, expand the first drawer of components. Click the **Java** button and drop it onto the free-form surface. Rename the component Payment.
3. Right-click the **Payment** component and select **Add** → **Interface**. In the Add Interface dialog, click the **Show WSDL** radio button, select the **Payment** interface, and click **OK**.
4. Right-click the **Payment** component and select **Generate Implementation**. In the dialog that appears, click **New Package**, type `javacomp` for the package name, and click **OK**. Select the `javacomp` package and click **OK**. A new Java class appears in a Java editor.
5. Locate the `InputCriterion` method, and replace its contents with the following code:

```
String actualCost = paymentInput.getString("ActualCost");
System.out.println("Payment done with actualCost : " + actualCost);
return paymentInput;
```
6. Press **Ctrl+S** to save the Java file, and close the Java editor.
7. Back in the assembly diagram, expand the third component drawer, click the **Selector** icon, and add it to the free-form surface. Rename the component to `PaymentSelector`.
8. Right-click the **PaymentSelector** component and select **Add** → **Interface**. In the Add Interface dialog, select the **Payment** interface and click **OK**.
9. Right-click **PaymentSelector** and select **Generate Implementation**. Click **New Folder**, name the new folder selector, and click **OK**, and **OK** again.
10. The Selector editor opens. Select **InputCriterion**. Click **Enter SCA Destination**, and select **Payment** as the default destination (Figure 12-30 on page 295). This makes the newly created local Java component the default implementation for this selector. Save and close the selector editor.

Save and close the assembly diagram. This completes the implementation of our service components.

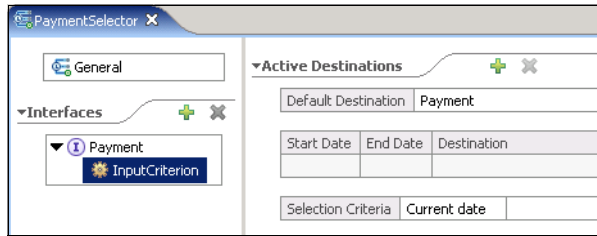


Figure 12-30 PaymentSelector

12.2.5 Create interface maps and data maps

Before you assemble the composite application, be aware of this integration dilemma.: Often a process or component has been defined to consume a certain type of interface, but the service needing to be consumed has a different interface (for example, between the VehicleRentFulfillmentProcessToBe process and the PoolMgrSvc module). This section resolves this by creating interface maps and data maps that translate between the service invocation and the service itself.

1. In the Business Integration view, expand the **VehicleRent** module and double-click the **VehicleRent** assembly diagram. The opened VehicleRent diagram appears on the free-form surface (Figure 12-31).

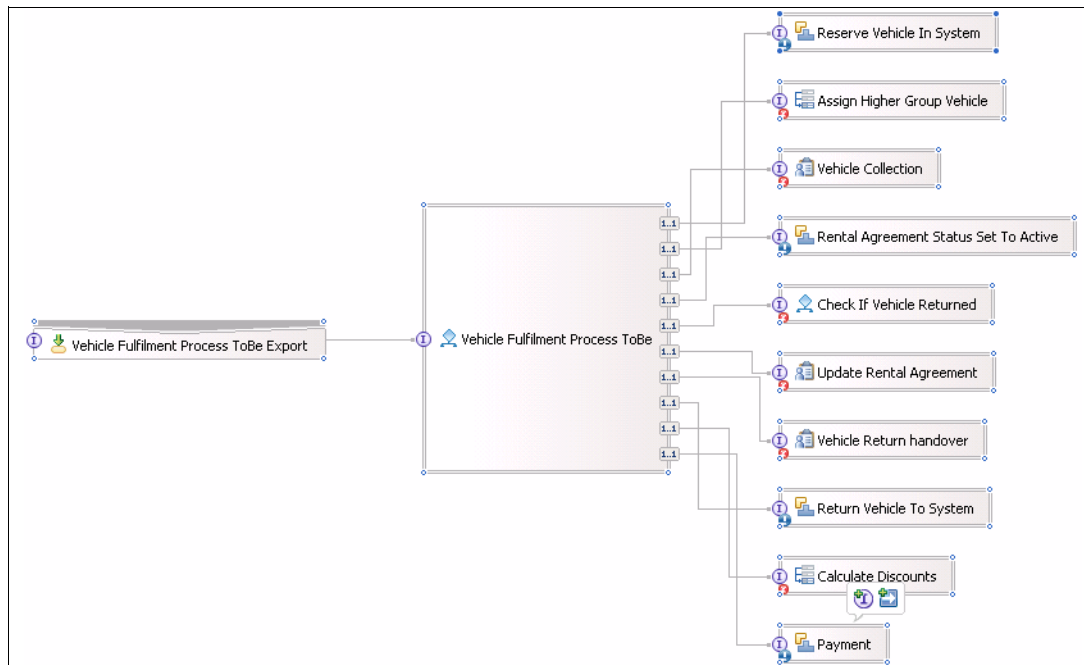


Figure 12-31 Initial state of the VehicleRent assembly diagram

2. On this assembly diagram, you will soon connect the main process to all of its dependent services. But first, examine one of these services for a moment. Click the component named **Reserve Vehicle In System**, then click the **interface** icon to the left of it, as shown in Figure 12-32.



Figure 12-32 Selecting Interface for Reserve Vehicle In System component

3. Make sure that the **Properties** tab and the **Details** subtab on the bottom pane are selected. Expand the **ReserveVehicleInSystem** interface, revealing a single operation called **InputCriterion** as shown in Figure 12-33.

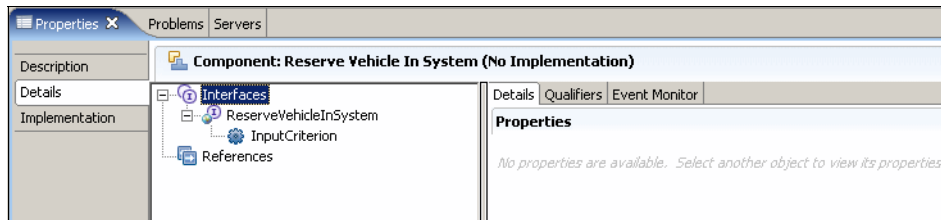


Figure 12-33 Properties for Reserve Vehicle In System

This is important because the **ReserveSvc**s module we created earlier does not have a matching interface. The **ReserveSvc**s module has an interface called **PoolManager**, with three operations **ReserveVehicle**, **ReleaseVehicle**, and **ReturnVehicle**. In this section, we will take care of the interface mismatch by creating interface maps and data maps.

4. Create the **MapToReserve** interface map:
 - a. In the Business Integration view, expand the **VehRentSvc**s module, then double-click to open the **VehRentSvc**s assembly diagram.
 - b. Select the interface map component from the palette (shown in Figure 12-34) and click the free-form surface to add it. Rename the map to **MapToReserve**.

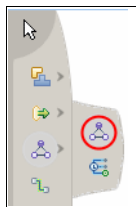


Figure 12-34 Selecting Interface Map from assembly diagram palette

- c. Repeat the previous step to create two more interface maps, naming one MapToRelease and the other MapToReturn.
5. Configure the MapToReserve interface map:
 - a. Right-click the **MapToReserve** component and select **Add → Interface**. In the Add Interface dialog, select **ReserveVehicleInSystem** and click **OK**.
 - b. Right-click the **MapToReserve** component again and select **Add → Reference**. In the Add Reference dialog, select **PoolManager** and click **OK**. Choosing the invocation interface and the reference interface provides the input and output interfaces for the interface map.
6. Implement the MapToReserve interface map:
 - a. Right-click the **MapToReserve** component and select **Generate Implementation**. The Generate Implementation dialog appears, asking what folder to generate the map into. Choose the default by clicking **OK**. This opens the interface mapping editor.
 - b. The interface mapping editor displays the two interfaces. Click the **InputCriterion** operation in the ReserveVehicleInSystem interface, and while holding down the left mouse button, drag a line over to the **reserveVehicle** operation on the PoolManager interface. This creates an arrowed line pointing in the direction of the PoolManager interface.
 - c. Click the arrowed line as shown in Figure 12-35. Parameter mappings appear.

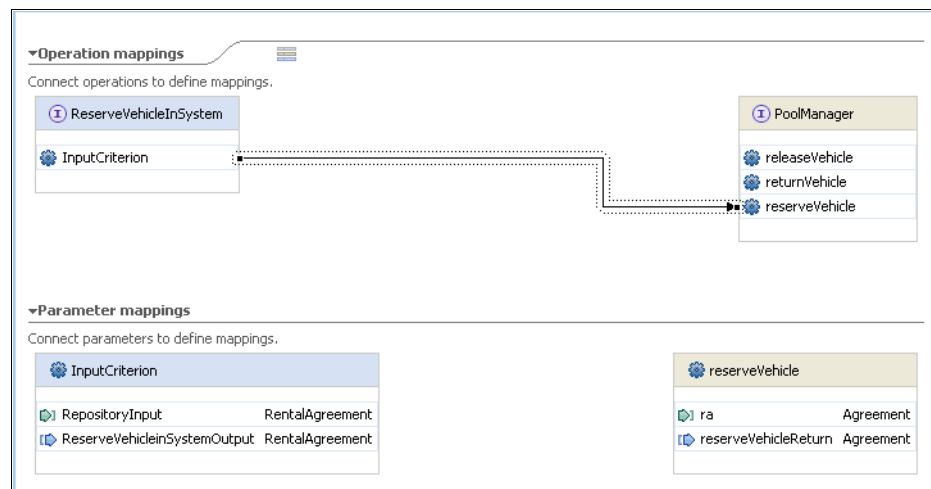


Figure 12-35 Mapped InputCriterion to reserveVehicle on MapToReserve interface map

- d. The parameter mappings view shows the need to map the parameters of these operations. Drag a line from **RepositoryInput** on the left to **ra** on the

right, and then from **reserveVehicleReturn** on the right to **ReserveVehicleInSystemOutput** on the left. This produces two Move operations, one pointing left to right and the other pointing right to left (Figure 12-36), which indicate the directionality of the maps as well as the type of map operations, in this case Move.

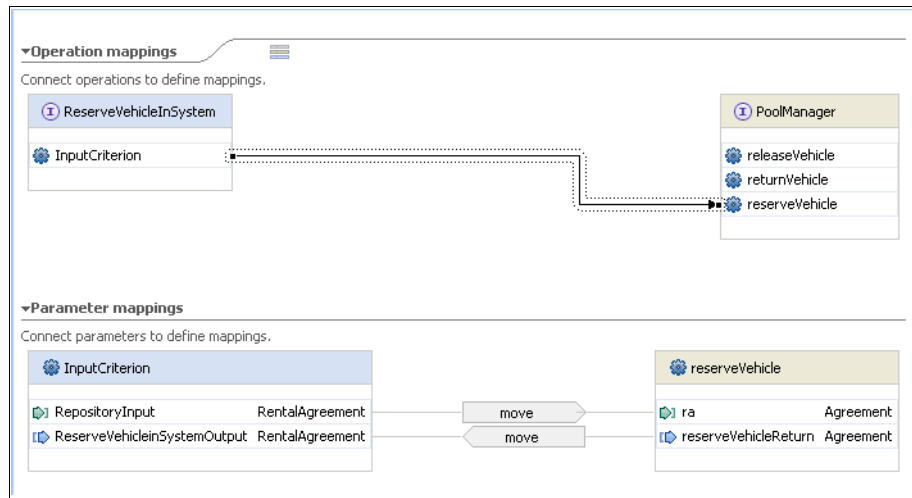


Figure 12-36 Initial parameter mappings for MapToReserve

Note: Change the default Move operations because RentalAgreement (a business object defined by the process) and Agreement (a business object defined by the service) do not contain identical fields and datatypes. The solution is to define two data maps (also known as Business Object Maps): one that converts RentalAgreement objects to Agreement objects, and one that converts Agreement objects to RentalAgreement objects. Then configure your Interface Maps to use these Data Maps so that the business objects will be appropriately transformed as needed.

- e. Click the top **Move** operation, then click the **Properties** tab at the bottom.
- f. Implement the business object maps:
 - i. Select **Map** from the Parameter Mapping Type list.
 - ii. Select the **Details** subtab. Click **New** to create a new business object map.
 - iii. In the New Business Object Map dialog, enter RentalAgmtToAgmt for the name, and click **Next**. Make sure that **RentalAgreement** is selected for the input type and **Agreement** is selected for the output type. Click **Finish**.

- iv. In the Business Integration view on the left, expand **VehRentSvcs** → **Mapping** → **Data Maps**, then double-click **RentalAgmtToAgmt**. This opens the Business Object Map editor. You need to map the fields on the left to the appropriate fields on the right.

Note: This is a very powerful feature because you can do many different types of operations to the fields in one business object to transform them to the other business object type. In addition to moving, you can perform join, extract, assign, and custom operations (which enable you to build expressions visually or in Java that accomplish the specific transformation operation that you need). For this example, we use only move operations to keep the example simple. However, consider that the data type conversions in this example will be performed automatically and that the move operations enable you to easily take the subset of data you need from the source business object and move it into the target business object, without any coding.

- v. In the Business Object Map editor, select the **ReservationID** from **RentalAgreement**, hover over it until the yellow connector appears, then click it and drag to the **agreementID** on **Agreement**. This produces a Move operation between these two fields.
- vi. Perform the same operation for all of the fields in Table 12-2.

Table 12-2 RentalAgreement and Agreement fields

RentalAgreement field	Agreement field
VehicleGroupRequested	vehicleGroupRequested
VehicleGroupReceived	vehicleGroupReceived
PickupDateTime	pickupDate
PlannedReturnDateTime	plannedReturnDate
ActualReturnDateTime	actualReturnDate
RentalStatus	rentalStatus
DriverDetails → LicenseID	driverID
DriverDetails → RiskLevel	driverRiskLevel
DriverDetails → LoyaltyClubNumber	royaltyClubID
VehicleDetails → Available	requestedVehicleAvailable

RentalAgreement field	Agreement field
VehicleDetails → Manufacturer	vehicleMake
VehicleDetails → ModelName	vehicleModel
VehicleDetails → InventoryID	vehicleInventoryID

The result is a completed business object map (Figure 12-37). Press Ctrl+S to save the business object map and close the business object map editor.

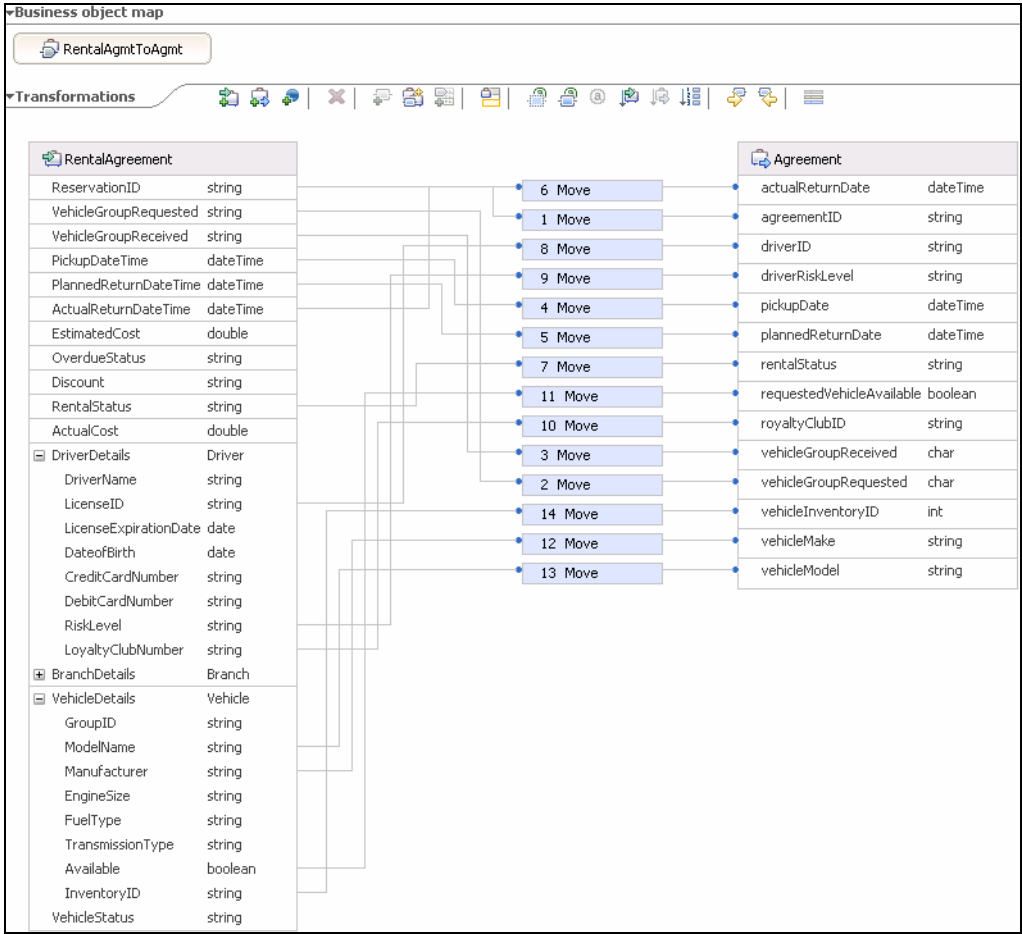


Figure 12-37 Completed RentalAgmtToAgmt business object map

- vii. Complete another business object map to transform the business objects in reverse, from Agreement to RentalAgreement. Within the MapToReserve interface map diagram, make sure the connector between InputCriterion and reserve Vehicle is selected. In the Parameter mappings section, select the second move operation.
 - viii. In the **Properties** tab, set the Parameter Mapping Type to **Map**. Select the **Details** subtab. Click **New** to create a new Business Object Map.
 - ix. Name this business object map AgmtToRentalAgmt, click **Next**, make sure **Agreement** is checked for the input type and **RentalAgreement** is checked for the output type, and click **Finish**.
 - x. Open **AgmtToRentalAgmt** in the business object map editor. Create all of the move operations in reverse the order from the previous map. Use Table 12-2 on page 299 as a guide, but perform the field mappings in reverse order. After completing this business object map, save and close the editor.
 - xi. Save and close the interface mapping editor.
7. Create the MapToRelease interface map:
- a. In the VehRentSvcs assembly diagram, right-click the **MapToRelease** interface map and select **Add** → **Interface**. In the Add Interface dialog, select the **RentalAgreementStatusSetToActive** interface and click **OK**.
 - b. Right-click the **MapToRelease** interface map again and select **Add** → **Reference**. In the Add Reference dialog, select the **PoolManager** interface, and click **OK**.
 - c. Right-click the **MapToRelease** map and select **Generate Implementation**, clicking **OK** to accept the default folder. This opens the interface map editor.
 - d. Drag a line from **InputCriterion** to **releaseVehicle**, then click this line to show the parameter mappings. Draw lines between **RentalStatusActiveInput** and **ra**, and in the other direction between **releaseVehicleReturn** and **RentalStatusActiveOutput**.
 - e. Click the first **Move** operation, then the **Properties** tab. Change the Parameter Mapping Type to **Map**.
 - f. Click the **Details** tab, and set Business Object Map to **RentalAgmtToAgmt**.
 - g. Click the remaining **Move** operation in the map editor, then the **Properties** tab. Change the Parameter Mapping Type to **Map**.
 - h. Click the **Details** tab, and set Business Object Map to **AgmtToRentalAgmt**.
 - i. Save and close the interface map.

8. Create the MapToReturn interface map:
 - a. In the VehRentSvc assembly diagram, right-click the **MapToReturn** interface map and select **Add → Interface**. In the Add Interface dialog, select the **ReturnVehicleToSystem** interface and click **OK**.
 - b. Right-click the **MapToReturn** interface map again and select **Add → Reference**. In the Add Reference dialog, select the **PoolManager** interface, and click **OK**.
 - c. Right-click the **MapToReturn** map and select **Generate Implementation**, clicking **OK** to accept the default folder. The interface map editor opens.
 - d. Drag a line from **InputCriterion** to **returnVehicle**, then click this line to display the parameter mappings. Draw lines between **VehicleReturnInput** and **ra**, and in the other direction between **returnVehicleReturn** and **VehicleReturnOutput**.
 - e. Click the first **Move** operation, then the **Properties** tab. Change the Parameter Mapping Type to **Map**.
 - f. Click the **Details** tab, and set Business Object Map to **RentalAgmtToAgmt**.
 - g. Click the remaining **Move** operation in the map editor, then the **Properties** tab. Change the Parameter Mapping Type to **Map**.
 - h. Click the **Details** tab, and set Business Object Map to **AgmtToRentalAgmt**.
 - i. Save and close the interface map.
 - j. Save and close the assembly diagram. You have completed the interface maps and data maps, and now can assemble the composite application.

12.2.6 Assemble components into the composite application

To assemble the composite application, create complete assembly diagrams for each of the newly created modules. Then link the main assembly diagram for the VehicleRent module to the services in the other modules.

1. In the Business Integration view, expand the **VehRentSvc** module, and double-click the **VehRentSvc** assembly diagram. At this stage, the diagram should resemble Figure 12-38 on page 303.

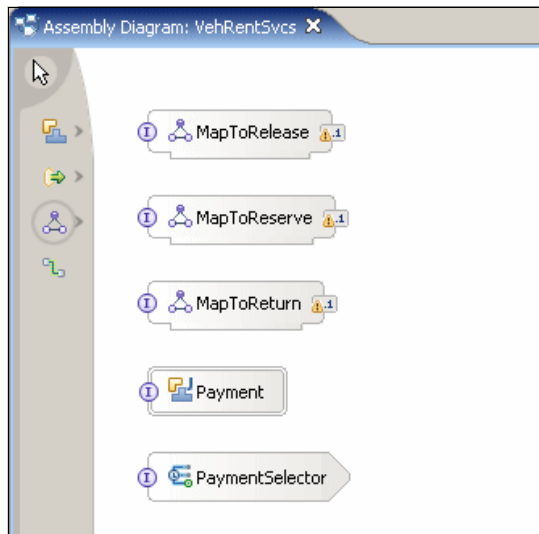


Figure 12-38 VehRentSvcs assembly diagram prior to assembly

2. Right-click any white space on the free-form surface and make sure that **Arrange Contents Automatically** is checked. This keeps the contents of the assembly diagram organized as you make changes.
3. Expand the **ReserveSvcs** module in the Business Integration view, then expand the **ReserveSvcs** assembly diagram. Under it will be the Export1 component export. Drag Export1 over to the free-form surface of the VehRentSvcs assembly diagram. In the Component Creation dialog, select **Import with SCA Binding** and click **OK**. An Import1 component will be created: rename this to ReserveSvcsImport.
4. Right-click **ReserveSvcsImport**, and select **Wire to Existing**. Watch carefully, and you will see that WebSphere Integration Developer will fairly intelligently find what components call this Import and wire the Import to those components (in this case our interface maps) automatically. Press Ctrl+S and you will see that this resolves the yellow exclamation points seen in Figure 12-38, which were the result of unresolved references.
5. Drag each of your components onto the VehRentSvcs diagram. Under the VehRentSvcs module, expand **Business Logic**:
 - a. Expand **Processes**, then drag the **CheckIfVehicleReturned** process to the VehRentSvcs assembly diagram.
 - b. Expand **Human Tasks**, and drag (one by one) all three human tasks to the VehRentSvcs assembly diagram.
 - c. Expand **Rule Groups**, and drag (one by one) both rule groups to the VehRentSvcs assembly diagram.

- For each of the components on the assembly diagram (except for Payment and ReserveSvcslImport), right-click and select **Export** → **SCA Binding**. This enables all of the components in VehRentSvcsl to be invoked as Imports from other modules, which will be useful when connecting the VehicleRent module to these components. At the end of this step there will be a total of 10 Exports on the assembly diagram. At this point, the assembly diagram will resemble Figure 12-39.

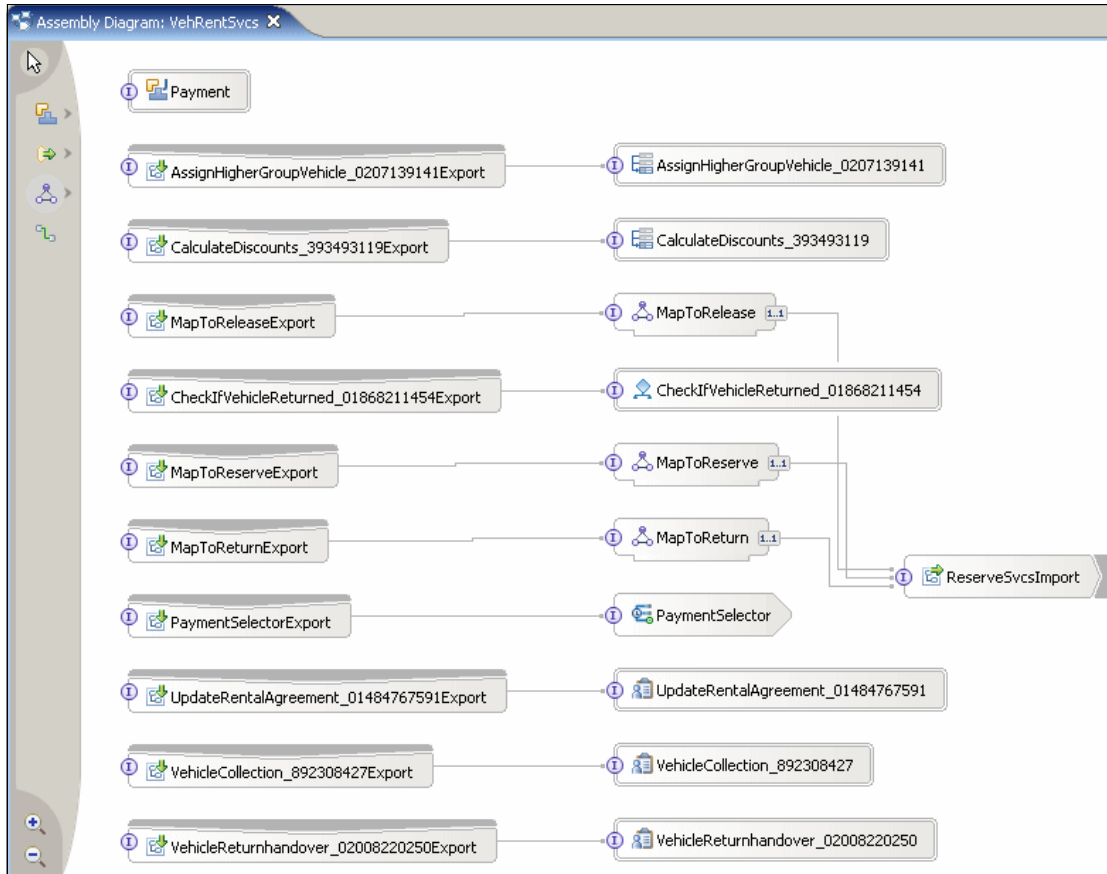


Figure 12-39 VehRentSvcsl assembly diagram

- Save and close the VehRentSvcsl assembly editor.
- Expand the **VehicleRent** module and double-click the **VehicleRent** assembly diagram. With your mouse, draw a box around all of the services that are wired to the process and press the Delete key. Now the assembly diagram should resemble Figure 12-40 on page 305.

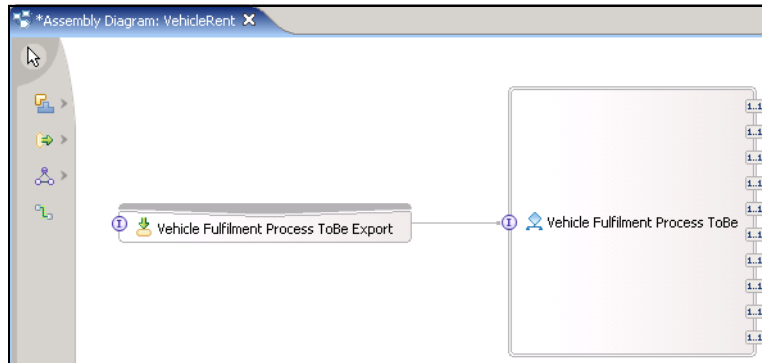


Figure 12-40 Modified VehicleRent assembly diagram

9. Under the VehRentSvcs module, expand the **VehRentSvcs** assembly diagram to expose the Exports underneath it. For each of the 10 Exports:
 - a. Drag the Export to the assembly diagram, and choose **Import with SCA Binding** to create an Import component.
 - b. Rename the default name to an appropriate name matching the Export. For instance, when dragging the AssignHigherGroupVehicleExport component onto the surface, an Import1 will be created. Rename Import1 to AssignHigherGroupVehicleImport.
 - c. Repeat for the remaining Exports.
10. Wire all references in the Process component to the newly created Imports. Right-click **Vehicle Fulfilment Process ToBe** and select **Wire (Advanced)**. The Advanced Wiring dialog appears. Use the settings in Table 12-3 to wire each source to a target. For example, select the **CalculateDiscountsPartner** source, and check the **CalculateDiscountsImport** target. Repeat this process to wire every source, then click **OK**.

Table 12-3 Advanced wiring

Wire source	Wire target
CalculateDiscountsPartner	CalculateDiscountsImport
Payment Partner	PaymentSelectorImport
RentalAgreementStatusSetToActivePartner	MapToReleaseImport
ReturnVehicleToSystemPartner	MapToReturnImport
VehicleCollectionPartner	VehicleCollectionImport
VehicleReturnhandoverPartner	VehicleReturnhandoverImport

Wire source	Wire target
WhileLoopReserveVehicleAssignHigherGroupVehiclePartner	AssignHigherGroupVehicleImport
WhileLoopReserveVehicleReserveVehicleInSystemPartner	MapToReserveImport
WhileLoopWaitonVehicleReturnCheckIfVehicleReturnedPartner	CheckIfVehicleReturnedImport
WhileLoopWaitonVehicleReturnUpdateRentalAgreementPartner	UpdateRentalAgreementImport

11. Press Ctrl+S to save the assembly diagram. It should resemble Figure 12-41. You should see no more red Xs either in your Business Integration view or in the Problems view.

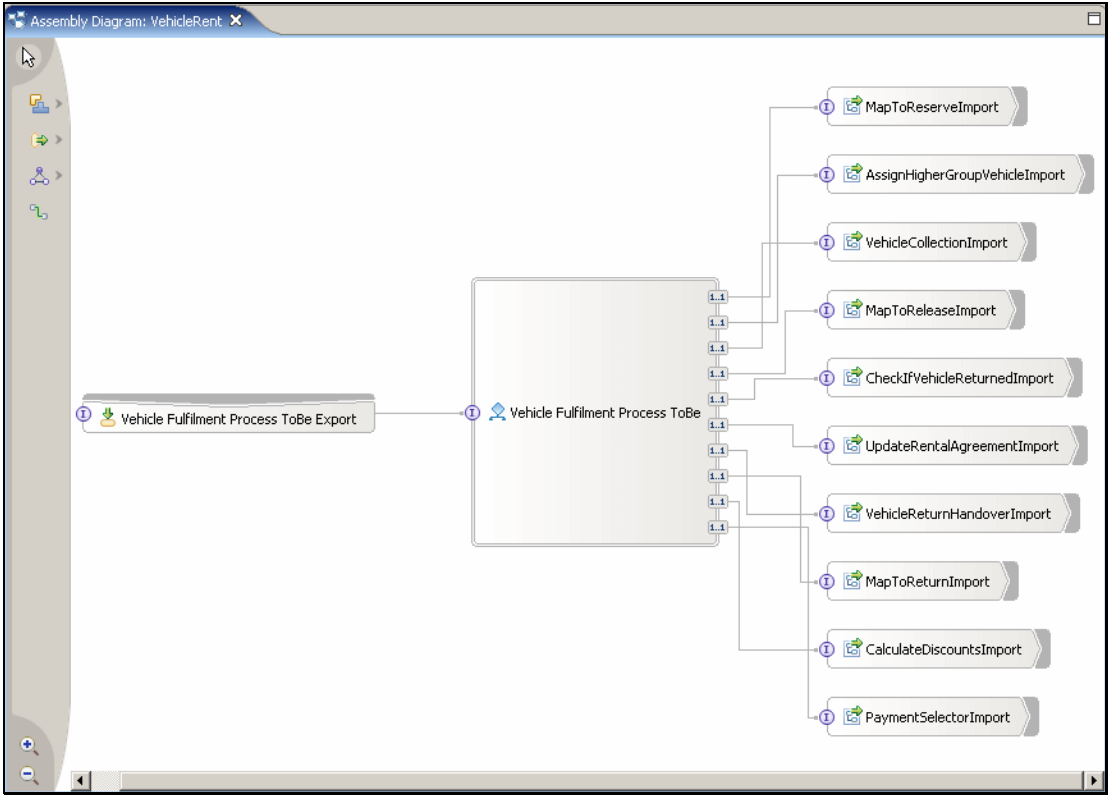


Figure 12-41 Completed VehicleRent assembly diagram

The composite application is completely assembled. Now, test the application.

12.3 Test and deploy the composite application

This section describes how to configure a test environment to test the composite application developed in this chapter. It contains the following sections:

- ▶ Setting up the test environment
- ▶ Configuring security in the test environment
- ▶ Unit-testing components in the composite application
- ▶ Exporting deployable artifacts for WebSphere Process Server

12.3.1 Setting up the test environment

WebSphere Integration Developer supplies a WebSphere Process Server unit test environment. This section outlines a few more steps prior to configuring the test environment for security. This enables you to control who can perform the human tasks involved in vehicle collection, vehicle return, and rental agreement update.

Authorization and authentication is a mandatory element in any Web application. WebSphere Process Server provides three different approaches to secure your server and application:

- ▶ Local OS (Operating System)

Local OS user registries are part of the underlying operating system that application servers are running on. Local OS user registries are always available, and are typically the simplest way to configure users because you have already set them up to log in to your machine.

- ▶ LDAP (Lightweight Directory Access Protocol)

When working with multiple servers, such as clustering and high availability, and with multiple machines, you will probably run into limitations using a single machine's local operating system user registry, and have to move up to a directory server using LDAP for storing user and group information.

- ▶ Custom user registry

A third option is a user registry that is not one of the operating system user registries supported by WebSphere Process Server and is not an LDAP user registry. This typically requires a limited software development effort to complete.

We will explore the custom user registry option to implement security for the human task application. We use the sample file-based registry (FileRegistrySample.java) provided in the Information Center for WebSphere Application Server V6. This enables us to simulate human task functionality

simply and easily in our development environment without having to set up a separate LDAP server or administer OS-level users and groups.

Define groups and users

To implement security, start by specifying groups and users. Our application requires a Branch Agent and a Branch Vehicle Agent.

1. Define the users shown in Table 12-4 on your local operating system. Set the passwords for each user to be the same as the user name. For example, the password for user wid should be wid.
2. Define the groups shown in Table 12-4 on your local operating system, and assign users to these groups as shown.

Note: Recall from earlier in the chapter that we are enabling roles (Branch Agent, Branch Vehicle Agent) as groups in our test scenario. This allows us to use roles for human tasks as our business model required. Using a simple file-based registry that emulates roles as groups makes our testing easier.

Table 12-4 Groups and users for human task security

Group	Users
Administrators	wid
Branch Agent	agent1
Branch Vehicle Agent	agent2

User registry and staff configuration files

Our implementation uses two property files and a Java archive for the file-based user registry. It also uses a customized XSL file for mapping roles to groups.

1. Locate the groups.props and users.props file in the \WIDResources directory of the additional material supplied with this book. Copy these two properties files into the following directory:
`<WID-HOME>/pf/wps/UserCustomRegistry/`
`<WID-HOME>` is the installation directory of WebSphere Integration Developer. You might have to create the UserCustomRegistry subdirectory.
2. FileRegistrySample.jar is a Java program that contains the code to read the property files and implement all of the required methods for the file-based registry. It is also located in the \WIDResources directory of the additional material supplied with this book. Copy it into the following directory:
`<WID-HOME>/runtimes/bi_v6/lib/ext`

3. UserRegistryTransformationwithRoleMembers.xml is an XSL file that enables Roles to be mapped to groups, that will be configured into the Staff service to support human tasks. It is also located in the \WID\Resources directory of the additional material supplied with this redbook. Follow these steps:
 - a. Navigate to the following directory:
`<WID-HOME>/runtimes/bi_v6/ProcessChoreographer/Staff`
 - b. Rename UserRegistryTransformation.xml to
`UserRegistryTransformation.xml.bak`
 - c. Copy UserRegistryTransformationwithRoleMembers.xml into this directory and rename it to `UserRegistryTransformation.xml`.

12.3.2 Configuring security in the test environment

These steps enable security in the WebSphere Process Server test environment:

1. Start the server (unit-test environment).
2. Define security using a user registry.
3. Enable global security.
4. Install the file transfer utility for security (to see the console).
5. Specify security in Integration Developer.
6. Specify a Cloudscape datasource.

Start the server

Start the unit test environment as follows:

1. In the Business Integration perspective, navigate to the bottom window and click the **Servers** tab.
2. Right-click the **WebSphere Process Server v6.0** entry and select **Start**.
3. Wait a minute or so for the server to start. Monitor the console view and wait until this message appears:

```
Server server1 open for e-business
```

Define security using a user registry

To define security in the unit test environment:

1. With the server started, open the administrative console by right-clicking the server in the Servers view and selecting **Run administrative console**.
2. Log in with any user ID.
3. Expand **Security** → **Global Security**.

4. Under Authentication (right side) expand **JAAS Configuration**, and select **J2C Authentication data**. You should see at least three alias entries with a user ID of wid (Figure 12-42).

Each of these entries uses wid for both user ID and password (encrypted).

Global security > J2EE Connector Architecture (J2C) authentication data entries			
Specifies a list of user IDs and passwords for Java 2 connector security to use.			
Preferences			
<input type="button" value="New"/> <input type="button" value="Delete"/>			
<input type="button" value="Add"/> <input type="button" value="Remove"/> <input type="button" value="Up"/> <input type="button" value="Down"/>			
Select	Alias	User ID	Description
<input type="checkbox"/>	SCA Auth Alias	wid	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	widCell/BPEAuthDataAliasJMS_widNode_server1	wid	Authorization Alias for Process Choreographer
<input type="checkbox"/>	widCell/widNode/server1/EventAuthDataAliasCloudScape	none	CloudScape authentication alias for the Common Event Infrastructure
<input type="checkbox"/>	widNode/CommonEventInfrastructureJMSAuthAlias	wid	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
Total 4			

Figure 12-42 J2C authentication data entries

Remember that the user ID wid was defined in users.props with the password wid. If you use a different password you must update the three aliases with it.

5. Click the **Global security** link to return to the previous page. Under User registries, click **Custom**. In the custom user registry screen, follow these steps (Figure 12-43 on page 311):
 - a. Enter wid as user ID and password.
 - b. The Custom registry class name by default points to our program name com.ibm.websphere.security.FileRegistrySample, so leave this setting.
 - c. Select **Ignore case for authorization**.
 - d. Click **Apply**.

Figure 12-43 Custom registry definition

6. Click **Custom properties**, and add two custom properties (Figure 12-44):
 - a. Click **New**, make the following entries, then click **OK**:

Name	groupsFile
Value	\${USER_INSTALL_ROOT}/UserCustomRegistry/groups.props
 - b. Click **New** again, make the following entries, then click **OK**:

Name	usersFile
Value	\${USER_INSTALL_ROOT}/UserCustomRegistry/users.props

Select	Name	Value	Description
<input type="checkbox"/>	groupsFile	\${USER_INSTALL_ROOT}/UserCustomRegistry/groups.props	Groups
<input type="checkbox"/>	usersFile	\${USER_INSTALL_ROOT}/UserCustomRegistry/users.props	Users

Total 2

Figure 12-44 Custom registry properties

7. Click the **Global security** link, then select **Authentication mechanism** → **LTPA** under Authentication (right side). Perform the following steps:
 - a. Enter wid in both password fields and click **Apply** (Figure 12-45 on page 312).

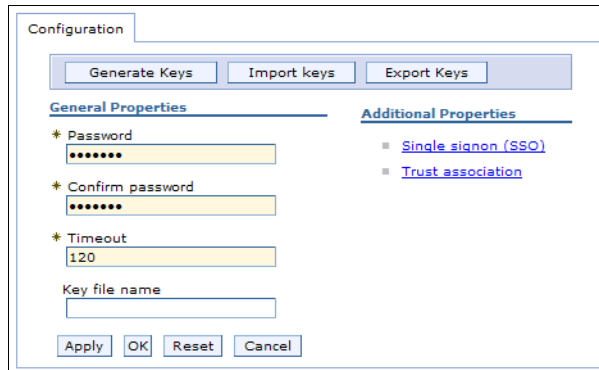


Figure 12-45 Authentication using LTPA

- b. Select **Single signon (SSO)**, select **Enabled** (the default), and click **OK**.

Enable global security

To run the server with security enabled (Figure 12-46 on page 313):

1. Select **Security** → **Global security**.
2. Select **Enable global security**, and deselect **Enforce Java 2 security**.
3. For Active authentication mechanism, select **Lightweight Third Party Authentication (LTPA)**.
4. For Active user registry, select **Custom user registry**.
5. Click **OK**.

The screenshot shows the 'Configuration' console with the 'General Properties' tab selected. The 'Enable global security' checkbox is checked. Other settings include 'Active protocol' set to 'CSI and SAS', 'Active authentication mechanism' set to 'Lightweight Third Party Authentication (LTPA)', and 'Active user registry' set to 'Custom user registry'. The right sidebar shows links for 'User registries', 'Authentication', 'Authorization', and 'Additional Properties'.

Figure 12-46 Enable global security

6. Click **Save**, and click **Logout** in the administrative console.

Redeploy the file transfer utility for security

When security is enabled, a special file transfer utility that uses security must be used by WebSphere Integration Developer to retrieve the console. To redeploy this utility, run a JACL script provided by the server:

1. Open a command window in the <WID-HOME>/runtimes/bi_v6/bin directory.
2. Run the following command:

```
wsadmin -profile redeployFileTransfer.jacl -lang jacl
        -c "fileTransferAuthenticationOn widCell widNode server1"
```

Example 12-1 on page 313 shows the expected output.

Example 12-1 Expected output from the file transfer utility redeploy

```
WASX7209I: Connected to process "server1" on node widNode using SOAP
...;
The type of process is: UnManagedProcess
Uninstall filetransfer -cell widCell -node widNode -server server1
ADMA5017I: Uninstallation of filetransfer started.
ADMA5011I: The cleanup ... for application filetransfer is complete.
ADMA5106I: Application filetransfer uninstalled successfully.
```

```

Install C:\WID601\runtimes\bi_v6\systemApps/filetransferSecured.ear
-cell widCell -node widNode -server server1 -appname filetransfer
-usedefaultbindings -nocreateMBeansForResources
ADMA5016I: Installation of filetransfer started.
ADMA5011I: The cleanup ... for application filetransfer is complete.
ADMA5013I: Application filetransfer installed successfully.

```

Specify security in WebSphere Integration Developer

Because the server is started and stopped from Integration Developer, we have to specify the user ID and password in the server configuration:

1. In the Servers view, double-click the server to open the server configuration.
2. Select **SOAP** as the connection type, and ensure that the port is 8880.
3. Expand **Security**. Select **Security is enabled on this server**. For user ID and password, enter wid or whatever you specified in the server custom registry.

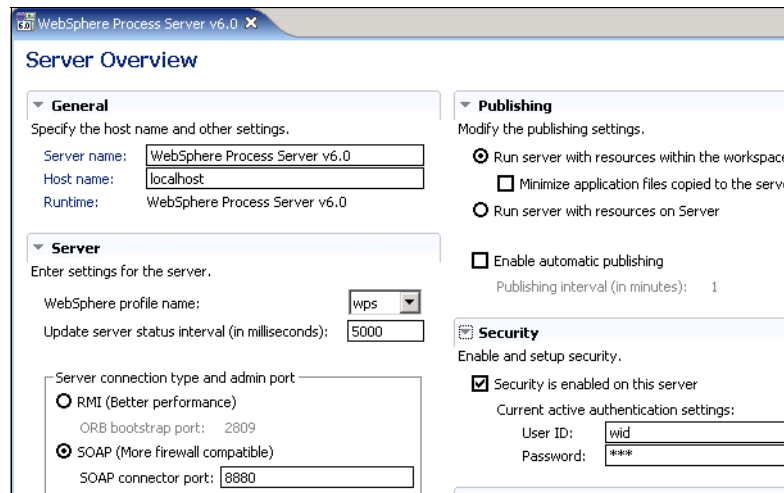


Figure 12-47 Server configuration settings

4. Save and close the configuration.

Specify the Cloudscape datasource in the test environment

One more step to take to prepare the test environment: Create a Cloudscape JDBC datasource, which is needed by the ReserveSvc module, in WebSphere Process Server:

1. The Cloudscape database contents are stored in a file system structure. Copy the RESERVE directory (which contains our Cloudscape database) from the

\WID\Database directory of the additional material supplied with this book to the <WID-HOME>/runtimes/bi_v6/cloudscape/databases directory.

2. In the Servers view of WebSphere Integration Developer, right-click the server and click **Run administrative console** (if it is not already running). Log in and expand **Resources**, then click **JDBC Providers**, then **Cloudscape JDBC Provider**, then click the **Data Sources** link.
3. Click **New** to create a new datasource, enter the values in Table 12-5, and click **OK**.

Table 12-5 Parameters for RESERVE datasource

Parameter	Value
Name	RESERVE
JNDI Name	jdbc/reserve
Database name	\${WAS_INSTALL_ROOT}/cloudscape/databases/RESERVE

4. Save your changes.
5. In the Data sources page, check the RESERVE data source and click **Test Connection**. A message should show that the test connection completed successfully. If it returns an error, be sure that the Database name parameter matches the full directory where you copied the RESERVE database.
6. Log out and close the administrative console.

Restart the server

The final stage before testing is to restart the server to apply the security changes. Follow these steps:

1. In the Servers view, right-click the server and select **Restart** → **Start**.
2. When the server has restarted, right-click the server again and select **Run administrative console**. Specify wid for user id and password. If you can log in, you have successfully applied security and can proceed to unit testing.

12.3.3 Unit-test components in the composite application

This section tests the components that were built earlier in this chapter by using the component testing feature in WebSphere Integration Developer. Because everything is component-based in this environment, it is easy to test components either in isolation or in combination. The first test is on an interface map that connects to a Web service. Then the vehicle rental process itself is tested using all of the services that it relies on (that is, testing the composite application

end-to-end). This second test includes interaction with the process using the BPC (Business Process Choreographer) Explorer to test the human tasks.

Before the first test, deploy the developed projects onto the unit-test environment:

1. In the Servers tab, right-click the server and click **Add and remove projects**.
2. In the Add and Remove Projects dialog you can deploy projects to the server. The left side of this window shows all of the projects that have been created. To add them to the server, click **Add All**, then click **Finish** (Figure 12-48).

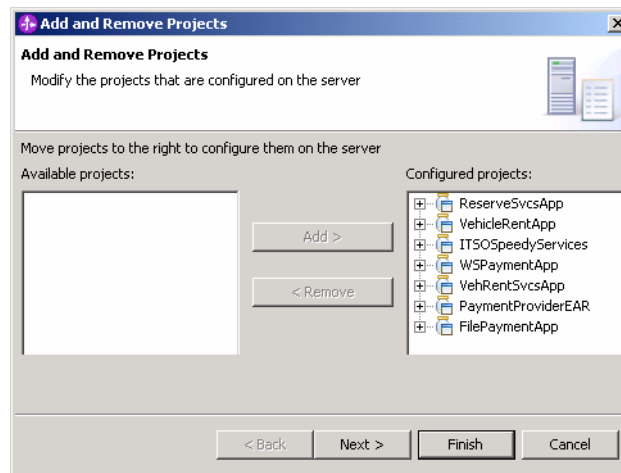


Figure 12-48 Add and Remove Projects dialog

3. Wait a few minutes while the projects are deployed into the test environment and started.

Testing a map using component testing

One of the most powerful features of WebSphere Integration Developer is component testing. Because everything in it is integrated using SCA, everything is an SCA component. This makes testing easy, because you can test components in isolation, then two components working together, then build up to an entire composite application made of many components. We exercise this by testing MapToReserve, which tests both the MapToReserve interface map itself and the ReserveSvc Web service.

1. In the Business Integration view, expand the **VehicleRent** module and double-click the **VehicleRent** assembly diagram.
2. Right-click the **MapToReserveImport** element in the diagram and select **Test Component**. This Import connects to the MapToReserveExport in the VehSvc module, which connects to the MapToReserve interface map, which

connects to the ReserveSvcImport, which ultimately connects to the reserveVehicle operation in the EJB Web service imported into the ReserveSvc module.

3. The component test view appears. Click the **Configurations** tab, which offers the choice components that will be emulated rather than tested. Several emulators are configured by default. If the component you want to test has an emulator configured, it will not actually be invoked, only emulated. Sometimes this helps in isolating down to particular components and testing them separately from a component that depends on them.

In this test, do not use any of the default emulators. Select all of the emulators and click **Remove** so that none remain (Figure 12-49).

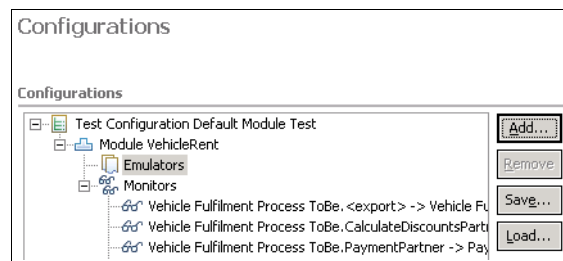


Figure 12-49 Removal of all emulators

4. Click the **Events** tab. There is a single event called Invoke, and under Detailed Properties you see the details of what you are testing as well as the data objects that you are providing as input. This component takes a single RentalAgreement object. The editor allows you to fill out all data fields for a Rental Agreement, but for this test you only need to enter the following data (circled in Figure 12-50 on page 318), then click **Continue**:
 - a. Set VehicleGroupRequested to A.
 - b. Set VehicleDetails.InventoryID to null by clicking in the value field for this parameter and selecting **<null>** from the pull-down menu.

Configuration: Default Module Test

Module: VehicleRent

Component: MapToReserveImport

Interface: ReserveVehicleInSystem

Operation: InputCriterion

Initial request parameters

Name	Type	Value
VehicleGroupRequ...	string	A
VehicleGroupRecei...	string	
PickupDateTime	dateTime	2002-01-01T11:00:00
PlannedReturnDat...	dateTime	2002-01-01T11:00:00
ActualReturnDateT...	dateTime	2002-01-01T11:00:00
EstimatedCost	double	0
OverdueStatus	string	
Discount	string	
RentalStatus	string	
ActualCost	double	0
+ DriverDetails	Driver	
+ BranchDetails	Branch	
- VehicleDetails	Vehicle	
GroupID	string	
ModelName	string	
Manufacturer	string	
EngineSize	string	
FuelType	string	
TransmissionType	string	
Available	boolean	false
InventoryID	string	<null>
VehicleStatus	string	

Data Pool

Continue

Figure 12-50 Initial request parameters

5. The Deployment Location dialog appears. Click **Use this as the default and do not ask again**, then click **Finish** to accept the default WebSphere Process Server v6.0 server to deploy the component test.
6. In the User Login dialog, enter wid for user ID and password. Click **OK**.
7. The component test starts to run, and the view in the component test editor shows an Invoke and a Return on the left, and a changed RentalAgreement on the right, with a ReservationID, a VehicleGroupReceived type, and several details in VehicleDetails including manufacturer, model, availability, and inventoryID. This should resemble Figure 12-51 on page 319.

Events

- Invoke (MapToReserveImport:InputCriterion)
 - Started
 - Invoke (MapToReserveImport:InputCriterion)
 - Return (MapToReserveImport:InputCriterion)
 - Stopped

Detailed Properties

Module: [VehicleRent](#)
 Component: [MapToReserveImport](#)
 Interface: [ReserveVehicleInSystem](#)
 Operation: [InputCriterion](#)

Return parameters:

Name	Type	Value
ReserveVehicle...	RentalAgreement	
ReservationID	String	30550
VehicleGrou...	String	A
VehicleGrou...	String	A
PickupDateT...	Date	2002-01-01T11:...
PlannedRet...	Date	2002-01-01T11:...
ActualRetur...	Date	2002-01-01T11:...
EstimatedCost	Double	0.0
ActualCost	Double	0.0
OverdueSta...	String	<null>
RentalStatus	String	<null>
Discount	String	<null>
DriverDetails	Driver	
DriverName	String	<null>
LicenseID	String	<null>
LicenseE...	Date	<null>
DateOfBirth	Date	<null>
CreditCa...	String	<null>
DebitCar...	String	<null>
RiskLevel	String	<null>
LoyaltyCl...	String	<null>
BranchDetails	Branch	<null>
VehicleDetails	Vehicle	
GroupID	String	<null>
Model	String	FIRESTAR
Manufact...	String	PAGODA

Figure 12-51 Component test of MapToReserveImport component

This completes the steps of a unit test of an interface map and a Web service component, which revealed whether the interface map and its associated data map handled mapping between the invocation interface (from the process) and the implementation interface (on the Web service itself). It also showed whether the Web service could be invoked successfully through its SCA interface. The component test accomplished both of these, and enabled us to reserve a car by passing in a RentalAgreement as input, and in turn receiving an updated RentalAgreement showing information about the reserved vehicle.

Testing the process using component test

This section takes component testing further by testing the entire process as a whole, in addition to all of the components we have created to work with the

process. The two testing paths request a type B vehicle, which will be available for the first request, but unavailable on the second request (because the provided Cloudscape sample database has six B cars in inventory but only one in AVAILABLE state). This second path tests the functionality of the AssignHigherGroupVehicle business rule.

In addition to using the component test again, we use the BPC Explorer to exercise the human tasks. Therefore, the paths test all of our functionality, including the maps, Web services, business rules, human tasks, adapter module, and selector. Testing will extend further in the next chapter by dynamically changing the business rule values and modifying the target service for the selector.

Note: When testing the process, you will make changes to the underlying application database. For example, say the database has only one available car in class B. If you run a test that results in a rental of a class B car and that car is not returned, the next time you run the same test the user might receive an upgrade to a class C car because all class B cars are out of stock.

If you wish to reset the application database to its original state:

1. Stop the test server.
2. Locate the RESERVE Cloudscape database at
<WID-HOME>/runtimes/bi_v6/cloudscape/databases and delete it.
3. Find the RESERVE Cloudscape database in the \WID\Database directory of the additional material and copy it to
<WID-HOME>/runtimes/bi_v6/cloudscape/databases.
4. Start the server.

Straight-through path with no vehicle reassignment

Perform the following test:

1. In the VehicleRent assembly diagram, right-click the **Vehicle Fulfillment Process To Be** and click **Test Component** to open the Component Test view.
2. Click the **Configurations** tab. This test removes all but for one of the emulators. Remove all emulators except for CheckIfVehicleReturnedImport and click back to the **Events** tab when you are finished (Figure 12-52 on page 321).

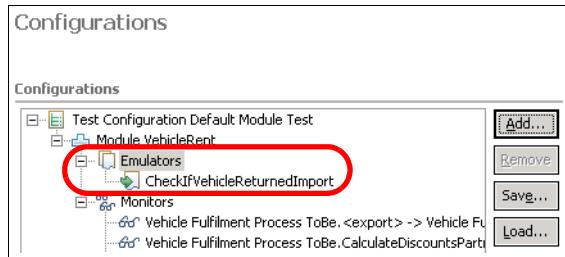


Figure 12-52 Removal of all emulators except *CheckIfVehicleReturnedImport*

Note: We are using an emulator to make testing a bit easier. Recall that *CheckIfVehicleReturned* is the subprocess that restarts the main process. There are several ways to test this scenario. You could use two component test instances: one for the main process and one for the subprocess. You could use a component test instance and the BPC Explorer the same way, or you could use an emulator to simulate the results of the subprocess within the main process being tested in Component Test.

There are good reasons for each of these types of testing, but for this test we use the emulator approach.

3. Enter the values shown in Table 12-6 and click **Continue**. Enter user ID *wid* and password *wid* as deployment credentials and click **OK** to start the component test.

Table 12-6 Input values to start a process instance

Field	Value	Constraint
VehicleGroupRequested	B	A, B or C (the three types of vehicles in the Reserve database).
RiskLevel	low	Low, middle or high. This value is used in the business rule when an upgrade is needed.
InventoryID	null (using pull-down menu)	Actual InventoryID will be returned by PoolManager Web Service.

4. Events appear in the Component Test. A response from MapToReserveImport is followed by a request to VehicleCollection, similar to Figure 12-53.

The screenshot displays the Eclipse IDE interface for a component test. The 'Events' window on the left shows a sequence of events: Started, Invoke (Vehicle Fulfillment Process ToBe:InputCriterion), Request (Vehicle Fulfillment Process ToBe --> Vehicle Fulfillment Process ToBe), Response (Vehicle Fulfillment Process ToBe <-- MapToReserveImport), and Request (Vehicle Fulfillment Process ToBe --> VehicleCollection). The 'Response' event is highlighted. The 'Detailed Properties' pane on the right shows the response parameters for the 'Response' event, including ReservationID, VehicleGroupR..., PickupDateTime, PlannedReturn..., ActualReturn..., EstimatedCost, ActualCost, OverdueStatus, RentalStatus, Discount, DriverDetails, BranchDetails, and VehicleDetails.

Name	Type	Value
ReserveVehiclein...	RentalAgreement	
ReservationID	String	98218
VehicleGroupR...	String	B
VehicleGroupR...	String	B
PickupDateTime	Date	2002-01-01T11:00.
PlannedReturn...	Date	2002-01-01T11:00.
ActualReturn...	Date	2002-01-01T11:00.
EstimatedCost	Double	0.0
ActualCost	Double	0.0
OverdueStatus	String	<null>
RentalStatus	String	<null>
Discount	String	<null>
DriverDetails	Driver	
DriverName	String	<null>
LicenseID	String	<null>
LicenseExpi...	Date	<null>
DateOfBirth	Date	<null>
CreditCard...	String	<null>
DebitCardN...	String	<null>
RiskLevel	String	low
LoyaltyClubID	String	<null>
BranchDetails	Branch	<null>
VehicleDetails	Vehicle	
GroupID	String	<null>
Model	String	CENTURION
Manufacturer	String	AUBURN
EngineSize	String	<null>

Figure 12-53 Completed vehicle reservation in first path of main process component test

The results of the test so far show that the vehicle has been reserved (which is reflected in the database), a reservation ID has been returned, both the requested and received vehicle types are returned, and vehicle information is returned including Manufacturer, Model, and Inventory ID.

The next step in the process is a human task: The vehicle rental agent checks out the vehicle to the customer at the rental counter or rental gate. This step simulates the actions of the vehicle rental agent using the BPC Explorer.

- Click the **Servers** tab at the bottom of the screen, right-click **WebSphere Process Server v6.0**, and select **Launch** → **BPC Explorer**.

Note: The BPC Explorer is a console that is useful for a number of things. First, as shown here, it can be used by human beings in a workflow to complete task items. In a fully implemented workflow application, a more customized BPC Explorer provides an employee-appropriate interface.

Another important way to use BPC Explorer is for process administration, as explained in Chapter 13, “Deploy with WebSphere Process Server” on page 333.

- Log in to BPC Explorer using user ID agent2 and password agent2. (This is the ID for the Branch Vehicle Agent Role.)
- A task displays in the My Tasks view. Click the check box next to this task and click **Work On** (Figure 12-54).

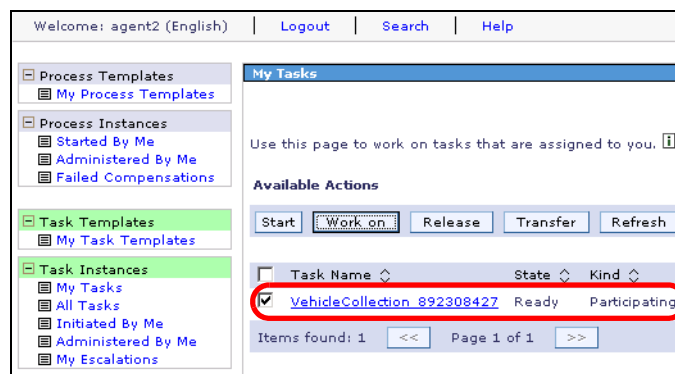


Figure 12-54 My Tasks view in the BPC Explorer

- The next window shows an input message presented to the branch vehicle agent, and an empty output message for the agent to fill in. In the output message VehicleCollectionOutput, fill in the following fields as shown in Table 12-7 on page 324, then scroll back to the top and click **Complete**.

Table 12-7 Input values to complete VehicleCollection task

Field	How to locate the value to use
ReservationID	Locate the ReservationID value in the input message VehicleCollectionInput. Use this same value in the output message.
PlannedReturnDateTime	This value should be a date/time that is later than the current date/time.
InventoryID	Locate the InventoryID value in the input message VehicleCollectionInput. Use this same value in the output message.

Note: The BPC Explorer does not copy task's input message into task's output message automatically. Values must be entered manually into the output message even though they are same as the values of the input message. In a fully implemented application, a customized Web client would copy the necessary input message values into the output message automatically, and would use many of the other field values in this message that are not used in our sample. For a good example of a customized Web client, refer to *WebSphere Process Integration V6: Business Process Management Modeling through Monitoring*, SG24-7148.

- The task disappears from the BPC Explorer console for the Branch Vehicle Agent, meaning that it has been completed. The process receives the output message from this human task and resumes.

Back in the Component Test window, you see the process resuming, showing further events including an indication that the vehicle has now been released (customer has exited the rental gate). The process pauses at the CheckIfVehicleReturnedImport process, which is being emulated, as shown in Figure 12-55 on page 325.

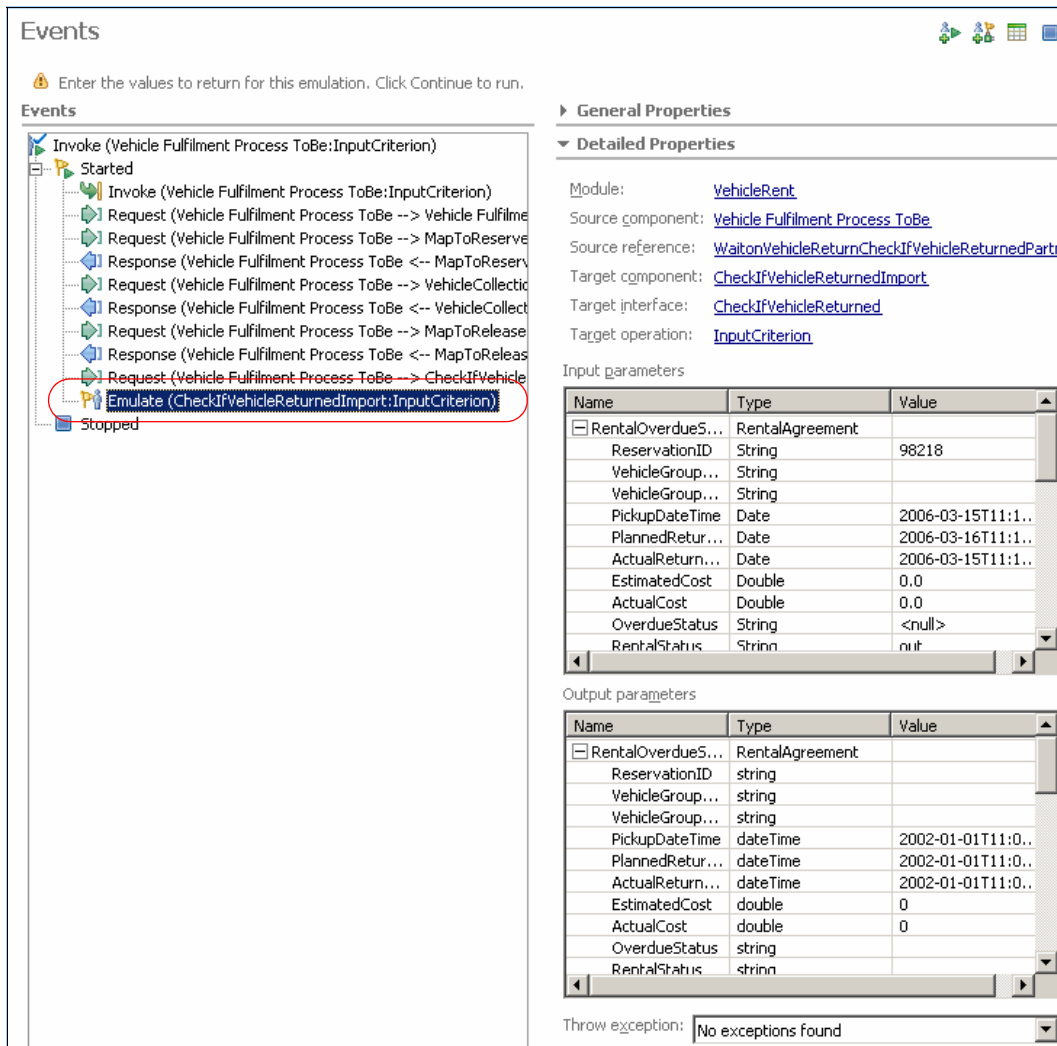


Figure 12-55 Completed vehicle release in first path of main process component test

10. Now emulate the vehicle return event: In the Output parameters window, enter the values shown in Table 12-8 on page 326, then click **Continue**.

Table 12-8 Input values to emulate vehicle return event

Field	How to locate the value to use
ReservationID	This value should be same as the ReservationID in the Input parameters window.
InventoryID	This value should be same as the InventoryID in the Input parameters window.

11. The process resumes as a result of the vehicle return event, and the next task is the VehicleReturnHandover human task. Click the **BPC Explorer** window again. You are already logged in as agent2, so click **My Tasks** to see this human task appear in the view. As before, select the check box for the task and click **Work On**.

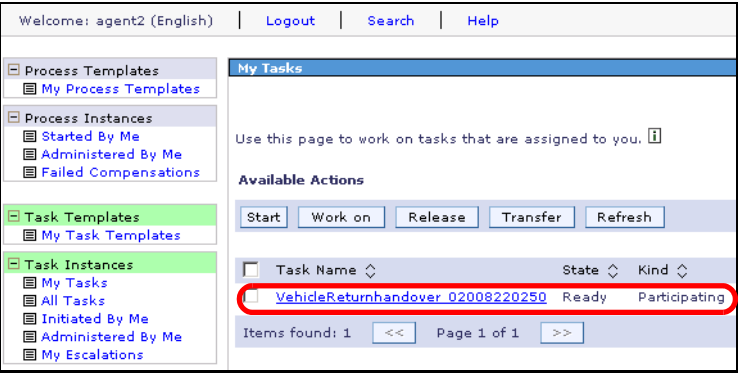


Figure 12-56 My Tasks view in the BPC Explorer

12. Enter the values in Table 12-9 in the VehicleHandoverOutput output message to complete this human task, then click **Complete**.

Table 12-9 Input values to complete VehicleReturnHandover task

Field	How to locate the value to use
ReservationID	Locate the ReservationID value in the input message VehicleHandoverInput. Use this same value in the output message.
VehicleGroupRequested	Enter B
PickupDateTime	Should be a date/time value earlier than ActualReturnDateTime.
ActualReturnDateTime	Should be a date/time value later than PickupDateTime.

Field	How to locate the value to use
LoyaltyClubNumber	Enter A001
InventoryID	Locate the InventoryID value in the input message VehicleHandoverInput. Use this same value in the output message.

13. Back in the Component Test view, wait a few moments as several events fire in succession and the process runs to completion, resembling Figure 12-57.

The screenshot shows the 'Events' window with a list of events on the left and 'General Properties' and 'Detailed Properties' on the right. The process flow includes events like 'Invoke (Vehicle Fulfilment Process ToBe:InputCriterion)', 'Request (Vehicle Fulfilment Process ToBe --> Vehicle Fulfilment Process ToBe)', 'Response (Vehicle Fulfilment Process ToBe <-- MapToReserveImport:Import)', 'Request (Vehicle Fulfilment Process ToBe --> VehicleCollectionImport:Import)', 'Response (Vehicle Fulfilment Process ToBe <-- VehicleCollectionImport:Import)', 'Request (Vehicle Fulfilment Process ToBe --> MapToReleaseImport:Import)', 'Response (Vehicle Fulfilment Process ToBe <-- MapToReleaseImport:Import)', 'Request (Vehicle Fulfilment Process ToBe --> CheckIfVehicleReturnedImport:Import)', 'Emulate (CheckIfVehicleReturnedImport:InputCriterion)', 'Response (Vehicle Fulfilment Process ToBe <-- CheckIfVehicleReturnedImport:Import)', 'Request (Vehicle Fulfilment Process ToBe --> VehicleReturnHandoverImport:Import)', 'Response (Vehicle Fulfilment Process ToBe <-- VehicleReturnHandoverImport:Import)', 'Request (Vehicle Fulfilment Process ToBe --> MapToReturnImport:Import)', 'Response (Vehicle Fulfilment Process ToBe <-- MapToReturnImport:Import)', 'Request (Vehicle Fulfilment Process ToBe --> CalculateDiscountsImport:Import)', 'Response (Vehicle Fulfilment Process ToBe <-- CalculateDiscountsImport:Import)', 'Request (Vehicle Fulfilment Process ToBe --> PaymentSelectorImport:Import)', and 'Response (Vehicle Fulfilment Process ToBe <-- PaymentSelectorImport:Import)'. The process is marked as 'Stopped'.

The 'Detailed Properties' section shows the following information:

- Module: [VehicleRent](#)
- Source component: [Vehicle Fulfilment Process ToBe](#)
- Source reference: [PaymentPartner](#)
- Target component: [PaymentSelectorImport](#)
- Target interface: [Payment](#)
- Target operation: [InputCriterion](#)

The 'Response parameters' table is as follows:

Name	Type	Value
PaymentOutput	RentalAgreement	
ReservationID	String	25540
VehicleGroupRequ...	String	B
VehicleGroupRecei...	String	
PickupDateTime	Date	2006-04-18T12:44:00
PlannedReturnDat...	Date	2006-04-19T12:44:00
ActualReturnDateT...	Date	2006-04-19T12:44:00
EstimatedCost	Double	0.0
OverdueStatus	String	<null>
Discount	String	<null>
RentalStatus	String	in
ActualCost	Double	50.0
DriverDetails	Driver	
DriverName	String	<null>
LicenseID	String	<null>
LicenseExpiration	Date	<null>

Figure 12-57 Completed entire process in first path of main process component test

The VehicleReturnHandover human task completes and indicates that the Branch Vehicle Agent has checked in the vehicle when the customer returned it. After this, the updated RentalAgreement is sent to the CalculateDiscounts business rule, which calculates the vehicle rate using a table of base rates plus a dynamic loyalty club calculation. This updates the RentalAgreement with the ActualCost and sends it to the Payment service, which simulates payment processing. The process is complete, showing how to quickly and easily test process components end to end using the Component Test feature in WebSphere Integration Developer.

Note: Every time you execute this process you are making changes to the underlying RESERVE Cloudscape database, even in the case of error.

For example, you might run three process instances that result in B class cars being rented, without these cars being returned. Therefore, the RESERVE database could run out of new cars available for rental. In this instance, stop the server, and replace the RESERVE database with the original one supplied with the additional material that accompanies this book.

Straight-through path with vehicle reassignment

This test performs many of the same steps a second time, but tests the functionality of the other business rule, AssignHigherVehicleGroup, by reserving the last available vehicle of type B, then attempting to reserve a vehicle of type B. This causes the vehicle reservation to return as unavailable, which fires the business rule to reassign the vehicle group based on the customer’s driving record (risk level). The reassigned vehicle group is then requested and the process continues to completion.

The database as shipped in the sample code directory contains only one available type B vehicle (all the rest of the type B vehicles are in reserved status). In the previous test we reserved the one available class B car, but then returned it. This time, mark it as reserved before proceeding.

- 1. In the VehicleRent assembly diagram, right-click the **MapToReserve** component and select **Test Component**. Enter B for VehicleGroupRequested and a null value (from the pull-down list) for InventoryID, and click **Continue**. Enter user ID wid and password wid to deploy and start the component test.
- 2. The return value from this component test is a reserved type B vehicle (Figure 12-58).

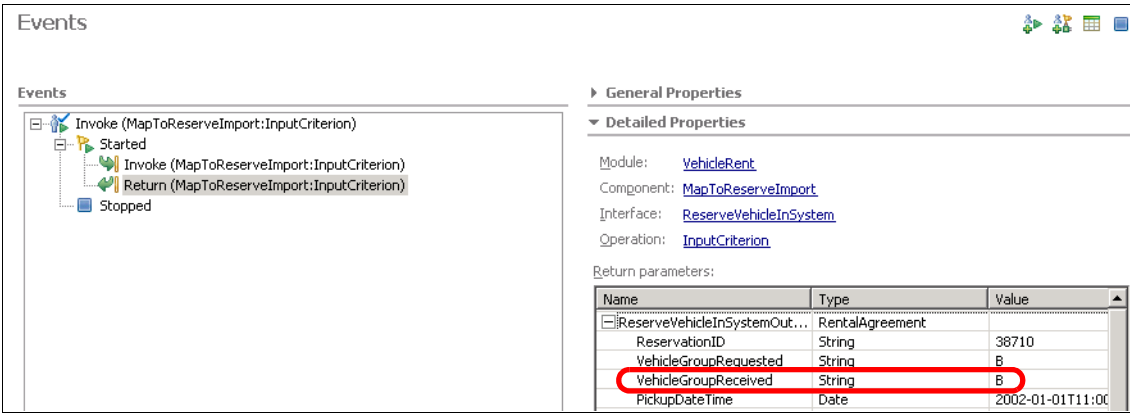


Figure 12-58 Reservation of a B class car

3. In the VehicleRent assembly diagram, right-click **Vehicle Fulfilment Process ToBe** and select **Test Component**.
4. Click the **Configurations** tab, remove all of the emulators except **CheckIfVehicleReturnedImport**, then click back on the **Events** tab.
5. In the Initial request parameters, enter the values shown in Table 12-10 then click **Continue**.

Table 12-10 Input values to start a process instance

Field	Value
VehicleGroupRequested	B
RiskLevel	low
InventoryID	null (using pull-down)

6. Events and messages flow through the process. This time you see a different path invoked. Examine the response value from the MapToReserveImport invocation (scroll down): this time VehicleDetails.Available is false, meaning that the requested vehicle was unavailable to reserve.

This caused the next component to fire, the AssignHigherVehicleGroupImport business rule. The business rule determines which vehicle group to request on behalf of the customer as an upgrade, based on driver risk and vehicle type requested. This business rule outputs a modified RentalAgreement with a different value for VehicleGroupRequested, which is passed back to the MapToReserveImport component again.

This time when MapToReserveImport is invoked, a type C vehicle is requested (based on the default values we entered previously in the business rule) and is reserved successfully. The process pauses at the VehicleCollection human task as before, ready for human intervention by our trusty branch vehicle agent. Your results should resemble Figure 12-59 on page 330.

Events

Invoke (Vehicle Fulfilment Process ToBe:InputCriterion)

- Started
- Invoke (Vehicle Fulfilment Process ToBe:InputCriterion)
- Request (Vehicle Fulfilment Process ToBe --> Vehicle Fulfilment Process ToBe)
- Request (Vehicle Fulfilment Process ToBe --> MapToReserve)
- Response (Vehicle Fulfilment Process ToBe <-- MapToReserve)
- Request (Vehicle Fulfilment Process ToBe --> AssignHigherGroup)
- Response (Vehicle Fulfilment Process ToBe <-- AssignHigherGroup)
- Request (Vehicle Fulfilment Process ToBe --> MapToReserve)
- Response (Vehicle Fulfilment Process ToBe <-- MapToReserve)
- Request (Vehicle Fulfilment Process ToBe --> VehicleCollectionImport)
- Stopped
- Stopped

General Properties

Detailed Properties

Module: [VehicleRent](#)

Source component: [Vehicle Fulfilment Process ToBe](#)

Source reference: [VehicleCollectionPartner](#)

Target component: [VehicleCollectionImport](#)

Target interface: [VehicleCollection](#)

Target operation: [InputCriterion](#)

Request parameters

Name	Type	Value
VehicleCollection...	RentalAgreement	
ReservationID	String	89874
VehicleGroup...	String	C
VehicleGroup...	String	C
PickupDateTime	Date	2002-01-01T11:0...
PlannedRetur...	Date	2002-01-01T11:0...
ActualReturn...	Date	2002-01-01T11:0...
EstimatedCost	Double	0.0
ActualCost	Double	0.0
OverdueStatus	String	<null>
RentalStatus	String	<null>
Discount	String	<null>
DriverDetails	Driver	
DriverName	String	<null>
LicenseID	String	<null>
LicenseEx...	Date	<null>
DateOfBirth	Date	<null>
CreditCard...	String	<null>
DebitCard...	String	<null>
RiskLevel	String	low
LoyaltyClu...	String	
BranchDetails	Branch	<null>
VehicleDetails	Vehicle	
GroupID	String	<null>
Model	String	LIMITLESS
Manufactu...	String	NEPTUNE

Figure 12-59 Tested vehicle group reassignment in second path of main process component test

- At this point, you may continue the test using the steps in the previous test path, or click the right-most blue button on the Component Test view to terminate the component test.

You have seen the powerful component testing capabilities of WebSphere Integration Developer, and how this simplifies your life as an integration developer. In addition to the capabilities that we have exercised in this chapter, the debugging capability enables you to debug through each of the component

types as they are executed using Component Test. Although this is outside the scope of this book, you can find more information about this feature in the WebSphere Business Process Management Version 6.0 Information Center at: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wbit.help.debug.doc/topics/tdebug.html>

Note: If you want to experiment more with testing this sample, remember that this has been a comparatively small sample Cloudscape database. If you test the process multiple times, you might end up reserving all available vehicles, in which case the vehicle reservation service would start throwing exceptions. You can easily reset the Cloudscape database to initial settings by deleting the existing Cloudscape database directory and unzipping the original downloaded sample database again into the same location. Because this is a file-based database, it will reset all of the values in the database to their provided defaults, in our case with all except one vehicle in Available status.

12.3.4 Export deployable artifacts for WebSphere Process Server

Before deploying your artifacts, you need to make a change to the human tasks in the VehRentSvcs module, because we have completed testing human tasks in this chapter. This saves you from having to configure a staff user registry in WebSphere Process Server.

Follow these steps:

1. Expand the **VehRentSvcs** module → **Business Logic** → **Human Tasks**.
2. For each of the three human tasks:
 - a. Double-click the human task.
 - b. Click **Potential Owner** in Receiver Settings and click the **Properties** tab to view the properties for this staff setting.
 - c. Change the Verb from Role Members to **Everybody** (Figure 12-60).

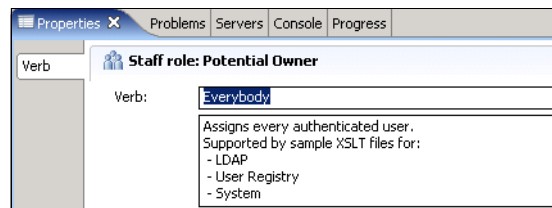


Figure 12-60 Everybody verb in Staff settings

- d. Save and close the editor.

3. Create your deployable artifacts as both ZIP files and EAR files, because both types of artifacts will be needed in the next chapter. We have chosen to separate the artifacts into multiple individual files in order to provide enhanced granularity and manageability of components in our environment. At the end of this section, you will have created 12 files for deployment to WebSphere Process Server.
 - a. For *each* zip file, click **File** → **Export** → **ZIP File**, then **Next** to start the export wizard, then select the projects to export, provide a filename, keep all defaults, and click **Finish**. Use the values in Table 12-11 to create these five ZIP files.

Table 12-11 Creating ZIP files for deployment to WebSphere Process Server

Projects to include	Filename for artifact
VehicleRent, VehicleRentLib	VehicleRent.zip
VehRentSvcs, VehicleRentLib	VehRentSvcs.zip
FilePayment, VehicleRentLib, CWYFF_FlatFile	FilePayment.zip
WSPayment, VehicleRentLib	WSPayment.zip
ReserveSvcs, VehicleRentLib	ReserveSvcs.zip

- b. For *each* EAR file, click **File** → **Export** → **EAR File**, then **Next**, then select the **EAR project**, provide a filename, keep all defaults, and click **Finish**. Use the values in Table 12-12 to create these seven EAR files.

Table 12-12 Creating EAR files for deployment to WebSphere Process Server

EAR project	Filename for artifact
FilePaymentApp	FilePaymentApp.ear
ITSOSpeedyServices	ITSOSpeedyServices.ear
PaymentProviderEAR	PaymentProvider.ear
ReserveSvcsApp	ReserveSvcsApp.ear
VehicleRentApp	VehicleRentApp.ear
VehRentSvcsApp	VehRentSvcsApp.ear
WSPaymentApp	WSPaymentApp.ear

After all 12 of these deployable artifacts are exported, you are finished with the integration developer tasks and can proceed to the next chapter.



Deploy with WebSphere Process Server

This chapter describes how to configure, install, run, and administrate the business process application built in Chapter 12, “Assemble with WebSphere Integration Developer” on page 263 using WebSphere Process Server.

We assume that you have WebSphere Process Server V6.0.1 installed, with a default profile created and running. This server instance should be separate from the server used in Chapter 12, “Assemble with WebSphere Integration Developer” on page 263.

This chapter covers the following topics:

- ▶ Configuring a server with data source and Business Rules Manager
- ▶ Installing the business process application
- ▶ Running the application using the Business Process Choreographer (BPC) Explorer
- ▶ Using the Business Rules Manager
- ▶ Administering a selector component

13.1 Configuring a server with data source and Business Rules Manager

This section configures WebSphere Process Server to use a Cloudscape database as a data source for the Web services application, then installs the Business Rules Manager for the administrator to update business rules for the business process application.

13.1.1 Configuring the server to use a data source

The Vehicle Reservation Web services module uses a Cloudscape database to maintain its reservation data. You have to create a data source for the application to connect to the database instance.

With WebSphere Process Server server running, follow these steps:

1. Locate the RESERVE database in the \WID\Database\RESERVE directory of the additional material supplied with this book. Copy this RESERVE directory to the following location:

<WPS_HOME>\profiles\ProcSrv01\databases

Note: Replace <WPS_HOME> with the home directory of your WebSphere Process Server installation. The instructions in this chapter assume that you are using the default profile name of ProcSrv01. If your profile name is different, use the appropriate path for your profile.

2. Open a Web browser and log in to the administrative console. The default URL for the administrative console is:
`http://localhost:9060/ibm/console`
3. In the administrative console, select **Resources** → **JDBC Providers** to list the Java Database Connectivity(JDBC) Providers that are already configured in WebSphere Process Server.
4. Select **Server : server1** as the scope of JDBC Provider. Click **Apply** to see the JDBC Providers in the scope of application server as shown in Figure 13-1 on page 335.

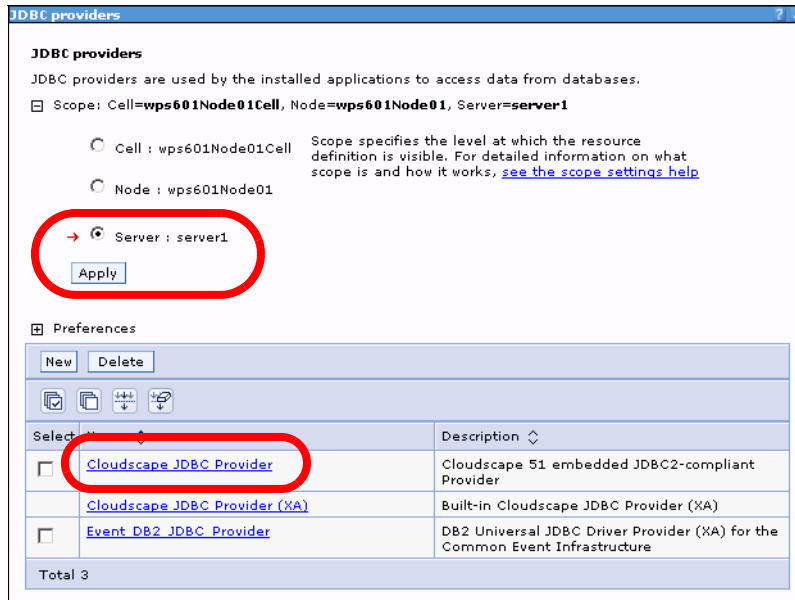


Figure 13-1 JDBC Provider in the scope of server1

5. Click **Cloudscape JDBC Provider** to create a new data source that will use Cloudscape JDBC Provider.
6. Click **Data sources** and click **New** to create a new data source.
7. Enter the following data source information in the data source properties, then click **Apply**.
 - Name: Reserve
 - JNDI Name: jdbc/reserve
 - Database name: \${USER_INSTALL_ROOT}/databases/RESERVE
8. Click **Save** twice to save the change to the master configuration.
9. Select the check box in front of **Reserve** data source and click **Test Connection** to test whether the data source can be connected. If the connection is tested successfully, the message in Example 13-1 appears.

Example 13-1 Successful message for test connection for data source

Test connection for data source RESERVE on server server1 at node widNode was successful.

10. Log out of the administrative console, then stop and start the server to apply the change.

13.1.2 Installing the Business Rules Manager

The Business Rules Manager is installed separately after WebSphere Process Server is installed. Before starting installation of the Business Rules Manager, make sure your WebSphere Process Server server instance is running.

1. To check whether the server is running, run these commands from a command prompt:

```
cd <WPS_HOME>\profiles\ProcSrv01\bin
serverstatus -all
```

2. The console log message in Example 13-2 appears if the server is running.

Example 13-2 Console message for serverstatus -all command

```
ADMU0116I: Tool information is being logged in file
           C:\WPS601\profiles\ProcSrv01\logs\serverStatus.log
ADMU0128I: Starting tool with the ProcSrv01 profile
ADMU0503I: Retrieving server status for all servers
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU0508I: The Application Server "server1" is STARTED
```

3. To install the Business Rules Manager, run the following command:

```
wsadmin -f <WPS_HOME>\bin\installBRManager.jacl
```

4. A successful completed installation produces the message in Example 13-3.

Example 13-3 Console message for successful installation of Business Rules Manager

```
ADMA5013I: Application BusinessRulesManager installed successfully.
```

After the Business Rules Manager is installed, confirm the installation by using the administrative console.

5. Open a Web browser and log in to the administrative console. The default URL for the administrative console is:

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

6. In the administrative console, select **Applications** → **Enterprise Applications** to list all installed applications. Figure 13-2 on page 337 shows that the BusinessRulesManager application is running.

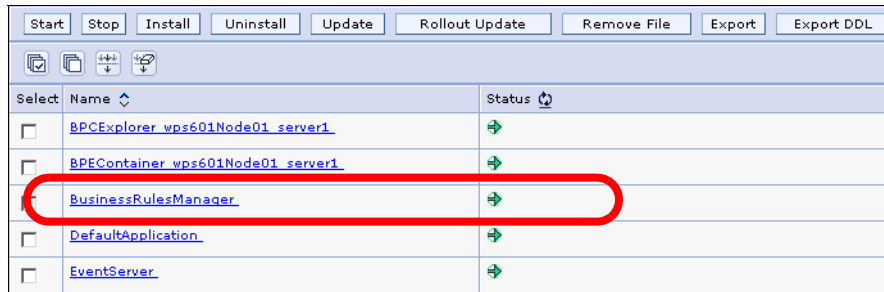


Figure 13-2 BusinessRulesManager running in administrative console

- Open a browser and launch the Business Rules Manager application. The default URL is:

<http://localhost:9080/br/webclient/pages/index.jsp>

Figure 13-3 shows the Business Rules Manager.



Figure 13-3 Business Rules Manager

13.2 Installing the business process application

This section installs the business process application we created in Chapter 12, “Assemble with WebSphere Integration Developer” on page 263. You can use either the EAR and ZIP files exported from that chapter, or the files that are supplied in the additional materials provided with this book. These files are located in the \WID\Export directory of the additional material.

This section describes two ways to install the application:

- ▶ Using the command line
- ▶ Using the administrative console

Important: Choose which method you wish to use, and complete the instructions for that method only. You should not attempt to install the application using both the command line and the administrative console.

13.2.1 Installing the application using command line

Use the files exported from WebSphere Integration Developer and start installation by using the command line. Either use the ZIP files exported from the previous chapter or use the ZIP files in the \WID\Export directory of the additional material supplied with this book.

1. To create a deployable EAR file, run these commands from a command line:

```
cd <WPS_HOME>\profiles\ProcSrv01\bin
servicedeploy <ZIP_HOME>\VehicleRent.zip -freeform
```

Note: Replace <WPS_HOME> with the home directory of your WebSphere Process Server installation and <ZIP_HOME> with the location of your ZIP files. The instructions in this chapter assume that you are using the default profile name of ProcSrv01. If your profile name is different, use the appropriate path for your profile.

2. The message in Example 13-4 displays when deployment is successful.

Example 13-4 Console message when deployment is successful

```
Exporting application
C:\WPS601\profiles\ProcSrv01\bin\VehicleRentApp.ear
Deployment has completed
Waiting for background jobs
Deleting workspace
```

3. Check whether the VehicleRentApp.ear file is generated:

```
dir *.ear
```

4. To install the EAR file, run this command to get in wsadmin interactive mode:

```
wsadmin.bat
```

5. In wsadmin interactive mode, install the generated EAR file by entering.

```
$AdminApp install VehicleRentApp.ear
```

6. The console message in Example 13-5 appears if installation is successful.

Example 13-5 Console message when installation is successful

```
ADMA5013I: Application VehicleRentApp installed successfully.
```

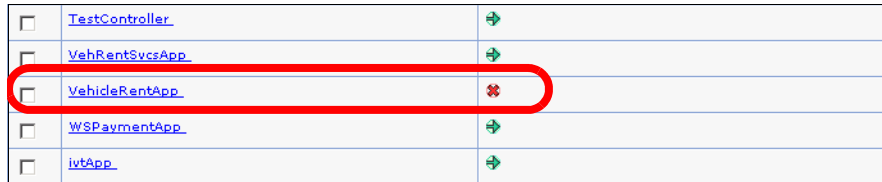
7. Save the changed configuration:

```
$AdminConfig save
exit
```


8. Open a browser and log in to the administrative console. The default URL for the administrative console is:

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

9. In the administrative console, select **Applications** → **Enterprise Applications** to list all installed applications. Figure 13-4 shows that the **VehicleRentApp** application is installed.



<input type="checkbox"/>	TestController	
<input type="checkbox"/>	VehRentSvcApp	
<input type="checkbox"/>	VehicleRentApp	
<input type="checkbox"/>	WSPaymentApp	
<input type="checkbox"/>	ivtApp	

Figure 13-4 *VehicleRentApp installed in administrative console*

10. Select the **VehicleRentApp** application, and click **Start** to start the application. Example 13-6 shows the successful start-up message.

Example 13-6 *Successful start-up message for VehicleRentApp*

Application VehicleRentApp on server server1 and node widNode started successfully

11. Repeat the installation steps for the other ZIP files as follows:

- VehRentSvc.zip
- ReserveSvc.zip
- FilePayment.zip
- WSPayment.zip

12. Repeat the installation steps starting at step 4 on page 338 for the following EAR files for Web services applications:

- ITSOSpeedyServices.ear
- PaymentProvider.ear

13.2.2 Installing the application using the administrative console

Note: Do not run the steps in this section if you have already completed the steps in 13.2.1, “Installing the application using command line” on page 338.

You can use the EAR files that are exported from WebSphere Integration Developer and start installation by using the administrative console. These

installation steps are not necessary if you have already installed applications using a command line.

Use either the EAR files from the previous chapter or the EAR files in the \WID\Export directory of the additional material supplied with this book.

1. Open a browser and log in to the administrative console. The default URL is:
`http://localhost:9060/ibm/console`
2. In the administrative console, select **Applications** → **Install New Application**.
3. In the Preparing for the application installation dialog, follow these steps:
 - a. Click **Browse**, and select the **VehicleRentApp.ear** file, which is exported from WebSphere Integration Developer.
 - b. Click **Next**, then **Next** again.
4. Accept the default configuration in the subsequent dialogs from **Step 1** through **Summary**. Click **Finish**. If the installation is successful, the message in Example 13-7 appears.

Example 13-7 Successful installation message

ADMA5011I: The cleanup of the temp directory for application VehicleRentApp is complete.
ADMA5013I: Application VehicleRentApp installed successfully.
Application VehicleRentApp installed successfully.
To start the application, first save changes to the master configuration.

5. Click **Save to Master Configuration**.
6. Click **Save** to save the configuration change to master configuration.
7. In the administrative console, select **Applications** → **Enterprise Applications**.
8. Select the **VehicleRentApp** application, and click **Start** to start the application.
9. Repeat the installation steps for the other EAR files as follows:
 - VehRentSvcApp.ear
 - ReserveSvcApp.ear
 - FilePaymentApp.ear
 - WSPaymentApp.ear
 - ITSOSpeedyServices.ear
 - PaymentProvider.ear

13.3 Running the application using BPC Explorer

This section uses the BPC Explorer to run and administrate the process. This section contains the following:

- ▶ Using the BPC Explorer as an end user
- ▶ Using the BPC Explorer as an administrator

13.3.1 Using the BPC Explorer as an end user

We will start one instance of business process and complete it using the BPC Explorer.

1. Open a Web browser, and launch the BPC Explorer. By default the URL is:
`http://localhost:9080/bpc`
2. Select **My Process Templates** from the left navigation frame to list the process templates where we can start a process instance.
3. Click the check box next to **Vehicle Fulfilment Process ToBe**. Click the **Start Instance** button as shown in Figure 13-5 to start a new process instance from the process template.

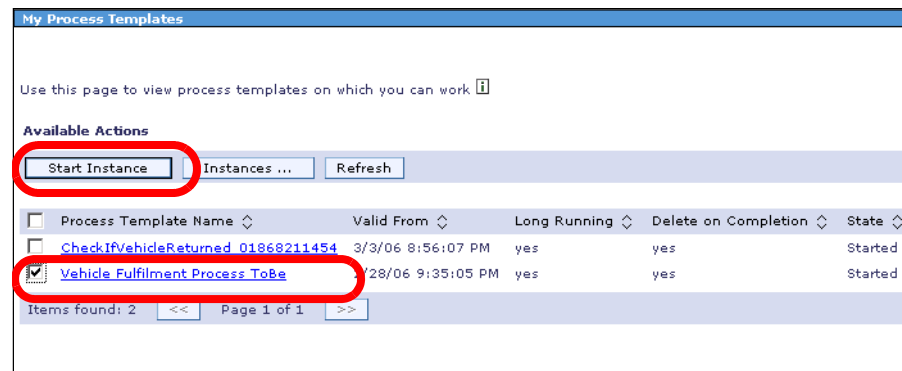


Figure 13-5 Process template list

4. Enter process input values as indicated in Table 13-1 on page 342.

Table 13-1 Sample input values to start a process instance

Field	Value	Constraint
Process Name	p001	Use a unique value, or leave blank for WebSphere Process Server to assign a unique value.
ReservationID	0	This is just an initial value: the actual ReservationID will be returned by the Pool Manager Web service.
VehicleGroupRequested	B	A, B or C. These types are in the Reserve database.
RiskLevel	low	Low, middle or high. Used in the business rule when an upgrade is needed.
LocationType	Airport	Airport, City or Suburb. Can be used for dimensional analysis in WebSphere Business Monitor.
Country	US	A name of a country such as Egypt, Germany, Korea, UK and US. Can be used for dimensional analysis in WebSphere Business Monitor.
City	Raleigh	A name of city. Can be used for dimensional analysis in WebSphere Business Monitor.
InventoryID	0	Just an initial value; the actual InventoryID will be returned by the Pool Manager Web service.

5. Click **Submit** to create a process instance as shown in Figure 13-6.

Process Input Message

Use this page to provide the input that is needed to start an instance of a business process. [?](#)

Available Actions

Submit

Process Template Name VehicleFulfilmentProcessToBe_1610588518
Operation InputCriterion

Process Name p001

Process Input Message

StartVehicleFulfilmentProcessInput	ReservationID	0
	VehicleGroupRequested	B
	VehicleGroupReceived	
	PickupDateTime	3/10/06 3:05 PM
	PlannedReturnDateTime	3/10/06 3:05 PM
	ActualReturnDateTime	3/10/06 3:05 PM

Figure 13-6 Process input message page

6. In the navigation frame, click **My Tasks** to see the tasks that are assigned to you. The **VehicleCollection** task is assigned in Figure 13-7.
7. Click the check box for **VehicleCollection** and click **Work on** to start working on the task.

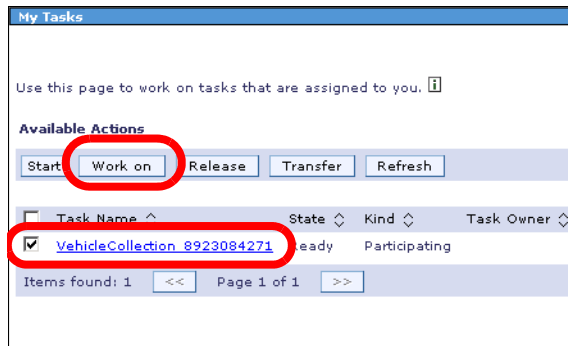


Figure 13-7 VehicleCollection task assigned in My Tasks page

Note: If you test the business process several times by requesting the same vehicle group (such as B) without returning the vehicles, this group becomes unavailable and the requested group is upgraded by the business rule. If so, the VehicleGroupReceived field of the VehicleCollection task input message will have been changed to a value other than the originally requested group.

8. Enter the values in Table 13-2, then click **Complete** to finish this task.

Table 13-2 Sample input values to complete VehicleCollection task

Field	How to locate the value to use
ReservationID	This value should be the same as the ReservationID in the Task Input Message for this VehicleCollection task.
PlannedReturnDateTime	This should be a date/time that is later than the current date/time.
InventoryID	This value should be same as the InventoryID in the Task Input Message for this VehicleCollection task.

Note: The BPC Explorer does not copy a task's input message into a task's output message automatically. You must add values manually into a task's output message, even they are same as the values of the input message. In a real environment, a custom Web client could copy the input message into the output message automatically if necessary.

9. Return the vehicle by sending an event to the running process instance (in this case, `CheckIfVehicleReturned`, which is a subprocess waiting for the event to arrive. If the event comes within the `PlannedReturnDate` of the process instance input value, the `CheckIfVehicleReturned` process instance will proceed with the process normally. If the event does not come within the `PlannedReturnDate`, the process instance will generate a timeout event and create an `UpdateRentalAgreement` task to adjust the `PlannedReturnDate`.
- a. In the navigation frame, click **My Process Templates** to list all of the process templates.
 - b. Click the **CheckIfVehicleReturned** process template, which is a specialized subprocess for receiving a vehicle return event.
 - c. Click the **Process Instances** tab to list the running process instances.

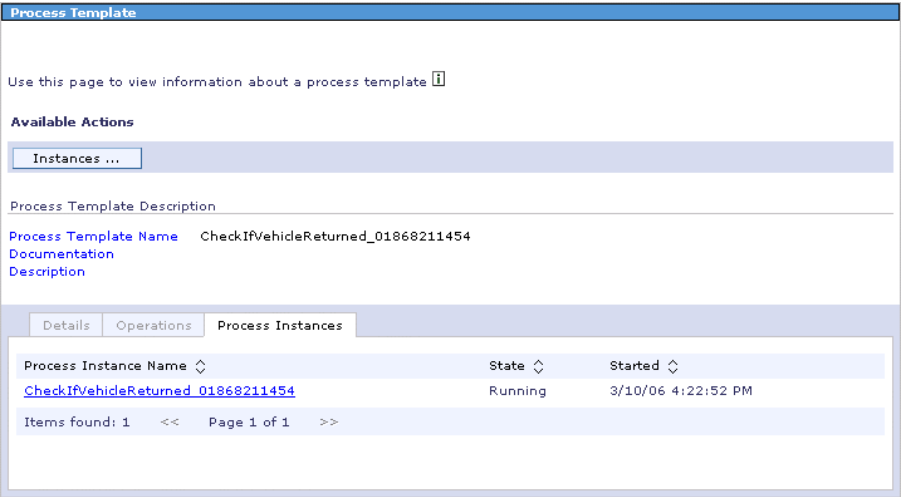


Figure 13-8 Process instances for `CheckIfVehicleReturned` subprocess

- d. Click the running process instance.
- e. Click **Events** tab.
- f. Click the **TurnIn** operation as shown in Figure 13-9. The `TurnIn` operation is the interface that is waiting for the vehicle return event.



Figure 13-9 Events tab to send event to running process instance

- g. Enter the values in Table 13-3 and click **Submit** to send the event to the running process instance.

Table 13-3 Sample input values to send the TurnIn event

Field	How to locate the value to use
ReservationID	The same value as entered in Table 13-2 on page 343.
InventoryID	The same value as entered in Table 13-2 on page 343.

Note: If several process instances are running, you can use any ReservationID that belongs to any process instance. When you send the event, the process instance that has the same value as the sending ReservationID receives it. WebSphere Process Server correlates the sending event and the waiting process instance by using the ReservationID.

10. In the navigation frame, click **My Tasks** to see the tasks assigned to you. If you see an UpdateRentalAgreement task, this indicates that you did not send the return event within the date/time specified in PlannedReturnDate. If so, follow these steps:
- Check UpdateRentalAgreement and click **Work on**.
 - Enter the values from Table 13-4 then click **Complete** to complete the task.

Table 13-4 Sample input values to complete UpdateRentalAgreement task

Field	How to locate the value to use
ReservationID	This value should be same as the ReservationID in the Task Input Message for this UpdateRentalAgreement task.
RentalStatus	Enter a value of out
PlannedReturnDateTime	This value should be a date/time later than the current date/time.
InventoryID	This value should be same as the InventoryID in the Task Input Message for this UpdateRentalAgreement task.

11. In the navigation frame, click **My Tasks** to see the tasks assigned to you. This time, the **VehicleReturnhandover** task is assigned in the My Tasks list.
12. Click the check box next to **VehicleReturnhandover** and click **Work on**.
13. Enter the values from Table 13-5 on page 346 and click **Complete** to finish this task.

Table 13-5 Sample input values to complete VehicleReturnhandover task

Field	How to locate the value to use
ReservationID	This value should be same as the ReservationID in the Task Input Message for this VehicleReturnhandover task.
VehicleGroupRequested	This should be same as the VehicleGroupRequested that was input when the process started.
PickupDateTime	This value should be a date/time earlier than the ActualReturnDateTime.
ActualReturnDateTime	This value should be a date/time later than the PickupDateTime.
LoyaltyClubNumber	Enter a value of A001
InventoryID	This value should be same as the InventoryID in the Task Input Message for this VehicleReturnhandover task.

Note: To calculate actual cost and discount, the CalculateDiscount business rule uses VehicleGroupRequested, PickupDateTime, ActualReturnDateTime, and LoyaltyClubNumber from the output message of the VehicleReturnhandover task.

14. The process instance is complete. Example 13-8 shows the message from the payment component in the SystemOut.log file. This message confirms that the process instance finished successfully. The payment component is called by the last activity of the process. The actual cost might vary from the value shown in Example 13-8.

Note: The <WPS_HOME>\profiles\ProcSrv01\logs\logs\server1 folder contains the System.out log file.

Example 13-8 Message from payment module that shows the process ended successfully

SystemOut	0 Payment done with actualCost :50.0
-----------	--------------------------------------

13.3.2 Using the BPC Explorer as an administrator

Now administer a process instance by using BPC Explorer: List the process instances, suspend and resume a process instance, list its activities, and then terminate the process instance. Follow these steps:

1. In a browser, launch the BPC Explorer. The default URL is:
`http://localhost:9080/bpc`
2. Start one process instance as guided in 13.3.1, “Using the BPC Explorer as an end user” on page 341. Do not finish the process instance. For example, you can stop after step 5.
3. Select **Process Instances** → **Administered By Me** in the left navigation frame to list the process instances.
4. You can suspend the running process instance by clicking the check box next to the running process instance, and clicking **Suspend** (Figure 13-10). The process instance state changes from Running to Suspended.

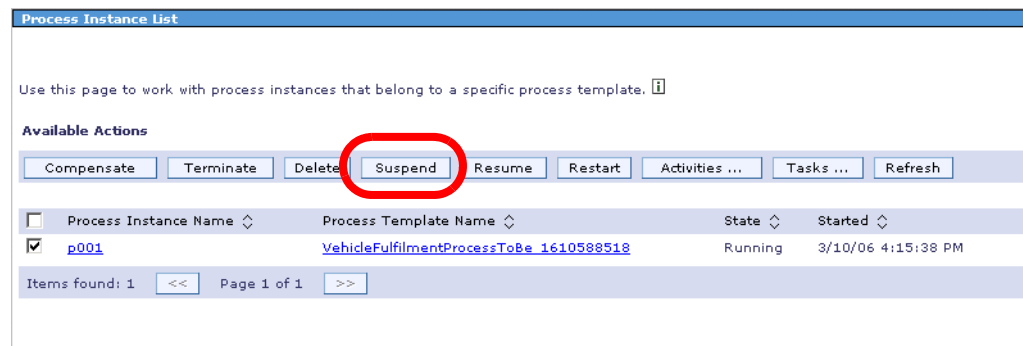


Figure 13-10 Administration of a process instance

5. Click the check box next to the suspended process instance. Click **Resume**. The process instance state changes from Suspended to Running.
6. Click the check box next to the running process instance, and click **Activities**. This lists all activities that are finished or running within the process instance (Figure 13-11 on page 348). You can also administer activities on this page.

Activities

Use this page to view the activities for a process instance, 1

Available Actions

RestartForce CompleteRefresh

<input type="checkbox"/>	Activity Name ▾	State ▾	Activity Kind ▾	Owner ▾	Activated ▾	Process Instance Name ▾
<input type="checkbox"/>	Vehicle Return handover	Running	Invoke		3/10/06 4:32:31 PM	p001
<input type="checkbox"/>	Check If Vehicle Returned	Finished	Invoke		3/10/06 4:22:52 PM	p001
<input type="checkbox"/>	Rental Agreement Status Set To Active	Finished	Invoke		3/10/06 4:22:51 PM	p001
<input type="checkbox"/>	Vehicle Collection	Finished	Invoke		3/10/06 4:15:46 PM	p001
<input type="checkbox"/>	Reserve Vehicle In System	Finished	Invoke		3/10/06 4:15:40 PM	p001

Items found: 5 << Page 1 of 1 >>

Figure 13-11 Activities list of a process instance

7. Select **Process Instances** → **Administrated By Me** from the left navigation frame to return to the instance list.
8. You can terminate the running process instance if you do not need to finish the process normally. Click the check box next to the running process instance, and click **Terminate**. The process instance state changes from Running to Terminated. Click the check box next to the running process instance, and click **Delete** to delete the process instance from the repository database.

13.4 Using the Business Rules Manager

You can change the preconfigured business rules while a process is running using the Business Rules Manager.

1. Open a browser and launch the Business Rules Manager application. The default URL is:
http://localhost:9080/br/webclient/pages/
2. Two Rule Books should be displayed. Expand **CalculateDiscounts** → **InputCriterion** and click **PriceRuleSet** to see the price and discount calculation business rule.
3. Click **Edit** to edit the business rule as shown in Figure 13-12 on page 349.

PriceRuleSet - Ruleset

Back

Edit

Copy

General Information

Last Published	Mar 7, 2006 16:31 (Local Time)	Status	Original
Description			

Rules

Name	Rule
Init	Init
CalculateRentalDuration	Calculate rental hours and days
DailyChargeForClassA	If vehicle type is Class A, then daily charge is 50 USD
DailyChargeForClassB	If vehicle type is Class B, then daily charge is 70 USD
DailyChargeForClassC	If vehicle type is Class C, then daily charge is 90 USD
AssignDiscountRateForTypeACustomerRule	If driver's royalty type is A, then apply 40% discount
AssignDiscountRateForTypeBCustomerRule	If driver's royalty type is B, then apply 20% discount
CalculateFinalPrice	Calculate final price

Figure 13-12 Business rule set to calculate price and discount rate

4. Change the daily charge for the Class B vehicle from 70 to 80 by changing the DailyChargeForClassB rule as shown in Table 13-6.

Table 13-6 Rule for Class C vehicle

Name	Rule
DailyChargeForClassB	If vehicle type is Class B, then daily charge is 80 USD

5. Click **Save** to save the changed the rule set.
6. Click **Publish and Revert**.
7. Click **Publish** to publish the changed rule set to **server**. A successful publish displays the message shown in Example 13-9 in the Messages box.

Example 13-9 Success message when changed rule is published

Selected rule page(s) have been published successfully.

8. You can run a new process instance to see whether the changed rule is applied correctly by looking at the SystemOut.log. When you finish one process instance, the SystemOut.log shows the actual cost calculated by the changed business rule, as shown in Example 13-10.

Example 13-10 SystemOut.log showing actual cost calculated by business rule set

SystemOut 0 Payment done with actualCost :46.0

13.5 Administering a selector component

Now dynamically change the payment module destination by using a selector component. The new payment module will first use the WebSphere Adapter for FlatFile. Using this module, payment information is generated in the file system as an XML-formatted file. After testing the WebSphere Adapter for FlatFile, change the payment module destination again, this time to use a Web service. The Web service will simply send a log message when it has been invoked.

Follow these steps:

1. Open a browser and log in to the administrative console. The default URL is:
`http://localhost:9060/ibm/console`
2. In the administrative console, select **Servers** → **Application Servers**.
3. Click **server1** in the Application servers list.
4. In the Business Integration section, click **Selectors** as shown in Figure 13-13.



Figure 13-13 Selector administration menu

5. Click **PaymentSelector** in the selector list. The PaymentSelector selects the destination for payment activity.
6. Click **InputCriterion** in the selector table list. The InputCriterion is the interface operation that sends a payment request message.
7. Click the **default** target to change the selector destination for the default target.
8. Change the value for the Target Components to **Exported Target: FilePayment PaymentToFlatFileOutboundExport1** and click **OK** as shown in Figure 13-14 on page 351 to use the payment module that is implemented using WebSphere Adapter for FlatFile. FilePayment is the name of the destination module. PaymentToFlatFileOutboundExport1 is the export component in the FilePayment module.

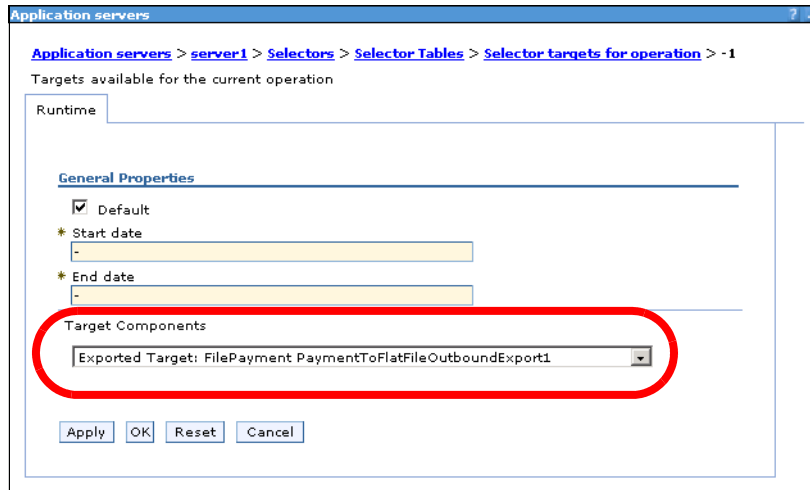


Figure 13-14 Selector target to use FlatFile adapter payment module

9. Select the **default** check box and click **Commit** to commit the change to the server repository.
10. The new payment application will attempt to create a file in the following directory. Create this directory:
C:\flatfile\output
11. Use the BPC Explorer to start and complete a process instance and check whether a payment information file is generated in C:\flatfile\output as shown in Example 13-11. The payment information file is generated by WebSphere Adapter for FlatFile. The file name is the same as the reservationID of the rental agreement.

Example 13-11 WebSphere Adapter for FlatFile output file contents

```
<?xml version="1.0" encoding="UTF-8"?>
<_:RentalAgreement xsi:type="itso:RentalAgreement"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:itso="http://ITSOSpeedyRentalsBusinessItemsCatalog"
xmlns:_="http://Businessitems/Carrentalcatalog">
  <ReservationID>44521</ReservationID>
  <VehicleGroupRequested>B</VehicleGroupRequested>
  <VehicleGroupReceived></VehicleGroupReceived>
  <PickupDateTime>2006-03-07T20:36:00.0Z</PickupDateTime>
  <PlannedReturnDateTime>2006-03-07T20:36:00.0Z</PlannedReturnDateTime>
  <ActualReturnDateTime>2006-03-10T20:36:00.0Z</ActualReturnDateTime>
  <ActualCost>126.0</ActualCost>
  <RentalStatus>in</RentalStatus>
```

```
<DriverDetails>
  <LicenseID></LicenseID>
  <RiskLevel></RiskLevel>
  <LoyaltyClubNumber>A001</LoyaltyClubNumber>
</DriverDetails>
<VehicleDetails>
  <Model></Model>
  <Manufacturer></Manufacturer>
  <Available>>false</Available>
  <InventoryNumber>11</InventoryNumber>
</VehicleDetails>
</_:RentalAgreement>
```

12. Change the selector destination to the Web services payment module: In the administrative console, click the **default** target to change the selector destination for the default target.
13. Change the value for Target Components to **Exported Target: WSPayment PaymentToPaymentServiceExport** to use the Web services payment module. Click **OK**.
14. Select the **default** check box and click **Commit** to commit the change to the server repository.
15. Start and complete a process instance and check SystemOut.log if the Web services payment module is called. If it is, you will see a log message similar to Example 13-12.

Example 13-12 SystemOut.log showing that the Web services payment module is called

SystemOut	0 Payment WebServices called.
SystemOut	0 *** Payment done with actual cost of 42.0



Manage with WebSphere Business Monitor

This chapter describes how to use WebSphere Business Monitor to analyze the results of the process instances of the process we created in Chapter 14, “Manage with WebSphere Business Monitor” on page 353. The chapter includes the following steps:

- ▶ Deploying the business measures model
- ▶ Testing the business measures model deployment
- ▶ Creating custom dashboards
- ▶ Monitoring the process
- ▶ Exporting results back to WebSphere Business Modeler

14.1 Overview

The following section shows how we created and used the monitoring infrastructure that enables the Manage phase of our business process management scenario. Before performing this implementation phase with WebSphere Business Monitor, we had already completed the Model, Assemble, and Deploy phases as described in previous chapters.

This chapter describes how to implement the following stages to use the monitoring infrastructure:

- ▶ The primary input to this phase is the business measures model that was created using the Business Measures Editor (BME) capability in WebSphere Business Modeler. In order to monitor a process, the WebSphere Business Monitor environment must be configured to recognize the process model, and this is achieved through deployment of a business measures model.
- ▶ The first stage of deployment involves generating schema information for the databases that will store the process data, and then updating the WebSphere Business Monitor databases using the generated schemas. The schema generation also generates tables that are used by DB2 replication and the second phase involves configuring and starting the replication.
- ▶ When the database infrastructure is complete and the replication is configured, the final stage of deployment imports the business measures model. When a model is imported, WebSphere Business Monitor begins to listen for events coming from the WebSphere Process Server execution environment. When these events arrive, it extracts the business data from them, according to the definitions in the business measures model, and persists that data to the datastores.
- ▶ After deployment is complete, the dashboards for monitoring the process can be built. This involves combining the 10 supplied views into custom dashboards, grouping information as appropriate for the dashboard users.
- ▶ When the dashboard is completed, process instances can be run on the WebSphere Process Server runtime, and they will be monitored by WebSphere Business Monitor. The dashboard can then be used to analyze the results, and finally the data can be exported back to WebSphere Business Modeler to improve the accuracy of simulations and analysis.

14.2 Deploying the business measures model

This section shows you how to use WebSphere Business Monitor to generate the database schema and DB2 replication configuration based on our experience

with our business measures model. We then describe the steps that to follow to update the databases and start the replication, and show how to import the model to begin the monitoring.

Note: Much of this section involves the configuration of the databases that are used by WebSphere Business Monitor. For further information, refer to the Information Center at:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.btools.help.monitor.doc/Doc/concepts/dbServices/monitordbconfigrepli.html>

14.2.1 Schema generation

Schema generation creates three sets of artifacts:

- ▶ Database schemas for updating the WebSphere Business Monitor databases to store process monitoring data
- ▶ DB2 Replication schemas and daemons for copying the business data between the WebSphere Business Monitor databases
- ▶ IBM DB2 Cube Views™ metadata

All of the artifacts are generated through an easy-to-use interface in the WebSphere Business Monitor administrative console. The database schemas are generated as DDL files, and the replication configuration is generated as batch files.

Configuring the schema generator

The following steps describe how we used the schema generator with our business measures model:

1. Start the Monitor Server and then start the administrative console, as described in Appendix B, “WebSphere Business Monitor system administration” on page 477.
2. Expand **WebSphere Business Monitor** → **Schema Generator** and click **Configuration**.
3. Specify the location of the business measures model on the Monitor Server, including the full path and name of the file. This file is called Monitor.zip and was exported directly from WebSphere Business Modeler. It is provided with the additional material supplied with this book in the \Modeler\Export directory. In our case, we had copied this file into C:\BMM\ on the Monitor Server, so we specified C:\BMM\Monitor.zip for the location.

Note: It is not possible to upload a business measures model from a remote machine when using the administrative console. Instead, you must have already copied the model to the Monitor Server manually, and must then specify the local path on the Monitor Server.

4. For the Output Directory, specify a location where you want to produce the generated schema and replication artifacts. We used C:\SchemaGen\. This is where you will find the generated artifacts after you run the schema generation. It can be any location you like, as long as it is a useful place to find the artifacts.
5. Check the **Ignore older deployments and generate all artifacts** option. This option indicates whether you are updating the schema of a previously deployed business measures model (a business measures model version upgrade). You should therefore check this option if you have a new business measures model, or if you want to overwrite an existing one.

Note: Checking this option is not strictly necessary when no previous business measures model has been deployed, but it is good practice to select it if this is a new model deployment. If you later deploy an updated version of the model, be sure that this option is not selected if you want to update the existing infrastructure to merge the new monitoring results with the old results.

6. We used the supplied default Table Space Properties File, C:\IBM\WebSphere\Monitor\install\mondb\default_minimum_tablespace.properties. You can choose to edit this file if you want to influence the table space properties for the generated schemas.
7. The overall configuration should be the same as in Figure 14-1.

Schema Generator Configuration Preferences
Preferences used for the generation of schemas

General Configuration State to Runtime Configuration

General Properties

Table Space Properties File
C:\IBM\WebSphere\Monitor\install\mondb\default_minimum_tablespace.properties

Business Measures Model
C:\BMM\Monitor.zip

Output Directory
C:\SchemaGen\

☒ Ignore older deployments and generate all artifacts

Apply OK Reset Cancel

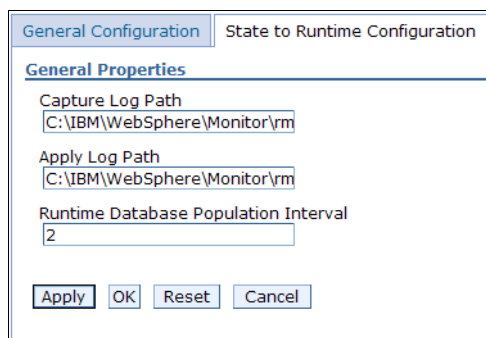
Figure 14-1 Schema generation settings

Be sure to click **Apply** before navigating away or your changes will be lost.

8. At this stage, the schema generation configuration is ready to go, but we also customize some additional settings. Click **State to Runtime Configuration**.

On this page you can influence the replication configuration between the State and Runtime databases. Replication uses two types of daemons, Capture and Apply. The configuration page enables you to change the log location that you want to use for both daemons. We accepted the defaults.

9. You can also influence how quickly the replication between these databases occurs. We reduced the default value from 5 minutes to 2 minutes by specifying the Runtime Database Population Interval as 2, as shown in Figure 14-2.



General Configuration | State to Runtime Configuration

General Properties

Capture Log Path
C:\IBM\WebSphere\Monitor\rm

Apply Log Path
C:\IBM\WebSphere\Monitor\rm

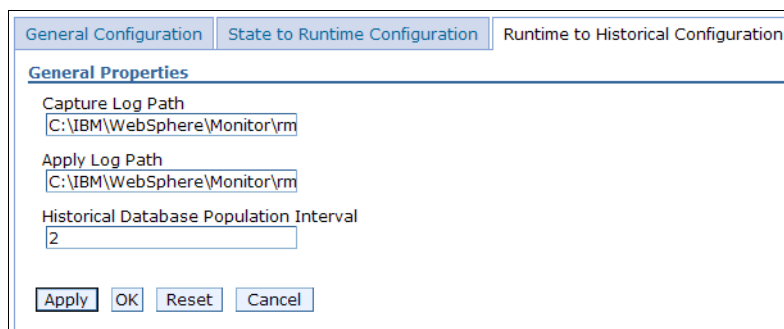
Runtime Database Population Interval
2

Apply OK Reset Cancel

Figure 14-2 State to Runtime replication configuration

Be sure to click **Apply** before navigating away or your changes will be lost.

10. Click **Runtime to Historical Configuration**. This page enables you to influence the replication daemons that will copy data between the Runtime and History databases. Again we accepted the default log locations, but we changed the replication interval from the default, 1440 minutes (24 hours), to just 2 minutes as shown in Figure 14-3. This increased how true to real-time our dashboards would be.



General Configuration | State to Runtime Configuration | Runtime to Historical Configuration

General Properties

Capture Log Path
C:\IBM\WebSphere\Monitor\rm

Apply Log Path
C:\IBM\WebSphere\Monitor\rm

Historical Database Population Interval
2

Apply OK Reset Cancel

Figure 14-3 Runtime to History replication configuration

Be sure to click **Apply** before navigating away or your changes will be lost.
11. Click the **Save** link, then click **Save** to persist your configuration changes.

Running the schema generator

Now that you have configured the schema generator, it is time to generate the artifacts as described in the following steps:

1. From the Administrative Console, expand **WebSphere Business Monitor** → **Schema Generator** and select **Generate**.
2. Click **Generate** and wait for generation to complete. You should see success messages as shown in Figure 14-4.

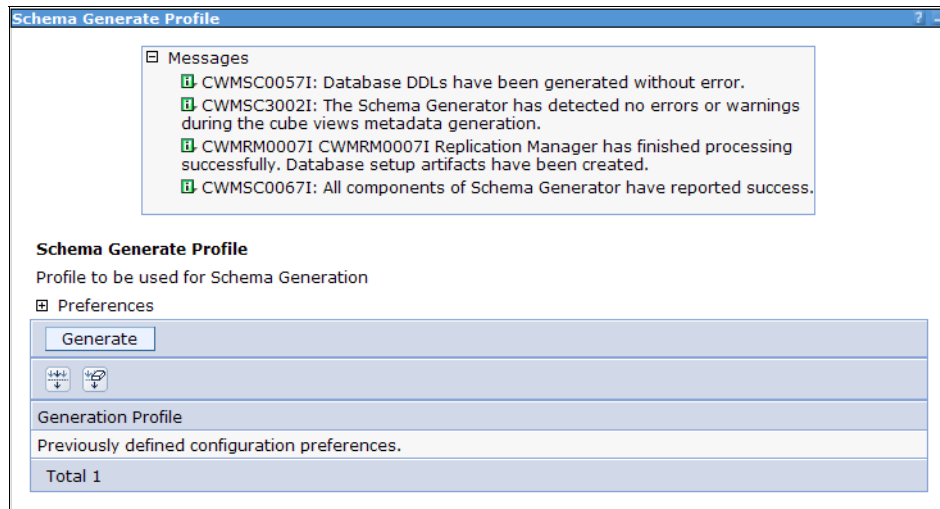


Figure 14-4 Successful schema generation

3. Log out from the Administrative Console by clicking **Logout** at the top.
4. Confirm that the artifacts were generated successfully by looking at the output directory contents (in our case, C:\SchemaGen\), shown in Figure 14-5 on page 359.

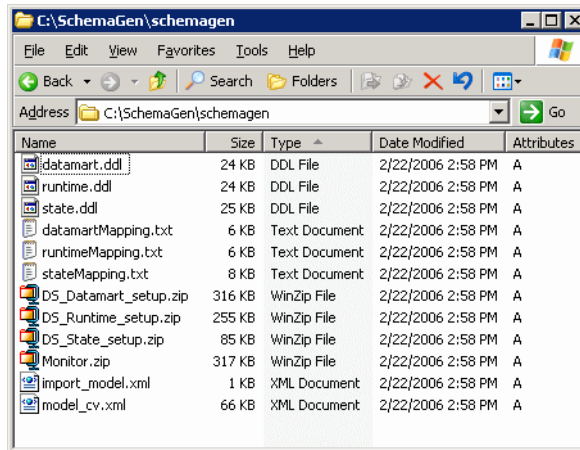


Figure 14-5 Generated artifacts

Note: We specified a generation directory of C:\SchemaGen\ . You can see in Figure 14-5 that an additional subdirectory was automatically created called *schemagen*. Therefore the files are all found in C:\SchemaGen\schemagen\.

5. Examine the generated artifacts to see that the following files were created:

- | | |
|----------------------------|--|
| state.ddl | Contains the schema changes to persist the model data into the State database. |
| runtime.ddl | Contains the schema changes to persist the model data into the Runtime database. |
| datamart.ddl | Contains the schema changes to persist the model data into the History database. |
| stateMapping.txt | Contains the name mappings for the State database between the generated table names and actual names in the business measures model. |
| runtimeMapping.txt | Contains the name mappings for the Runtime database between the generated table names and actual names in the business measures model. |
| datamartMapping.txt | Contains the name mappings for the History database between the generated table names and actual names in the business measures model. |
| DS_State_setup.zip | Contains the replication configuration artifacts for replicating from the State database. |

DS_Runtime_setup.zip	Contains the replication configuration artifacts for replicating to and from the Runtime database.
DS_Datamart_setup.zip	Contains the replication configuration artifacts for replicating to the Runtime database.
import_model.xml	Used to configure DB2 Cube Views and specifies the operation to perform using model_cv.xml.
model_cv.xml	Specifies the DB2 Cube Views metadata object. It is used for configuring DB2 Cube Views.
Monitor.zip	The generated implementation of the business measures model. It is used when importing the business measures model.

Note: The generated artifacts use the name Datamart instead of History. The names are interchangeable because they mean the same thing.

Next, use the generated artifacts to configure the databases, database replication, and cubes.

14.2.2 Database configuration

Now that the schema artifacts (DDL files) are generated, they should be executed. The following steps describe how we used them to update the databases so that they can hold data for the specific business measures model.

1. Use the three DDL files: state.ddl, runtime.ddl, and datamart.ddl. We recommend that you run each DDL locally on the machine that hosts the database it applies to. For example, if you have the same configuration as we used, then the State database exists on the Monitor Server machine whereas the Runtime, History, and Repository exist on the Dashboard Client machine.

Note: You can run the DDLs from remote servers as long as the databases were cataloged locally. For example, in our scenario we could have cataloged the Runtime, History, and Repository databases on the Monitor Server, then executed all scripts on the Monitor Server.

2. This in mind, we copied the entire contents of the C:\SchemaGen\schemagen\ directory to the Dashboard Client, copying them into C:\DashboardSchemaGen\schemagen\ as shown in Figure 14-6 on page 361. Only certain artifacts were necessary, but it was easier to copy everything.

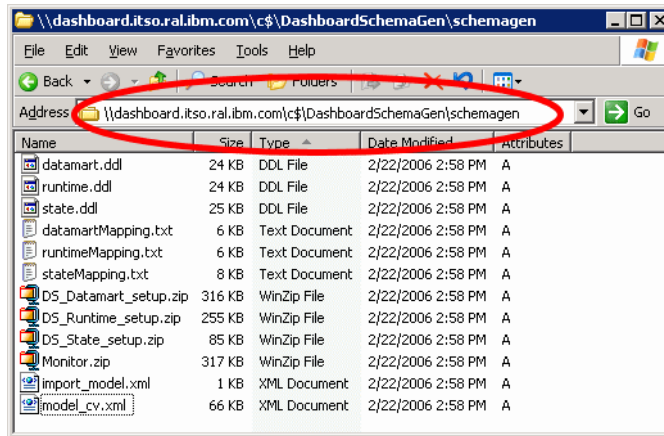


Figure 14-6 Copying the generated artifacts to the Dashboard Client

- Follow the Information Center instructions for running the DDL scripts to create the tables:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.btools.help.monitor.doc/Doc/tasks/dbServices/CreatOMDBtables.html>

- Open a DB2 Command Window on the *Monitor Server* by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
- Enter the following commands:

```
cd \SchemaGen\schemagen\
db2 terminate
set DB2CODEPAGE=1208
db2 connect to STATE user db2admin
```

You are prompted for the db2admin password, which defaults to monPa55w0rd:

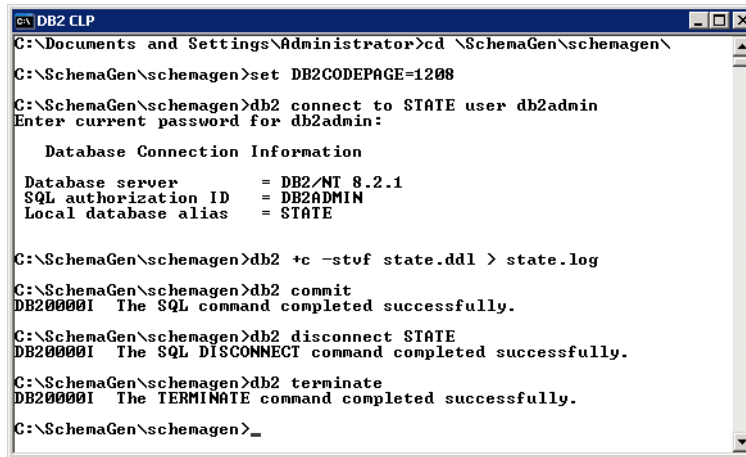
```
db2 +c -stvf state.ddl > state.log
```

At this point you should review the state.log file to check for errors. If there were none, continue entering these commands:

```
db2 commit
db2 disconnect STATE
db2 terminate
```

Important: Check the log files for *each* DDL because the command will not produce any warnings or errors. They will be found only in the log file.

The overall outcome should resemble Figure 14-7 on page 362.



```
C:\DB2 CLP
C:\Documents and Settings\Administrator>cd \SchemaGen\schemagen\
C:\SchemaGen\schemagen>set DB2CODEPAGE=1208
C:\SchemaGen\schemagen>db2 connect to STATE user db2admin
Enter current password for db2admin:

Database Connection Information

Database server      = DB2/NT 8.2.1
SQL authorization ID = DB2ADMIN
Local database alias = STATE

C:\SchemaGen\schemagen>db2 +c -stvf state.ddl > state.log
C:\SchemaGen\schemagen>db2 commit
DB20000I The SQL command completed successfully.
C:\SchemaGen\schemagen>db2 disconnect STATE
DB20000I The SQL DISCONNECT command completed successfully.
C:\SchemaGen\schemagen>db2 terminate
DB20000I The TERMINATE command completed successfully.
C:\SchemaGen\schemagen>_
```

Figure 14-7 *state.ddl* configuration

6. Open a DB2 Command Window on the *Dashboard Client* by clicking **Start** → **Run**. Enter db2cmd and click **OK**.

Important: Be sure that you are working on the Dashboard Client if you are following the same setup that we used, where the Runtime database was created on the Dashboard Client.

7. Enter the following commands:

```
cd \DashboardSchemaGen\schemagen\
db2 terminate
set DB2CODEPAGE=1208
db2 connect to RUNTIME user db2admin
```

You are prompted for the db2admin password, which defaults to monPa55w0rd.

```
db2 +c -stvf runtime.ddl > runtime.log
```

At this point, review the runtime.log file to check for errors. If there were none, continue entering these commands:

```
db2 commit
db2 disconnect RUNTIME
db2 terminate
```

8. The overall outcome should resemble Figure 14-8 on page 363.


```

C:\DashboardSchemaGen\schemagen>cd \DashboardSchemaGen\schemagen
C:\DashboardSchemaGen\schemagen>db2 terminate
DB20000I The TERMINATE command completed successfully.
C:\DashboardSchemaGen\schemagen>set DB2CODEPAGE=1208
C:\DashboardSchemaGen\schemagen>db2 connect to RUNTIME user db2admin
Enter current password for db2admin:

Database Connection Information
Database server      = DB2/NT 8.2.1
SQL authorization ID = DB2ADMIN
Local database alias = RUNTIME

C:\DashboardSchemaGen\schemagen>db2 +c -stvf runtime.ddl > runtime.log

C:\DashboardSchemaGen\schemagen>db2 commit
DB20000I The SQL command completed successfully.
C:\DashboardSchemaGen\schemagen>db2 disconnect RUNTIME
DB20000I The SQL DISCONNECT command completed successfully.
C:\DashboardSchemaGen\schemagen>db2 terminate
DB20000I The TERMINATE command completed successfully.
C:\DashboardSchemaGen\schemagen>_

```

Figure 14-8 runtime.ddl configuration

9. Open another DB2 Command Window, again on the *Dashboard Client* by clicking **Start** → **Run**. Enter db2cmd and click **OK**.

Important: Be sure that you are working on the Dashboard Client if you are following the same setup that we used, where the History database was created on the Dashboard Client.

10. Enter the following commands:

```

cd \DashboardSchemaGen\schemagen\
db2 terminate
set DB2CODEPAGE=1208
db2 connect to HISTORY user db2admin

```

You are prompted for the db2admin password, which defaults to monPa55w0rd.

```
db2 +c -stvf datamart.ddl > datamart.log
```

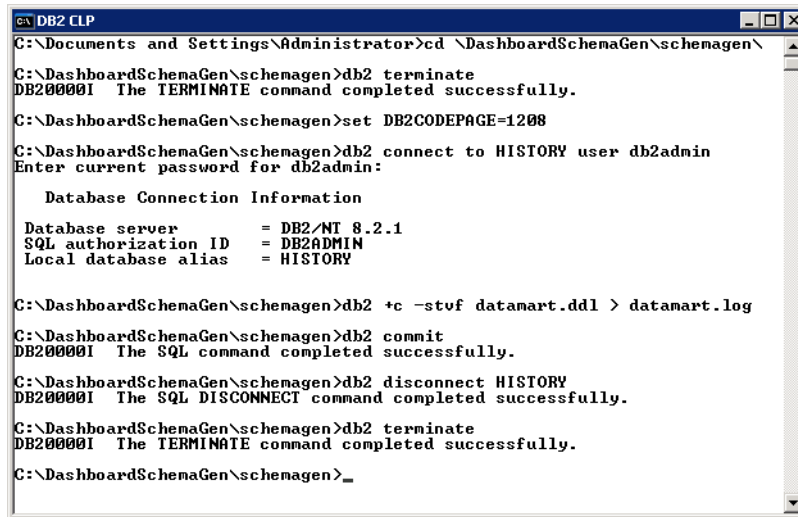
At this point, you should review the datamart.log file to check for errors. If there were none, continue entering these commands:

```

db2 commit
db2 disconnect HISTORY
db2 terminate

```

11. The overall outcome should resemble Figure 14-9 on page 364.



```
C:\Documents and Settings\Administrator>cd \DashboardSchemaGen\schemagen\
C:\DashboardSchemaGen\schemagen>db2 terminate
DB20000I The TERMINATE command completed successfully.
C:\DashboardSchemaGen\schemagen>set DB2CODEPAGE=1208
C:\DashboardSchemaGen\schemagen>db2 connect to HISTORY user db2admin
Enter current password for db2admin:

Database Connection Information
Database server      = DB2/NT 8.2.1
SQL authorization ID = DB2ADMIN
Local database alias = HISTORY

C:\DashboardSchemaGen\schemagen>db2 +c -stvf datamart.ddl > datamart.log
C:\DashboardSchemaGen\schemagen>db2 commit
DB20000I The SQL command completed successfully.
C:\DashboardSchemaGen\schemagen>db2 disconnect HISTORY
DB20000I The SQL DISCONNECT command completed successfully.
C:\DashboardSchemaGen\schemagen>db2 terminate
DB20000I The TERMINATE command completed successfully.
C:\DashboardSchemaGen\schemagen>_
```

Figure 14-9 datamart.ddl configuration

The databases are now configured for the business measures model. The next stage is to configure the replication between these databases.

14.2.3 Replication configuration

Replication occurs between two databases, and for WebSphere Business Monitor there are two replication sets to configure:

- ▶ State database to Runtime database replication (S2R)
- ▶ Runtime database to History database replication (R2H)

Each replication set consists of two types of daemons: the Capture daemons and the Apply daemons. Capture daemons are responsible for capturing the source database changes. Apply daemons are responsible for applying those changes to the target database. Therefore, there are four sets of daemons overall:

- ▶ S2R Capture daemons
- ▶ S2R Apply daemons
- ▶ R2H Capture daemons
- ▶ R2H Apply daemons

Each set requires one or more daemons per process model, and the number of daemons required in each set varies by model. There will be at least one daemon in each set.

In a distributed database topology, it is important to think about where each set of daemons should be located. For example, if the State database is on one

machine and the Runtime is on another, where should the S2R Apply daemons be located?

It is also important to consider what remote database cataloging requirements are in this scenario. Each set of daemons requires access to one or more databases, so you must ensure that all of the databases that a daemon set will access are either local or remotely cataloged. Table 14-1 summarizes the location and catalog requirements for each set of daemons.

Table 14-1 Daemon set access requirements and location recommendations

Daemon set	Which databases accessed?	Which machine to run daemon on?
S2R Capture	State	Same machine as State database
S2R Apply	State Runtime	Same machine as Runtime database
R2H Capture	Runtime	Same machine as Runtime database
R2H Apply	Runtime History	Same machine as History database

In our configuration, the State database was on the Monitor Server but the Runtime and History databases were on the Dashboard Client. Therefore we needed to configure the daemons as shown in Table 14-2.

Table 14-2 Our configuration for daemon locations

Daemon set	Which machine?
S2R Capture	Monitor Server
S2R Apply	Dashboard Client
R2H Capture	Dashboard Client
R2H Apply	Dashboard Client

We also had one additional cataloging requirement due to our configuration: We had to catalog the State database on the Dashboard Client machine so that the S2R Apply daemon set had access to it.

Configuring the State to Runtime Capture daemon

This daemon was configured on the *Monitor Server* because this is where the State database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

1. On the *Monitor Server*, extract the DS_State_setup.zip file from C:\SchemaGen\schemagen\ to C:\Replication\State\ (Figure 14-10).

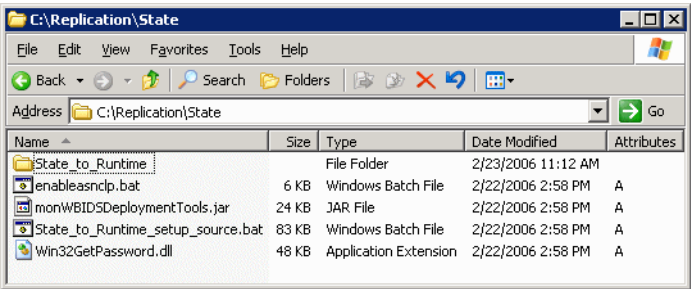


Figure 14-10 Extracting the State database replication artifacts

2. Open a DB2 Command Window on the *Monitor Server* by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
3. Enter the following commands:

```
cd \Replication\State\  
State_to_Runtime_setup_source.bat
```

The script runs and asks for the user ID and password for connecting to the State database as shown in Figure 14-11 on page 367. We used the default values, db2admin and monPa55w0rd.

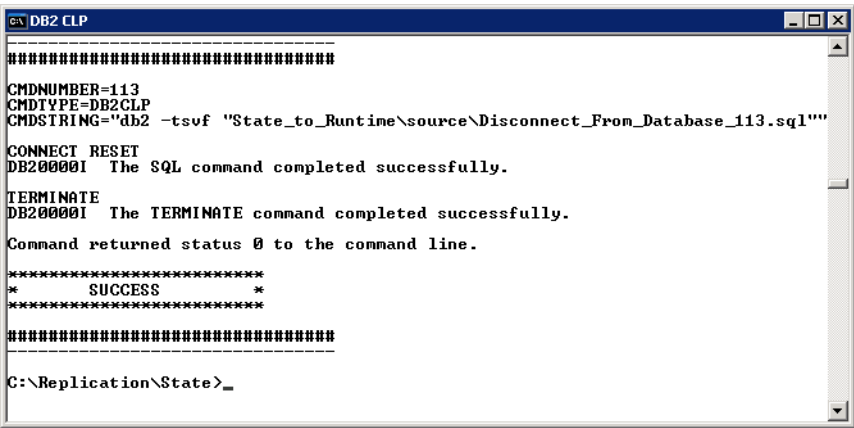
```

C:\DB2 CLP
C:\Documents and Settings\Administrator>cd \Replication\State\
C:\Replication\State>State_to_Runtime_setup_source.bat
Check State_to_Runtime_setup_source.bat.log For Errors
Testing Java Version
java version "1.4.2"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2)
Classic VM (build 1.4.2, JRE 1.4.2 IBM Windows 32 build cn142-20050609 <JIT enabled: jit>)
SETTING UP ASNCPL COMMAND LINE CLASSPATH
Testing Current Classpath Settings <> Empty means default settings.
The java class is not found: com/ibm/db2/tools/repl/replapis/cmdline/Asncpl
Testing "c:\Program Files\IBM\SQLLIB".
Testing "C:\IBM\SQLLIB".
Setting Classpath For Test.
Testing Current Classpath Settings <C:\IBM\SQLLIB> Empty means default settings.
ASNCPL Command Line Is Successfully Setup
Testing RMSecurityWrapper
The java class is not found: com/ibm/whimonitor/dataservices/deployment/RMSecurityWrapper
RMSecurityWrapper Failed Test # 1
Setting Classpath For Next Test
Testing RMSecurityWrapper
The environment for this command line session is successfully set up for RMSecurityWrapper
WBIRMSSTATUS=0
*****
*****SUCCESS*****
*****
Check State_to_Runtime_setup_source.bat.log For Errors
#####
CMDNUMBER=0
CMDTYPE=DEPLOYSHELL
CMDSTRING="CALL "State_to_Runtime\source\InitializeSecurity_0.bat""
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source\Connect_To_Database_1.db2connect".
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source>CreateCaptureCDRegistration_10.asnccli".
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source>CreateCaptureCDRegistration_40.asnccli".
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source>CreateCaptureCDRegistration_42.asnccli".
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source>CreateCaptureCDRegistration_69.asnccli".
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source>CreateCaptureCDRegistration_8.asnccli".
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source>CreateCaptureCDRegistration_84.asnccli".
CWMMRM9029I Scanning file "C:\Replication\State\.\State_to_Runtime\source>CreateCaptureCDRegistration_86.asnccli".
CWMMRM9034I Number of files analyzed: 9. Number of unique database references identified: 1.
CWMMRM9019I Type a user ID to be used to connect to database STATE:db2admin
CWMMRM9020I Type a password for this user:

```

Figure 14-11 Configuring the State to Runtime Capture daemon

4. The script should complete and the final command should show SUCCESS as in Figure 14-12. Review the entire `State_to_Runtime_setup_source.bat.log` file to check that there were no errors, only warnings or successes.



```
DB2 CLP

=====
CMDNUMBER=113
CMDTYPE=DB2CLP
CMDSTRING="db2 -tsvf "State_to_Runtime\source\Disconnect_From_Database_113.sql""

CONNECT RESET
DB20000I The SQL command completed successfully.

TERMINATE
DB20000I The TERMINATE command completed successfully.

Command returned status 0 to the command line.

*****
*      SUCCESS      *
*****

=====

C:\Replication\State>_
```

Figure 14-12 Successful configuration of the State to Runtime Capture daemon

Configuring State to Runtime Apply daemon

This daemon was configured on the *Dashboard Client* because this is where the Runtime database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

Important: Be sure to switch to the Dashboard Client machine.

1. On the *Dashboard Client*, extract the `DS_Runtime_setup.zip` file from `C:\DashboardSchemaGen\schemagen\` to `C:\DashboardReplication\Runtime\` as shown in Figure 14-13.

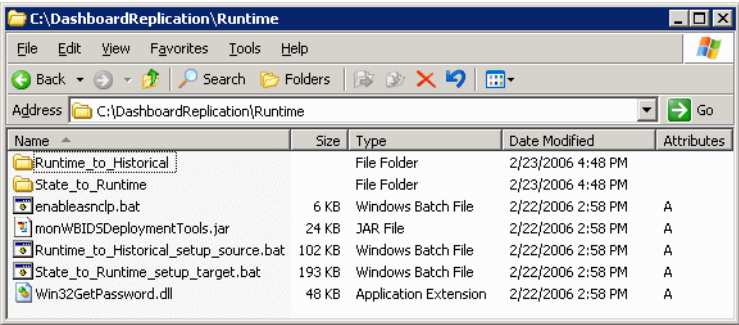


Figure 14-13 Extracting the Runtime database replication artifacts

Note: The DS_Runtime_setup.zip file contains the S2R Apply daemon and the R2H Capture daemon. This is because both daemons primarily access the Runtime database.

2. Open a DB2 Command Window on the *Dashboard Client* by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
3. Enter the following commands:

```
cd \DashboardReplication\Runtime\  
State_to_Runtime_setup_target.bat
```

The script runs, and this time asks for the user ID and password for connecting to the Runtime and State databases. We used the default values, db2admin and monPa55w0rd.
4. The script should complete and the final command should show SUCCESS. You should review the entire log file, State_to_Runtime_setup_target.bat.log, to check that there were no errors, only warnings or successes.

Configuring Runtime to History Capture daemon

This daemon was configured on the *Dashboard Client* because this is where the Runtime database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

1. On the *Dashboard Client*, you should have already extracted the DS_Runtime_setup.zip to C:\DashboardReplication\Runtime\ when configuring the State to Runtime Apply daemon.
2. Open a DB2 Command Window on the *Dashboard Client* by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
3. Enter the following commands:

```
cd \DashboardReplication\Runtime\  
Runtime_to_Historical_setup_source.bat
```

The script runs and asks for the user ID and password for connecting to the Runtime and State databases. We used the default values, db2admin and monPa55w0rd.
4. The script should complete and the final command visible should show SUCCESS. Review the entire Runtime_to_History_setup_source.bat.log file to check that there were no errors, only warnings or successes.

Configuring Runtime to History Apply daemon

This daemon was configured on the *Dashboard Client* because this is where the *History* database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

1. On the *Dashboard Client*, extract the DS_Datamart_setup.zip file from C:\DashboardSchemaGen\schemagen\ to C:\DashboardReplication\History\ as shown in Figure 14-14.

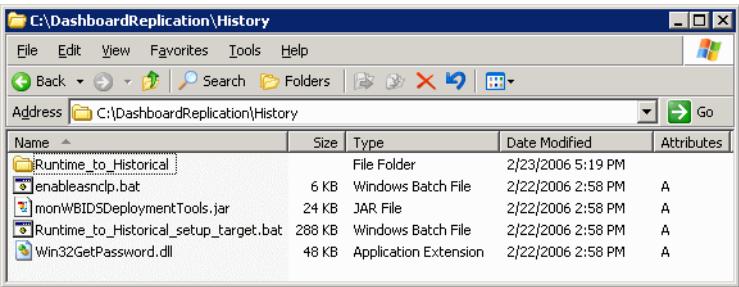


Figure 14-14 Extracting the History database replication artifacts

2. Open a DB2 Command Window on the *Dashboard Client* by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
3. Enter the following commands:

```
cd \DashboardReplication\History\  
Runtime_to_Historical_setup_target.bat
```

The script runs and asks for the user ID and password for connecting to the History and Runtime databases. We used the default values, db2admin and monPa55w0rd.
4. The script should complete and the final command visible should show SUCCESS. Review the entire Runtime_to_History_setup_target.bat.log file to check that there were no errors, only warnings or successes.

14.2.4 Starting the Replication daemons

This section describes how to complete the configuration of the daemons and start them running.

Starting the State to Runtime Capture daemon

This daemon was started on the *Monitor Server* because this is where the State database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

Important: Be sure to switch to the Monitor Server machine.

1. Open a DB2 Command Window on the Monitor Server by clicking **Start** → **Run**. Enter db2cmd and click **OK**.

2. Enter the following commands:

```
cd C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture\  
asnpwd INIT encrypt all using password.aut  
asnpwd ADD alias STATE id db2admin password monPa55w0rd using  
password.aut
```

This creates an encrypted user ID and password file for starting the daemons. Be sure to replace db2admin and monPa55w0rd with your own user ID and password values.

3. Before starting the daemon, edit the start script to tell it to use the password file you just created. Enter the following commands to edit the file:

```
cd C:\Replication\State\State_to_Runtime\source\  
notepad StartCapture_5.bat
```

Note: Each type of replication uses a set of one or more daemons. You should check the contents of the directory to find out how many daemons were generated. Each daemon name will have this format: StartCapture_<number>.bat. The <number> values are generated and are not important. Our example had only one script for the State to Runtime Capture daemon, so we only edited and started StartCapture_5.bat.

4. Edit the final line in the file to append the string pwdfile="password.aut" so that the line now reads as shown:

```
db2cmd asncap CAPTURE_SERVER=STATE CAPTURE_SCHEMA=CAPTURE_1  
CAPTURE_PATH="C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture"  
pwdfile="password.aut"
```

Note: This entire command should take up just one line.

5. Save the file and close Notepad.
6. Enter the following command to start the daemon:

```
StartCapture_5.bat
```

Figure 14-15 on page 372 shows the commands we entered.

```

C:\Documents and Settings\Administrator>cd C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture\

C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture>asnpwd INIT encrypt all using password.aut
2006-02-24-12.18.30.328000 ASN1981I "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".

C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture>asnpwd ADD alias STATE id db2admin password monPa55w0rd using password.aut
2006-02-24-12.18.49.859000 ASN1981I "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".

C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture>cd C:\Replication\State\State_to_Runtime\source\

C:\Replication\State\State_to_Runtime\source>notepad StartCapture_5.bat

C:\Replication\State\State_to_Runtime\source>StartCapture_5.bat

C:\Replication\State\State_to_Runtime\source>set DB2CODEPAGE=1208

C:\Replication\State\State_to_Runtime\source>db2cmd asncap CAPTURE_SERVER=STATE CAPTURE_SCHEMA=CAPTURE_1 CAPTURE_PATH="C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture" pwdfile="password.aut"

C:\Replication\State\State_to_Runtime\source>_

```

Figure 14-15 Starting the State to Runtime Capture daemon

7. The daemon should open in a separate window, resembling Figure 14-16.

```

DB2 CLP - db2setcp.bat asncap CAPTURE_SERVER=STATE CAPTURE_SCHEMA=CAPTURE_1 CAPTURE_PA...
2006-02-24-14.12.14.875000 ASN01001 CAPTURE "CAPTURE_1" : "WorkerThread". The Capture program initialization is successful.
2006-02-24-14.12.14.875000 ASN01091 CAPTURE "CAPTURE_1" : "WorkerThread". The Capture program has successfully initialized and is capturing data changes for "0" registrations. "0" registrations are in a stopped state. "?" registrations are in an inactive state.

```

Figure 14-16 State to Runtime Capture daemon successfully started

Starting the State to Runtime Apply daemon

This daemon was started on the Dashboard Client because this is where the Runtime database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

Important: Be sure to switch to the Dashboard Client machine.

1. Open a DB2 Command Window on the Dashboard Client by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
2. Enter the following commands as shown in Figure 14-15.

```

mkdir C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply
cd C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply
asnpwd INIT encrypt all using password.aut

```

```
asnpwd ADD alias STATE id db2admin password monPa55w0rd using  
password.aut
```

This creates an encrypted user ID and password file for starting the daemons. Be sure to replace db2admin and monPa55w0rd with your own user ID and password values.

As you can see above, it was also necessary to create the log directory C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply using a mkdir command because this directory did not already exist on the Dashboard Client.

Note: This log directory is user-specified in the administrative console schema generation configuration settings. If you specified a different directory, create a log directory that is appropriate for your configuration.

3. Before starting the daemon, edit the start script to tell it to use the password file you just created. Enter the following commands to edit the file:

```
cd C:\DashboardReplication\Runtime\State_to_Runtime\target  
notepad StartApply_12.bat
```

Note: Each type of replication uses a set of one or more daemons. You should check the contents of the directory to find out how many daemons were generated. Each daemon name will have this format: StartApply_<number>.bat. The <number> values are generated and are not important. Our example there was only one script for the State to Runtime Apply daemon so we only edited and started StartApply_12.bat.

4. Edit the final line in the file to append the string `pwdfile="password.aut"` so that the line now reads as shown (your example will differ if you configured a different log location):

```
db2cmd asnapply APPLY_QUAL=APPLY_4 CONTROL_SERVER=RUNTIME  
APPLY_PATH="C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply"  
pwdfile="password.aut"
```

Note: This entire command should take up just one line.

5. Save the file and close Notepad.
6. Enter the following command to start the daemon:

```
StartApply_12.bat
```

Figure 14-17 on page 374 shows the commands we entered.

```

C:\DB2 CLP
C:\Documents and Settings\Administrator>mkdir C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply
C:\Documents and Settings\Administrator>cd C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply
C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply>asnpwd INIT encrypt all using password.aut
2006-02-24-16.17.50.421000 ASN19811 "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".
C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply>asnpwd ADD alias STATE id db2admin password monPa55w0rd using password.aut
2006-02-24-16.17.59.015000 ASN19811 "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".
C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply>cd C:\DashboardReplication\Runtime\State_to_Runtime\target
C:\DashboardReplication\Runtime\State_to_Runtime\target>notepad StartApply_12.bat
C:\DashboardReplication\Runtime\State_to_Runtime\target>StartApply_12.bat
C:\DashboardReplication\Runtime\State_to_Runtime\target>set DB2CODEPAGE=1208
C:\DashboardReplication\Runtime\State_to_Runtime\target>db2cmd asnapply APPLY_QUAL=APPLY_4 CONTROL_SERVER=RUNTIME APPLY_PATH="C:\IBM\WebSphere\Monitor\rm\logs\s2r\apply" pwdfile="password.aut"
C:\DashboardReplication\Runtime\State_to_Runtime\target>_

```

Figure 14-17 Starting the State to Runtime Apply daemon

7. The daemon should open in a new window resembling Figure 14-18.

```

C:\DB2 CLP - db2setcp.bat asnapply APPLY_QUAL=APPLY_4 CONTROL_SERVER=RUNTIME APPLY_P...
2006-02-24-16.23.07.968000 ASN10451 APPLY "APPLY_4" : "Initial". The Apply program was started using database "RUNTIME".

```

Figure 14-18 State to Runtime Apply daemon successfully started

Starting the Runtime to History Capture daemon

This daemon was started on the Dashboard Client because this is where the Runtime database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

1. Open a DB2 Command Window on the Dashboard Client by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
2. Enter the following commands as shown in Figure 14-15 on page 372.

```

mkdir C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture\
cd C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture\
asnpwd INIT encrypt all using password.aut
asnpwd ADD alias STATE id db2admin password monPa55w0rd using password.aut

```

This creates an encrypted user ID and password file for starting the daemons. Be sure to replace db2admin and monPa55w0rd with your own user ID and password values.

As you can see above, it was also necessary to create the log directory, C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture using a mkdir command. This was because this directory did not already exist on the Dashboard Client.

Note: This log directory is user-specified in the administrative console schema generation configuration settings. If you specified a different directory, create a log directory that is appropriate for your configuration.

3. Before starting the daemon, edit the start script to tell it to use the password file you just created. Enter the following commands to edit the file:

```
cd C:\DashboardReplication\Runtime\Runtime_to_Historical\source\  
notepad StartCapture_5.bat
```

Note: Each type of replication uses a set of one or more daemons. You should check the contents of the directory to find out how many daemons were generated. Each daemon name will have this format: StartCapture_<number>.bat. The <number> values are generated and are not important. Our example had only one script for the State to Runtime Apply daemon, so we only edited and started StartCapture_12.bat.

4. Edit the final line in the file to append the string pwdfile="password.aut" so that the line now reads as shown (your example will differ if you configured a different log location):

```
db2cmd asncap CAPTURE_SERVER=RUNTIME CAPTURE_SCHEMA=CAPTURE_18  
CAPTURE_PATH="C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture"  
pwdfile="password.aut"
```

Note: This entire command should take up just one line.

5. Save the file and close Notepad.
6. Enter the following command to start the daemon:

```
StartCapture_5.bat
```

Figure 14-19 on page 376 shows the commands we entered.

```

C:\DB2 CLP
C:\Documents and Settings\Administrator>mkdir C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture\
C:\Documents and Settings\Administrator>cd C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture\
C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture>asnpwd INIT encrypt all using password.aut
2006-02-24-16.48.29.890000 ASN19811 "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".
C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture>asnpwd ADD alias STATE id db2admin password monPa55w0rd using password.aut
2006-02-24-16.48.37.671000 ASN19811 "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".
C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture>cd C:\DashboardReplication\Runtime\Runtime_to_Historical\source\
C:\DashboardReplication\Runtime\Runtime_to_Historical\source>notepad StartCapture_5.bat
C:\DashboardReplication\Runtime\Runtime_to_Historical\source>StartCapture_5.bat
C:\DashboardReplication\Runtime\Runtime_to_Historical\source>set DB2CODEPAGE=1208
C:\DashboardReplication\Runtime\Runtime_to_Historical\source>db2cmd asncap CAPTURE_SERVER=RUNTIME CAPTURE_SCHEMA=CAPTURE_18 CAPTURE_PATH="C:\IBM\WebSphere\Monitor\rm\logs\r2h\capture" pwdfile="password.aut"
C:\DashboardReplication\Runtime\Runtime_to_Historical\source>_

```

Figure 14-19 Starting the Runtime to History Capture daemon

7. The daemon should open in a separate window resembling Figure 14-20.

```

C:\DB2 CLP - db2setcp.bat asncap CAPTURE_SERVER=RUNTIME CAPTURE_SCHEMA=CAPTURE_18 C...
2006-02-24-16.51.04.343000 ASN01001 CAPTURE "CAPTURE_18" : "WorkerThread".
The Capture program initialization is successful.
2006-02-24-16.51.04.343000 ASN01091 CAPTURE "CAPTURE_18" : "WorkerThread".
The Capture program has successfully initialized and is capturing data changes for "0" registrations. "0" registrations are in a stopped state. "7" registrations are in an inactive state.
_

```

Figure 14-20 Runtime to History Capture daemon successfully started

Starting the Runtime to History Apply daemon

This daemon was started on the Dashboard Client because this is where the Runtime database was created. See Table 14-1 on page 365 for an explanation about where we chose to run the daemons.

1. Open a DB2 Command Window on the Dashboard Client by clicking **Start** → **Run**. Enter db2cmd and click **OK**.
2. Enter the following commands as shown in Figure 14-15 on page 372.

```

mkdir C:\IBM\WebSphere\Monitor\rm\logs\r2h\apply\
cd C:\IBM\WebSphere\Monitor\rm\logs\r2h\apply\
asnpwd INIT encrypt all using password.aut
asnpwd ADD alias STATE id db2admin password monPa55w0rd using password.aut

```

This creates an encrypted user ID and password file for starting the daemons. Be sure to replace db2admin and monPa55w0rd with your own user ID and password values.

As you can see above, it was also necessary to create the log directory, C:\IBM\WebSphere\Monitor\rm\logs\r2h\apply, using a `mkdir` command because this directory did not already exist on the Dashboard Client.

Note: This log directory is user-specified in the administrative console schema generation configuration settings. If you specified a different directory, create a log directory that is appropriate for your configuration.

3. Before starting the daemon, edit the start script to tell it to use the password file you just created. Enter the following commands to edit the file:

```
cd C:\DashboardReplication\History\Runtime_to_Historical\target\  
notepad StartApply_12.bat
```

Note: Each type of replication uses a set of one or more daemons. You should check the contents of the directory to find out how many daemons were generated. Each daemon name will have this format: StartApply_<number>.bat. The <number> values are generated and are not important. In our example there was only one script for the State to Runtime Apply daemon, so we only edited and started StartApply_12.bat.

4. Edit the final line in the file to append the string `pwdfile="password.aut"` so that the line now reads as shown (your example will differ if you configured a different log location):

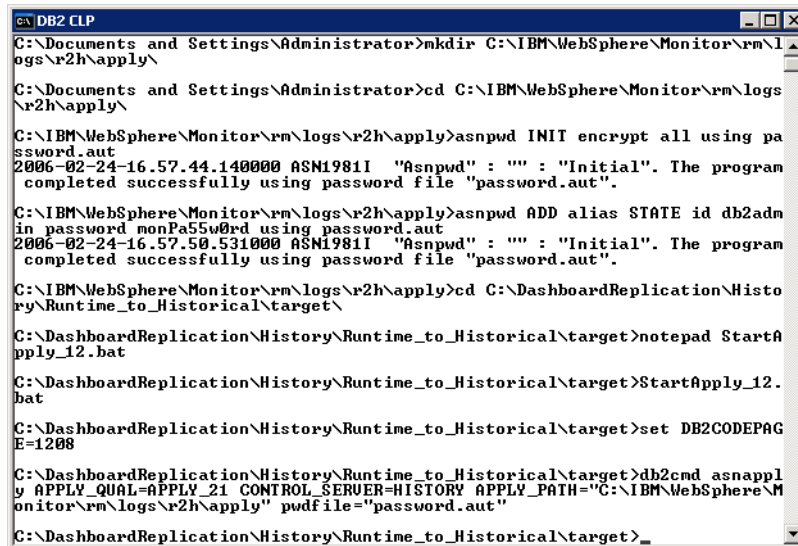
```
db2cmd asnapply APPLY_QUAL=APPLY_21 CONTROL_SERVER=HISTORY  
APPLY_PATH="C:\IBM\WebSphere\Monitor\rm\logs\r2h\apply"  
pwdfile="password.aut"
```

Note: This entire command should take up just one line.

5. Save the file and close Notepad.
6. Enter the following command to start the daemon:

```
StartApply_12.bat
```

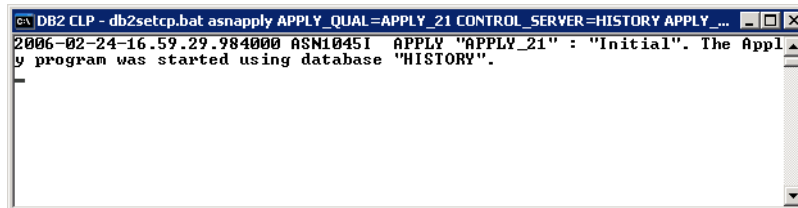
Figure 14-21 on page 378 shows the commands we entered.



```
C:\Documents and Settings\Administrator>mkdir C:\IBM\WebSphere\Monitor\rn\logs\r2h\apply\
C:\Documents and Settings\Administrator>cd C:\IBM\WebSphere\Monitor\rn\logs\r2h\apply\
C:\IBM\WebSphere\Monitor\rn\logs\r2h\apply>asnpwd INIT encrypt all using password.aut
2006-02-24-16.57.44.140000 ASN19811 "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".
C:\IBM\WebSphere\Monitor\rn\logs\r2h\apply>asnpwd ADD alias STATE id db2adm in password monPa55w0rd using password.aut
2006-02-24-16.57.50.531000 ASN19811 "Asnpwd" : "" : "Initial". The program completed successfully using password file "password.aut".
C:\IBM\WebSphere\Monitor\rn\logs\r2h\apply>cd C:\DashboardReplication\History\Runtime_to_Historical\target\
C:\DashboardReplication\History\Runtime_to_Historical\target>notepad StartApply_12.bat
C:\DashboardReplication\History\Runtime_to_Historical\target>StartApply_12.bat
C:\DashboardReplication\History\Runtime_to_Historical\target>set DB2CODEPAGE=1208
C:\DashboardReplication\History\Runtime_to_Historical\target>db2cmd asnapply APPLY_QUAL=APPLY_21 CONTROL_SERVER=HISTORY APPLY_PATH="C:\IBM\WebSphere\Monitor\rn\logs\r2h\apply" pwdfile="password.aut"
C:\DashboardReplication\History\Runtime_to_Historical\target>
```

Figure 14-21 Starting the Runtime to History Capture daemon

7. The daemon should open in a separate window resembling Figure 14-22.



```
C:\ DB2 CLP - db2setcp.bat asnapply APPLY_QUAL=APPLY_21 CONTROL_SERVER=HISTORY APPLY_...
2006-02-24-16.59.29.984000 ASN10451 APPLY "APPLY_21" : "Initial". The Apply program was started using database "HISTORY".
```

Figure 14-22 Runtime to History Capture daemon successfully started

Important: The daemons are now running and you must leave them running for WebSphere Business Monitor to work correctly.

By default they are configured to have a limit on how much *lag* there can be between running the Capture daemons. Therefore, if you stop them running for a longer period of time than the default (7 days), a problem will occur when you restart them. Read more and find out how to resolve this problem at:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.btools.help.monitor.doc/Doc/tasks/dbServices/configdmse rvopts.html>

Starting and stopping the daemons

After the initial configuration of the daemons is complete, it becomes simpler to stop and start them. Each StartCapture or StartApply batch file has a corresponding StopCapture or StopApply. To stop the daemons, run each of the Stop batch files. To start them again, you only have to run the Start batch files and do not have to re-create any of the directories or password files.

14.2.5 Cube configuration

This section describes how we created and configured the cubes for our process model.

Importing the cube definitions to DB2 Cube Views

To import the cube definitions into OLAP Center:

1. On the Dashboard Client, open OLAP Center using **Start → Programs → IBM DB2 → Business Intelligence Tools → OLAP Center**.
2. When prompted for DB2 Database Connection information, select the **HISTORY** database and enter the DB2 User ID and password. We used the default values, db2admin and monPa55w0rd as shown in Figure 14-23.

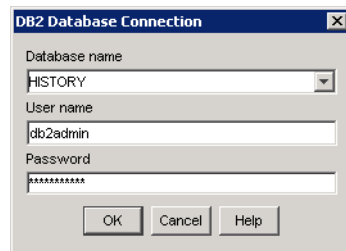


Figure 14-23 Connecting to the History database

3. Click **OK**.
4. When asked whether to configure the database for DB2 Cube Views, click **Yes** as shown in Figure 14-24.

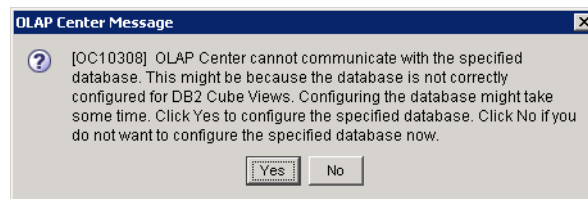


Figure 14-24 Configuring the History database for DB2 Cube Views

5. Wait for the process to complete, then select **OLAP Center** → **Import**.
6. Browse to the model_cv.xml file that was created by the schema generation and click **Next**. In our case this was located in C:\DashboardSchemaGen\schemagen\model_cv.xml.
7. Ensure that the option to **Replace existing objects** is selected. The window should resemble Figure 14-25. Click **Finish**.

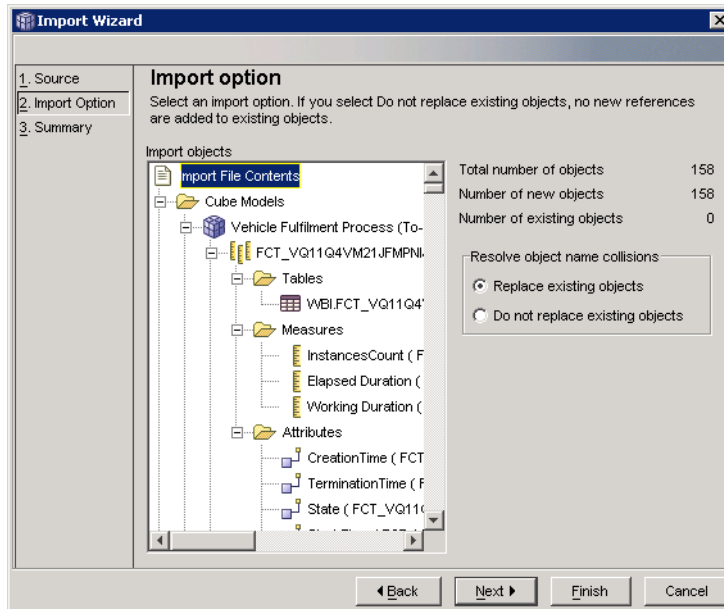


Figure 14-25 Importing the cube definitions

8. Your next window should resemble Figure 14-26.

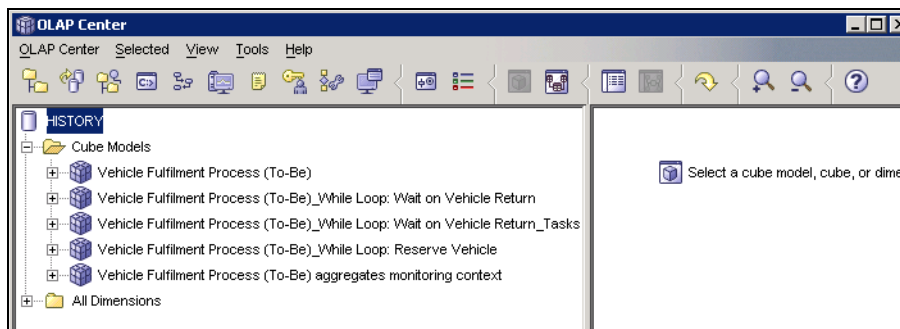


Figure 14-26 Completed cube import

9. Close OLAP Center.

Defining the cubes in DB2 Alphablox

The following steps show how to define the cubes to DB2 Alphablox.

1. On the Dashboard Client, ensure that the server is started, then open the DB2 Alphablox administration console and log in as described in Appendix B, “WebSphere Business Monitor system administration” on page 477.
2. Click the **Administration** tab, then click the **Cubes** tab at the top of the page.
3. Click **Create** to start the creation of a new cube.
4. The following sequence of steps should be repeated for every cube in the model. You can see the cubes listed in OLAP Center in Figure 14-26 on page 380.
 - a. Check the **Enabled** box to the right of the DB2 Alphablox Cube Name.
 - b. Select **HISTORY** for the Relational Data Source.
 - c. Check **Enable DB2 Cube Views Settings**.
 - d. From the Cube Model pull-down list, select the cube model **WBI.Vehicle Fulfilment Process ToBe**.
 - e. Click **Import Cube Definition**.
 - f. The section on the left updates with measures and dimensions.
 - g. Change the DB2 Alphablox Cube Name to **Vehicle Fulfilment Process ToBe**.
 - h. The completed dialog should resemble Figure 14-27 on page 382. Click **OK**.

Important: Be very careful when specifying the DB2 Alphablox Cube Name: It should match the Cube Model selection name exactly, except that you should omit the **WBI .** prefix. This includes matching all underscore characters, spaces, and case sensitivity.

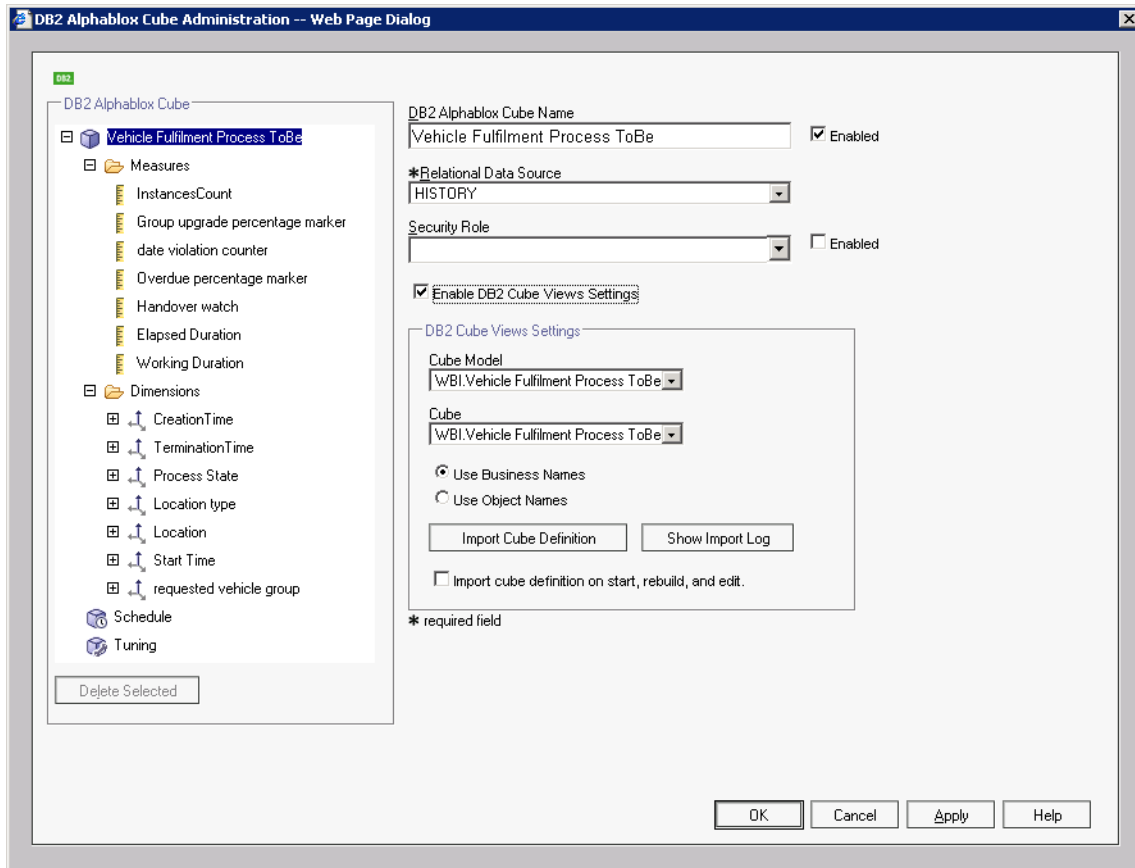


Figure 14-27 Defining the DB2 Alphablox Cube

- Repeat step 4 for each of the other cubes until you have configured all that are shown in Table 14-3. Refer to the tip after the table if you have problems reading the process name in the selection pull-down lists.

Table 14-3 Cube model and DB2 Alphablox cube names

Cube Model Name	DB2 Alphablox Cube Name
WBI.Vehicle Fulfilment Process ToBe aggregates monitoring context	Vehicle Fulfilment Process ToBe aggregates monitoring context
WBI.Vehicle Fulfilment Process ToBe_Reserve Vehicle	Vehicle Fulfilment Process ToBe_Reserve Vehicle

Cube Model Name	DB2 Alphablox Cube Name
WBI.Vehicle Fulfilment Process ToBe_Wait on Vehicle Return	Vehicle Fulfilment Process ToBe_Wait on Vehicle Return
WBI.Vehicle Fulfilment Process ToBe_Wait on Vehicle Return_Tasks	Vehicle Fulfilment Process ToBe_Wait on Vehicle Return_Tasks

Tip: If your process has a long name like ours, the cube model name might not be entirely visible in the administrative dialog pull-down lists, therefore you will be unable to complete the DB2 Alphablox Cube Name in the definition.

To work out which cube model has been selected, first select the cube model from the list and then import it as described above. Then select the **Measures** leaf under New Cube in the tree on the left and note the value of the Measures Fact Table name as shown in Figure 14-28.

Now switch to OLAP Center (see “Importing the cube definitions to DB2 Cube Views” on page 379), expand each of the cube models, and locate the model that has the matching table name, as shown in Figure 14-29 on page 384. Make a note of this name. You might find it easiest to right-click the cube model and click **Properties**, then copy the name to the clipboard. This helps guarantee that you get the name exactly right.

Switch back to the DB2 Alphablox administrative dialog and click **New Cube** at the root of the tree on the left. Now complete the DB2 Alphablox Cube Name field using the cube model name you just determined and click **OK**.

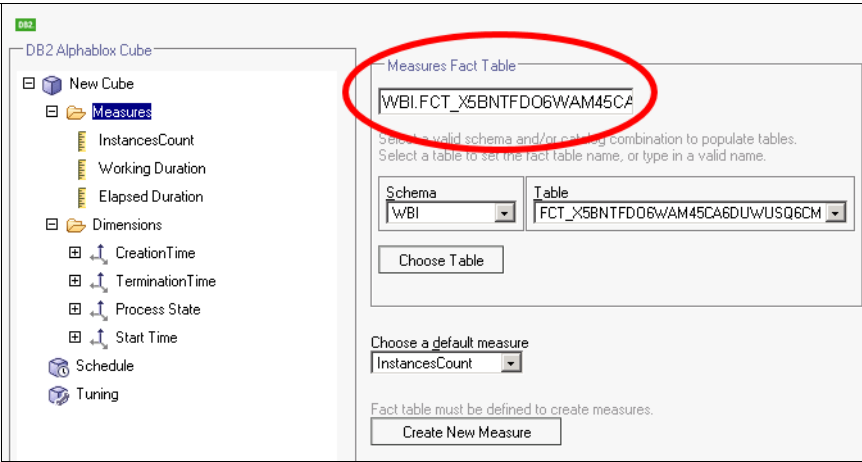


Figure 14-28 Locating the measures fact table name

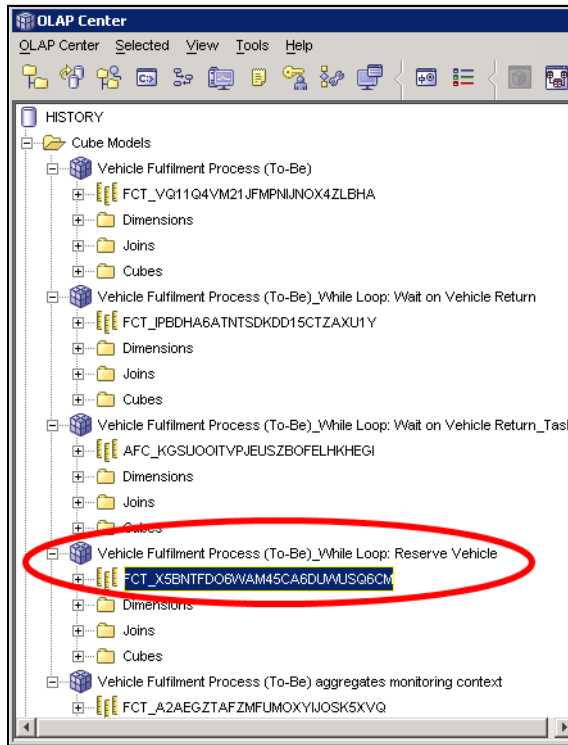


Figure 14-29 Determining the cube model name from the measures fact table name

6. You should see a list of DB2 Alphablox Cubes as shown in Figure 14-30.

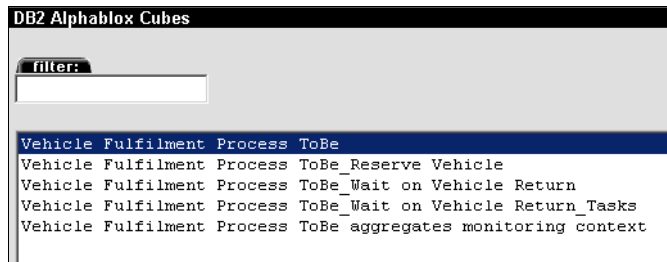


Figure 14-30 Completed DB2 Alphablox cube definitions

7. Click **General** from the tabs under Administration at the top, then click **DB2 Alphablox Cubes** under Runtime Management on the left.
8. Select the first cube and click **Start**.
9. Repeat the previous step for each of the cubes until all have been started as shown in Figure 14-31 on page 385.

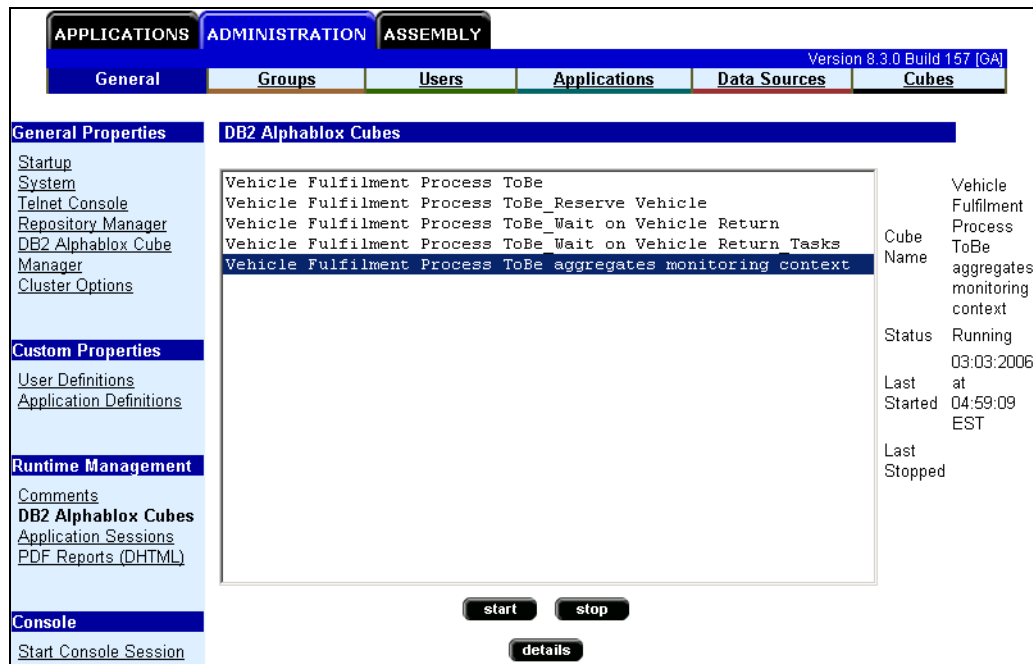


Figure 14-31 Starting the DB2 Alphablox cubes

10. Close the administrative console browser and OLAP Center.

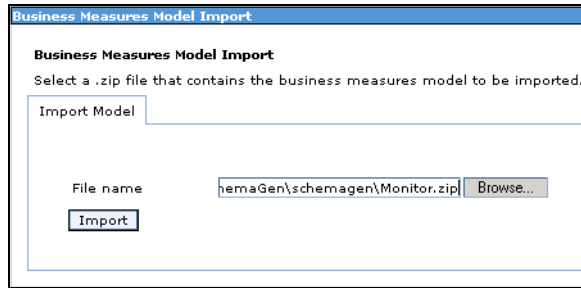
14.2.6 Model import

Now that the database and replication configuration is complete, import the model into WebSphere Business Monitor. This will cause WebSphere Business Monitor to begin capturing events coming from the WebSphere Process Server runtime and persisting the data into the State database. The replication daemons will then copy this data to the Runtime and History databases.

The following steps describe how we imported the ITSO Speedy Rentals process model into the WebSphere Business Monitor system:

1. Start the Monitor Server, then start the administrative console, as described in “Monitor Server administrative console” on page 478.
2. Expand **WebSphere Business Monitor** → **Server** → **Business Measures Model** and click **Model Import**.
3. Specify the location of the Business Measures Model that was created by the schema generation in the previous section. In our example this was C:\SchemaGen\schemagen\Monitor.zip, as shown in Figure 14-32.

Tip: The administrative console only allows you to upload a file from the local machine, so we ran the administrative console from our Monitor Server machine so that we could browse to the local file.

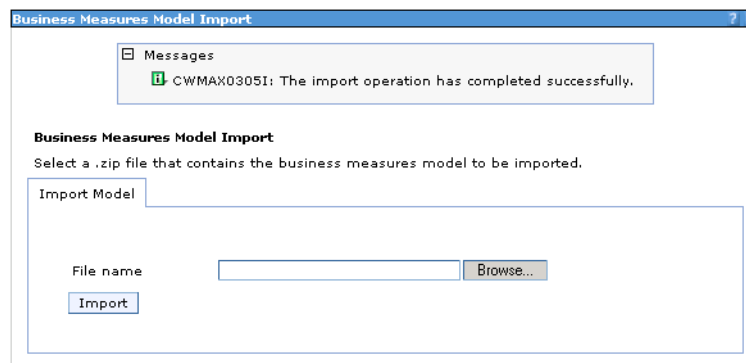


The image shows a dialog box titled "Business Measures Model Import". Inside, there is a section titled "Business Measures Model Import" with the instruction "Select a .zip file that contains the business measures model to be imported." Below this, there is a tab labeled "Import Model". Under the tab, there is a "File name" field containing the text "hemaGen\schemagen\Monitor.zip" and a "Browse..." button to its right. Below the file name field is an "Import" button.

Figure 14-32 Importing the model

Important: The file that you specify here is called Monitor.zip, which is the same name as the file that you originally exported from WebSphere Business Modeler and provided as input to the schema generator. However, the Monitor.zip file that you specify here for model import is different, and is found in the output directory of the schema generator: C:\SchemaGen\schemagen\ in our example.

4. Click **Import** and allow the import process time to complete (Figure 14-33). Refer to the log file for the Monitor Server if you encounter any error or warning messages, and to confirm that the import was successful.



The image shows the same "Business Measures Model Import" dialog box as in Figure 14-32, but now with a "Messages" pane at the top. The "Messages" pane has a checkbox labeled "Messages" which is checked, and a message box containing a green icon and the text "CWMAX0305I: The import operation has completed successfully." Below the messages pane, the "Business Measures Model Import" section is still visible, showing the "Import Model" tab, the "File name" field (which is now empty), the "Browse..." button, and the "Import" button.

Figure 14-33 Successful model import

The server is now actively listening for events from the ITSO Speedy Rentals application running on WebSphere Process Server.

14.2.7 Configuring the adaptive action manager

The action manager is responsible for capturing any situation events that were defined in the deployed business measures model, and taking actions such as producing alerts for display in the dashboard, producing e-mail alerts, and invoking Web services. We configured a situation event in our business measures model, then configured the action manager to capture this and produce alerts for display in our dashboard.

The action manager first requires some general configuration to tell it how to locate the user registry (LDAP configuration) or mail server (SMTP configuration). When this is complete, you can specify one or more notification templates. These define how a notification should be delivered (alert, cell phone, e-mail, pager). You can also define a Web service to invoke instead of an alert.

Next, define situation event bindings, which tell the action manager to listen for specific situation events. (Situation events themselves are defined using the business measures editor of WebSphere Business Modeler.)

Finally, add one or more notifications to each situation event binding. This causes the action manager to perform the notification whenever it receives a recognized situation event.

The following steps describe how we configured the action manager.

Important: To be able to configure the action manager to produce alerts, you must first configure your WebSphere Portal to use a user registry; typically a Lightweight Directory Access Protocol (LDAP) server such as IBM Tivoli Directory Server is used. We used the instructions in the WebSphere Portal information center to configure ours:

http://publib.boulder.ibm.com/infocenter/wpdoc/v510/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/intr_ldap.html

1. Start the Monitor Server, then start the administrative console, as described in “Monitor Server administrative console” on page 478.
2. Expand **WebSphere Business Monitor** → **Adaptive Action Manager**, and click **Configuration**.
3. Click the **LDAP** tab.
4. Configure the LDAP fields that are specific to your LDAP installation. Figure 14-34 on page 388 shows our examples.

The screenshot shows a configuration window with two tabs: 'General' and 'LDAP'. The 'LDAP' tab is active. Below the tabs is a section titled 'General Properties'. It contains several text input fields with labels to their left: 'LDAP URL:' with the value 'ldap://9.42.77.237:389', 'LDAP User ID:' with the value 'uid=wpsadmin,o=Admins,cn=', 'LDAP Password:' with masked characters '*****', 'LDAP Alert field:' with the value 'uid', 'LDAP Cell phone field:' with the value 'mobile', 'LDAP Email field:' with the value 'mail', and 'LDAP Pager field:' with the value 'pager'. At the bottom of the window are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

Figure 14-34 Configuring LDAP settings for action manager

5. Click **OK**.
6. Expand **WebSphere Business Monitor** → **Adaptive Action Manager** → **Template Definitions** and click **Notification**.
7. Click **New**.
8. Specify the Template name as **Overdue Alert**.
9. Change the action service type to **Alert**.
10. Enter an LDAP query that will resolve to the users whom you wish receive this alert. We used the following simple query that resolves to all users, so that all users would receive the alert:
`(&(objectclass=top)(objectClass=inetOrgPerson))`
11. Specify the subject as:
 Overdue limit violation
12. Specify the Body as:
 %Rental ID% exceeded overdue limit
13. The percentage symbols mark a substitution variable. For this to be replaced by a real value at runtime, the Reservation ID field must be defined as part of the situation event data in the business measures editor of WebSphere Business Modeler.
 %Reservation ID% exceeded the maximum number of overdue returns for a rental contract.
14. The completed notification template definition should resemble Figure 14-35 on page 389. Click **OK**.

* Template name:
Overdue Alert

Description:

Action service type:
☒ Alert
☐ Cell Phone
☐ Email
☐ Pager

To (LDAP Query):
(&(objectclass=top))(objectCls

Subject:
Overdue limit violation

Body:
%Rental ID% exceeded the maximum number of overdue returns for a rental contract.

LDAP Root:

Apply OK Reset Cancel

Figure 14-35 Completed notification template definition

15. Expand **WebSphere Business Monitor** → **Adaptive Action Manager** and click **Installed Situation Event Binding**.
16. Click **New**.
17. Specify the Situation Event name as `MaxOverdueCountExceeded`. This name must match the value specified for the `BusinessSituationName` field of the situation event that was defined in the business measures editor of WebSphere Business Modeler.
18. Click **OK**.
19. Click the new situation event binding.
20. Click **Add**.
21. Ensure that **Overdue Alert** is selected as the Template name and click **OK**.
22. Figure 14-36 on page 390 shows the completed situation event binding. Click **OK**.

General Properties

* Situation Event name:

Description:

☒ Preferences

Select	Template Name	Action Service Type
<input type="checkbox"/>	Overdue Alert	AlertHandler

Total 1

Figure 14-36 Completed situation event binding

23. In the tree on the left, expand **Applications** and click **Enterprise Applications**.
24. Check the IBM_WB_ACTIONMANAGER application and click **Stop**.
25. Check the IBM_WB_ACTIONMANAGER application again, and click **Start**.
26. Log out from the administrative console.

14.3 Testing the business measures model deployment

It is a good idea to test the business measures deployment at this stage. You can do this by following the steps described in the next sections, but it is useful to perform a quicker check at this point. Here are the steps we took for checking the model deployment:

1. Ensure that the implemented process model is deployed to your WebSphere Process Server environment.
2. Ensure that the business measures model is deployed to your WebSphere Business Monitor environment.
3. Ensure that all four sets of replication daemons are running for your business measures model.
4. Ensure that you have imported the business measures model into WebSphere Business Monitor.
5. Open DB2 Control Center on a machine that has access to the State, Runtime, and History databases. (All of the databases must be either local or remotely cataloged.) In our case we opened DB2 Control Center on the

Dashboard Client because this is where we installed the Runtime and History databases, and we also had the State database cataloged there.

6. Start a process instance on the WebSphere Process Server environment. The BPC Explorer is the easiest way to do this.
7. Check the log files for the Monitor Server machine of the WebSphere Business Monitor installation. (Refer to “Monitor Server log files” on page 478 to locate the log files.) There should be no exceptions.
8. Check the contents of the WebSphere Business Monitor databases. Each process instance that you start should initially create a row in a table of the State database. This row is then replicated into a table in the Runtime database and deleted from the State database. Finally, the row is replicated to a table in the History database and deleted from the Runtime database. The time it takes to replicate between each database depends on the intervals you specified when configuring schema generation. In our case, we specified 2 minutes as the interval for both replications.

Note: You will have to look in all three databases to verify whether the process instance has been captured. This is because it might or might not have begun replicating among the three databases depending on how much time has elapsed since you ran the process instance.

9. To determine which tables to look at in the databases, look for the table with the comment that matches the name of the process. Figure 14-37 on page 392 shows an example of checking the History database. In the example, the comment is Vehicle Fulfilment Process ToBe.

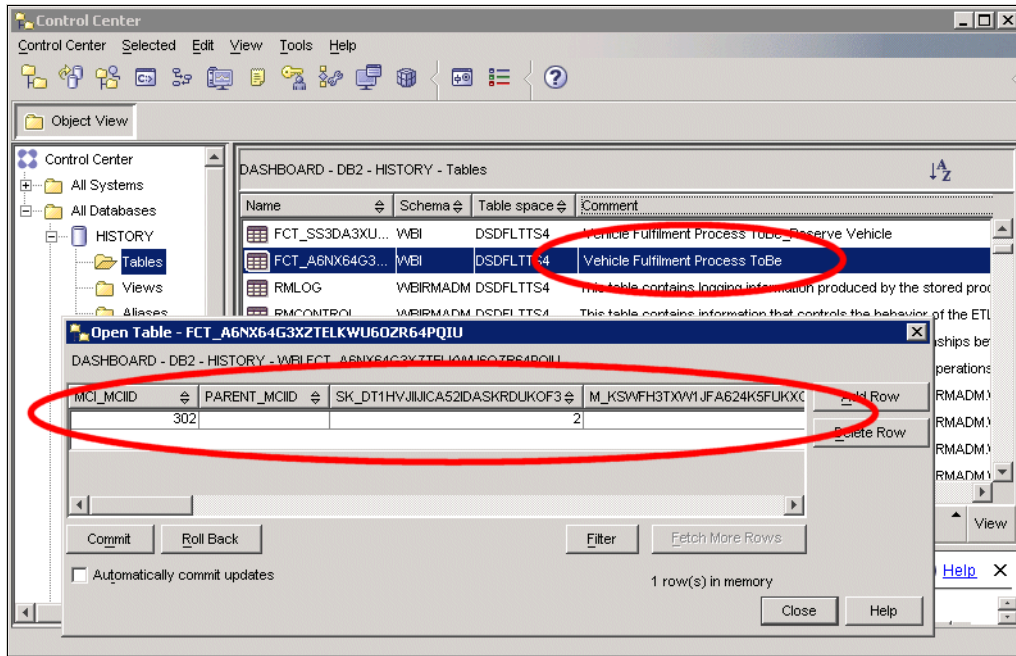


Figure 14-37 Checking the contents of the History database

10. You should see a row for each instance of the process that you have executed, and the row should appear eventually in the History database after the replication has completed. By waiting for a row to appear in the History database, you can verify that the replication is working, as well as the initial capturing of the data.

14.4 Creating the dashboards

This section describes how we configured a dashboard to present the business monitoring information. We built our dashboard using multiple pages with each page displaying one of more pieces of information.

14.4.1 Copying the dashboard view template portlets

The template dashboard views must be copied before use. The copies can then be customized for a specific business measures model.

Important: It is not always possible to reverse the customization of a dashboard view portlet after it has been configured for a specific business measures model. Therefore, it is essential that you copy the portlet templates and never use the original templates when creating a dashboard page. If you customize the originals, you might not be able to create new dashboard views for other business measures models without first reinstalling the Dashboard Client to obtain the original templates.

We followed these steps to copy the templates for the ITSO Speedy Rentals business measures model:

1. Start the Dashboard Client server and open the administrative console using Internet Explorer. Refer to “Dashboard Client administrative console” on page 479 if you do not know how to do this.

Restriction: Although WebSphere Portal supports use of other browsers, some of the WebSphere Business Monitor features and dashboard views require you to use Internet Explorer.

2. Log in to the console. By default, wpsadmin is both user ID and password.
3. Click the **Administration** link at the top of the administrative console.
4. Expand **Portlet Management**, then click **Portlets**.
5. This opens the list of portlets that are available. Only some of these portlets are provided by WebSphere Business Monitor; these are found at the end of the list. The list spans many pages but there are navigational tools that assist with moving between pages, as highlighted in Figure 14-38 on page 394.

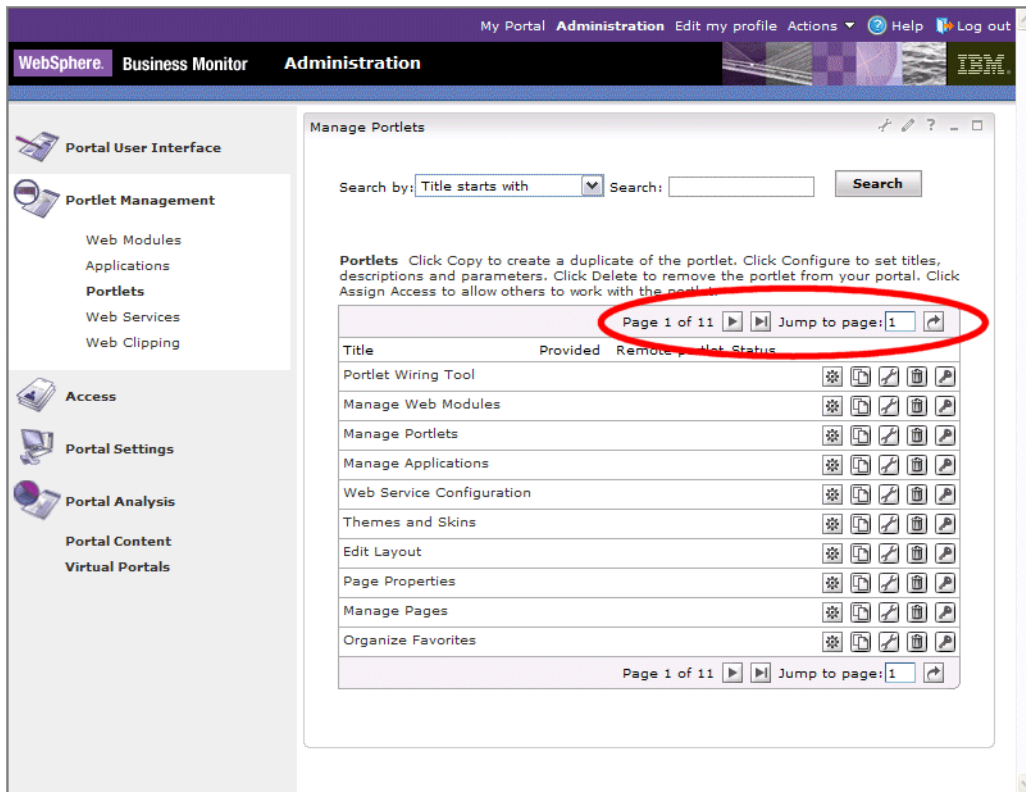



Figure 14-38 Navigate the list of portlets using the navigation tools

6. Enter 10 in the **Jump to page** field and click the arrow. Page 10 is where the first WebSphere Business Monitor portlets are found. You should see Alerts and Dimensions at the bottom of the list.
7. Copy the Alerts portlet:
 - a. Click the Copy Portlet icon  to the right of the Alerts portlet.
 - b. Specify a new name for the portlet as shown in Figure 14-39 on page 395. We used ITS0 Speedy Rentals Alerts.
 - c. Click **OK**.

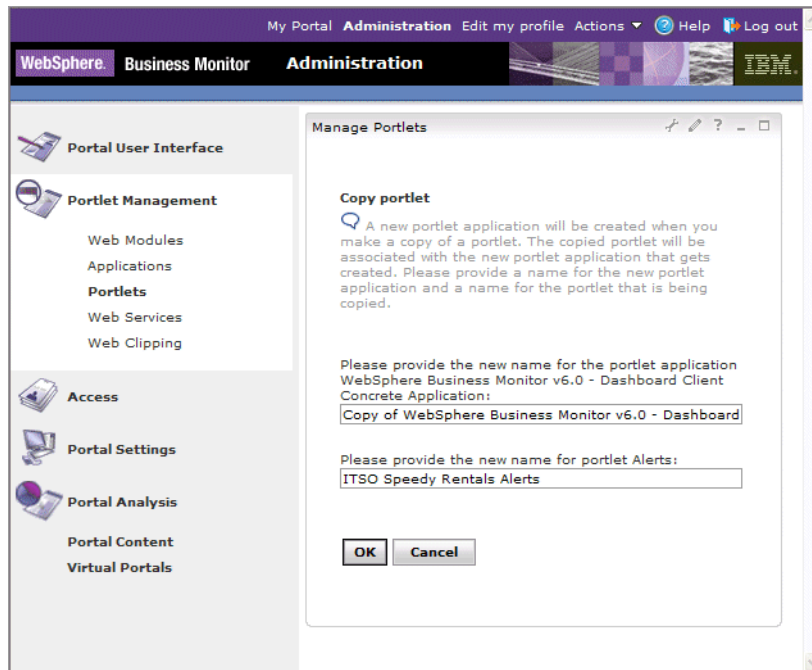


Figure 14-39 Creating a portlet copy

8. Use the same technique to create copies of all other portlets, referring to Table 14-4 for the names to use. Use the navigation tools to move to the other pages in the portlets list.

Table 14-4 Copying of portlets

Original Name	Copy Name
Alerts	ITSO Speedy Rentals Alerts
Dimensions	ITSO Speedy Rentals Dimensions
Export Values	ITSO Speedy Rentals Export Values
Gauges	ITSO Speedy Rentals Gauges
Key performance indicators (KPIs)	ITSO Speedy Rentals Key Performance Indicators (KPIs)
Active Instances	ITSO Speedy Rentals Active Instances
Organizations	ITSO Speedy Rentals Organizations
Process Diagrams	ITSO Speedy Rentals Process Diagrams

Original Name	Copy Name
Reports	ITSO Speedy Rentals Reports
Scorecards	ITSO Speedy Rentals Scorecards
Active Instances	ITSO Speedy Rentals Aggregated Values

Note: It is not necessary to copy the Welcome or Getting Started templates because they are not customizable.

9. The final list of portlets should resemble Figure 14-40.

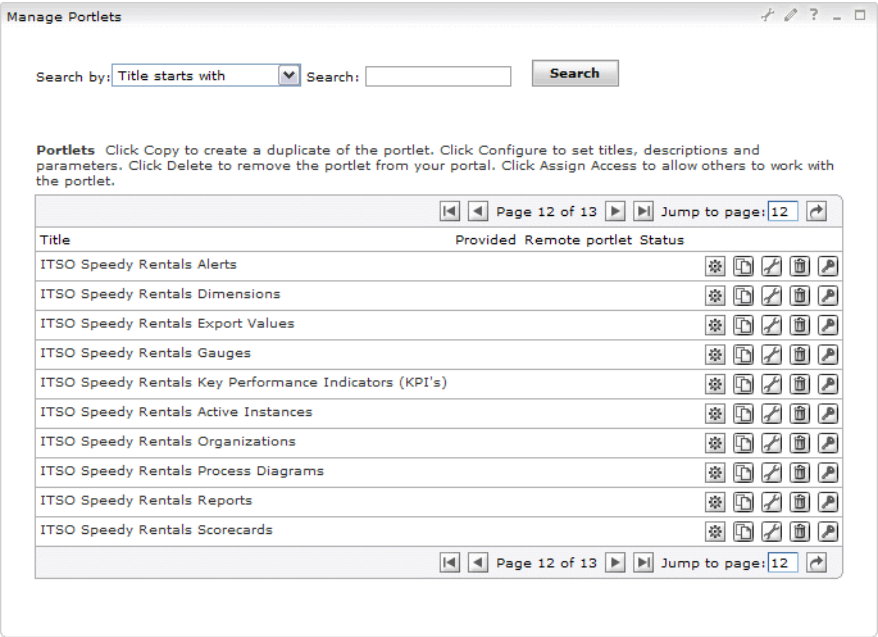



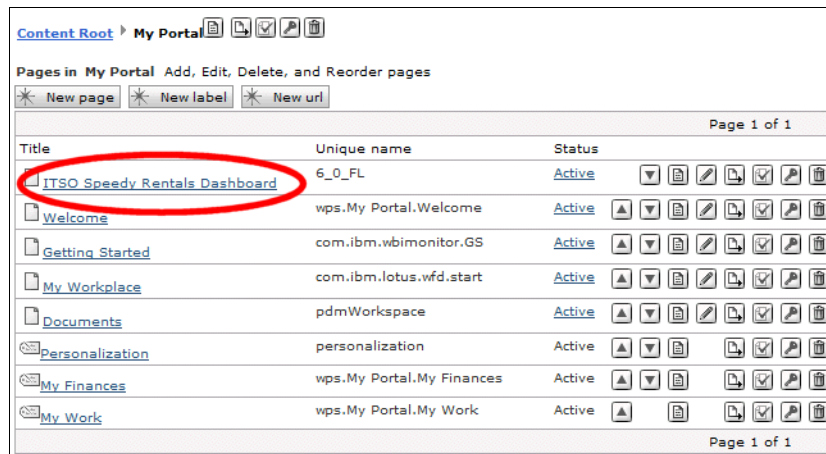
Figure 14-40 Copied dashboard view portlets

Attention: The second copy of the Active Instances portlet is actually on the next portlet page thus does not appear in Figure 14-40. Be sure that you have this additional portlet copy named ITSO Speedy Rentals Aggregated Values.

14.4.2 Creating the dashboard pages

Now create some pages for the dashboard and add the different portlets to those pages. The following steps describe how to do this:

1. Log in to the Dashboard Client's administrative console as before and click the **Administration** link.
2. Click **Portal User Interface** on the left, then click **Manage Pages**.
3. The list of pages at the content root level is displayed. These pages are part of the overall structure of the Portal installation. Click **My Portal** in the list to show the actual pages that a user will see.
4. Click the New page icon.
5. Specify the title of the page as **ITSO Speedy Rentals Dashboard** and click **OK**.
6. Click **OK** again to acknowledge the successful creation of the page.
7. The page appears at the bottom of the list. Repeatedly click the up arrow  for the new page to move it to the top of the list as shown in Figure 14-41.











































Title	Unique name	Status	
<u>ITSO Speedy Rentals Dashboard</u>	6_0_FL	Active	    
<u>Welcome</u>	wps.My Portal.Welcome	Active	    
<u>Getting Started</u>	com.ibm.wbimonitor.GS	Active	    
<u>My Workplace</u>	com.ibm.lotus.wfd.start	Active	    
<u>Documents</u>	pdmWorkspace	Active	    
<u>Personalization</u>	personalization	Active	    
<u>My Finances</u>	wps.My Portal.My Finances	Active	    
<u>My Work</u>	wps.My Portal.My Work	Active	    

Figure 14-41 Adding the main dashboard page

8. Click the new page title to show the subpages under the main dashboard page. Currently none are listed, so click the New page icon to add the first subpage.
9. Specify the name as **Executive Overview** and click **OK** twice.
10. Repeat the procedure to add additional pages with the following names:
 - Aggregated Analysis
 - Current Rentals
 - Business Analysis

- Historical Trends
- Operations
- Control Center

11. The final result should be as shown in Figure 14-42.

Title	Unique name	Status	
Executive Overview	6_0_FM	Active	
Aggregated Analysis	6_0_FN	Active	
Current Rentals	6_0_FO	Active	
Business Analysis	6_0_FP	Active	
Historical Trends	6_0_FQ	Active	
Operations	6_0_FR	Active	
Control Centre	6_0_FS	Active	

Figure 14-42 Completed dashboard pages

14.4.3 Creating the dashboard pages

Add the portlets you created previously to the new pages. The following steps describe how to do this:

1. Click the Edit Page Layout icon for the Executive Overview page. The icon resembles a pencil.
2. Click the One column layout icon (the first layout from the left), then click **OK** to change the layout to a single column.
3. Click **Add portlets**.
4. Use the navigator tools to locate the following portlets and select them all before clicking **OK**. The result should resemble Figure 14-43 on page 399.
 - ITSO Speedy Rentals Gauges
 - ITSO Speedy Rentals Key Performance Indicators (KPIs)
 - ITSO Speedy Rentals Scorecards
5. Click **Done**.

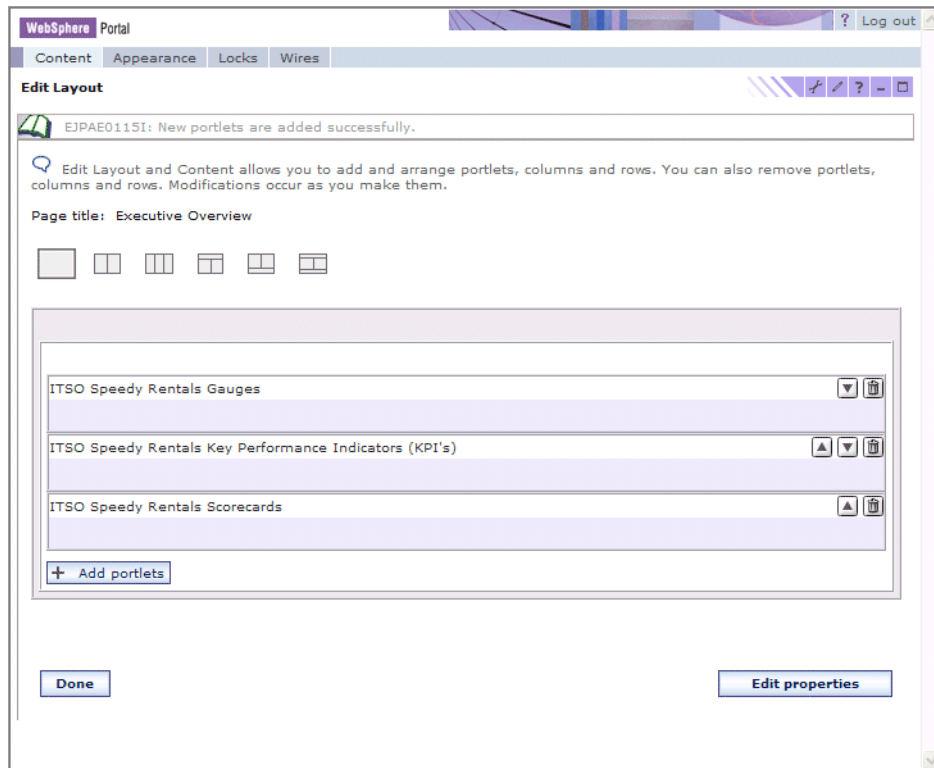


Figure 14-43 Adding portlets to the Executive Overview page

6. Use the same procedure to change the following pages to a single-column layout and add the corresponding portlets to the page (Table 14-5).


Table 14-5 Pages to change and portlets to add

Page to change to single-column layout	Portlet to add to the page
Aggregated Analysis	ITSO Speedy Rentals Aggregated Values
Current Rentals	ITSO Speedy Rentals Alerts ITSO Speedy Rentals Active Instances ITSO Speedy Rentals Process Diagrams
Business Analysis	ITSO Speedy Rentals Dimensions
Historical Trends	ITSO Speedy Rentals Reports
Operations	ITSO Speedy Rentals Organizations
Control Center	ITSO Speedy Rentals Export Values


- Finally, click the Edit Page Layout icon at the very top (to the right of the breadcrumb trail) to edit the parent ITSO Speedy Rentals Dashboard page.
- Change the page layout to a single column and add the **Welcome** portlet to the page.

Wiring cooperative portlets together

Some of the WebSphere Business Monitor portlets can cooperate with each other so that when the user clicks something in one portlet, another portlet is updated. To achieve this, the two interacting portlets must be wired together, and this is done by defining wires in the page administration. We configured three of the possible portlet interactions by following these steps:

- Return to the list of pages in the administrative console.
- Click the Edit Page Layout icon  for the Current Rentals page. The icon resembles a pencil.
- Click the **Wires** link at the very top of the page to display the currently defined wires. Make the following selections:

Source portlet	ITSO Speedy Rentals Alerts
Sending	Instance Information - OUT
Target page	Current Rentals
Target portlet	ITSO Speedy Rentals Active Instances
Receiving	Instance Information - IN
Wire Type	Public

- Click the add icon  to add the new wire as shown in Figure 14-44.

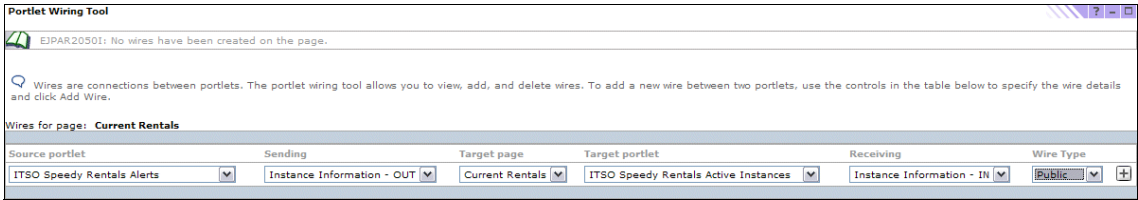


Figure 14-44 Creating a wire

- Repeat the previous steps, changing the source and target each time, to create the two additional wires described in Table 14-6.

Table 14-6 Wires between portlets

Source portlet	Target portlet
ITSO Speedy Rentals Active Alerts	ITSO Speedy Rentals Process Diagrams
ITSO Speedy Rentals Active Instances	ITSO Speedy Rentals Process Diagrams

6. The final list of wires should resemble Figure 14-45. Click **Done**.

Wires for page: Current Rentals					
Source portlet	Sending	Target page	Target portlet	Receiving	Wire Type
ITSO Speedy Rentals Alerts	Instance Information - OUT	Current Rentals	ITSO Speedy Rentals Active Instances	Instance Information - IN	Public
ITSO Speedy Rentals Alerts	Instance Information - OUT	Current Rentals	ITSO Speedy Rentals Process Diagrams	Instance Information - IN	Public
ITSO Speedy Rentals Active Instances	Instance Information - OUT	Current Rentals	ITSO Speedy Rentals Process Diagrams	Instance Information - IN	Public
Select One ...		Current Rentals			Personal

Figure 14-45 Completed co-operative portlet wires

The page layouts and contents are all completed.

14.4.4 Configuring dashboard views to show data

The dashboard page structure is now complete, and each of the WebSphere Business Monitor portlets have been arranged on the individual dashboard pages. Next, customize each of those portlets to show data for the ITSO Speedy Rentals business process.

Important: Before you work on this section, we recommend that you run several processes to put real data into the system for each view to display. This helps ensure that the views look meaningful when first configured. However, you are not required to do this, and you can configure all the views before you run any instances of the process.

Our screen captures all show data that was produced by running some instances of the ITSO Speedy Rentals process.

The following steps describe what we did:

1. Log in to the Dashboard Client administrative console if you are not already logged in.
2. Click the **My Portal** link at the top. The welcome page for WebSphere Business Monitor should be displayed first in the ITSO Speedy Rentals Dashboard, as shown in Figure 14-46 on page 402.

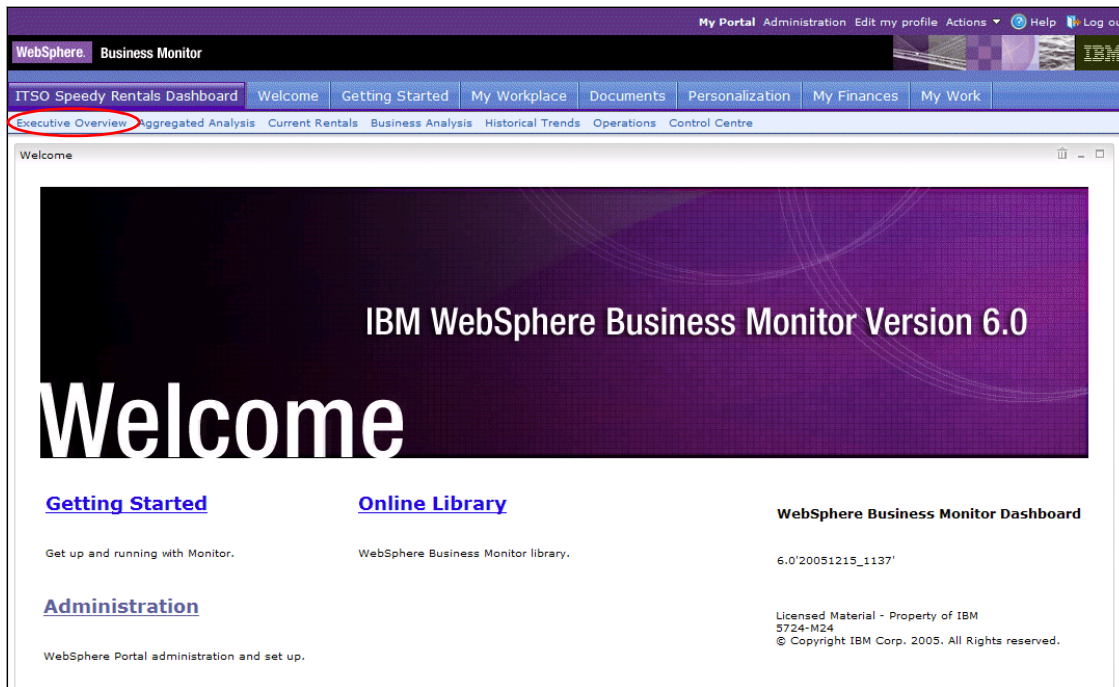



Figure 14-46 ITSO Speedy Rentals welcome page

3. Click the **Executive Overview** page link at the top to display that page.
4. You should see the three portlets that were added before. None of them have been configured yet, so they show only an informational message and do not show any data related to the ITSO Speedy Rentals business process.

Configuring the Gauges view

The first portlet to configure on the page is the Gauges view. Follow these steps:

1. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Gauges portlet to open the configuration page.
2. Expand the Business Measures Model tree and click **ITSO Speedy Rentals Process Catalog**.
3. Select the two KPIs, **Average number of late rental returns per rental** and **Percentage of rentals with late returns** as shown in Figure 14-47 on page 403. Click **Next**.

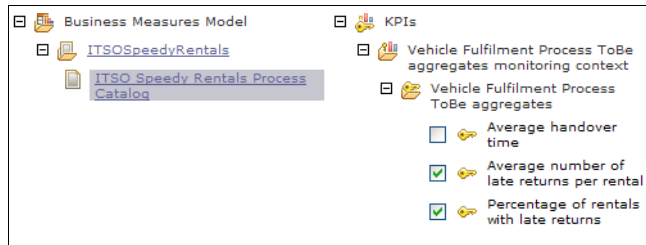


Figure 14-47 Configuring the Gauges view

- Specify minimum and maximum values for the range of each gauge. We used the values shown in Table 14-7.

Table 14-7 Gauge minimum and maximums

KPI	Gauge minimum	Gauge maximum
Average number of late rental returns per rental	0	2.0
Percentage of rentals with late returns	0	100

- Click **Next**, then **Finish**. The completed view should resemble Figure 14-48.

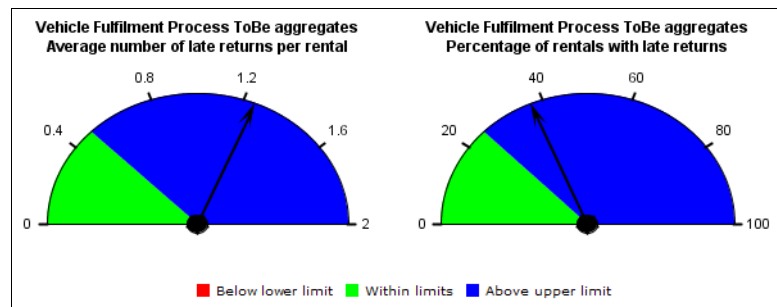



Figure 14-48 Completed gauges view

Configuring the key performance indicators view

The next portlet to configure on the page is the key performance indicators (KPI) view. Use the following steps:

- Click the Configure icon  to the upper right of the ITSOSpeedyRentals Key Performance Indicators (KPIs) portlet to open the configuration page.
- Expand **Business Measures Model** → **ITSOSpeedyRentals** and click **ITSOSpeedyRentals Process Catalog**.

3. Select all of the KPIs as shown in Figure 14-49. Click **Next**.

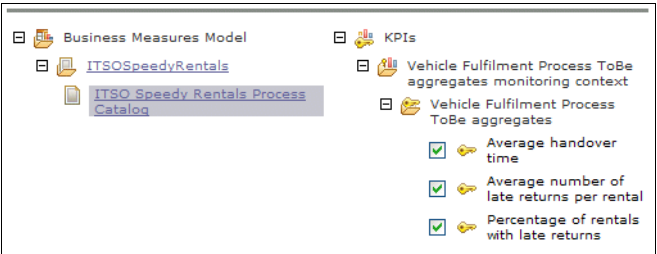


Figure 14-49 Selecting KPIs to display in key performance indicators view

4. (Optional) Change the icons by clicking the palette icon. You can also remove the legend and the help text if you wish. When you are through, click **Next**.
5. Click **Finish**.
6. The view should resemble Figure 14-50.


KPI	↑↓	Status ↑↓	Value
Vehicle Fulfilment Process ToBe aggregates.Average handover time		🕒	13 m, 2 s
Vehicle Fulfilment Process ToBe aggregates.Average number of late returns per rental		🔴↑	1.25
Vehicle Fulfilment Process ToBe aggregates.Percentage of rentals with late returns		🔴↑	37,5

🟡 Below limit 🟢 Within limits 🔴↑ Above limit 🕒 Not Available

Figure 14-50 Completed KPI view

Configuring the scorecards view

The next portlet to configure on the page is the scorecards view:

1. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Scorecards portlet to open the configuration page.
2. The default perspectives are shown. Select the last two, **Learning & Growth** and **Customer**, and click **Delete Perspective**. Click **Yes** when prompted to confirm the deletion.
3. Select the **Financial** perspective and click **Edit Perspective**.
4. Change the Perspective name to Efficiency and change the Description to Provides scores related to performance of the rental process.
5. Click **OK** (be careful not to click Next).
6. Perform the same steps to rename the **Internal Business Process** perspective to Overdue Rentals with a description of Provides scores related to overdue rentals.

7. Two perspectives should be seen as shown in Figure 14-51. Click **Next**.

<input type="checkbox"/>	Perspective	Description
<input type="checkbox"/>	Efficiency	Provides scores related to performance of the rental process
<input type="checkbox"/>	Overdue Rentals	Provides scores related to overdue rentals

Figure 14-51 Editing the perspective names

- Expand **Business Measures Model** → **ITSOSpeedyRentals** and click **ITSO Speedy Rentals Process Catalog**.
- Select both the **Average number of late returns per rental** and **Percentage of rentals with late returns** KPIs.
- Select the **Overdue Rentals** perspective from the pull-down list. Click **Add**.
- Use the same technique to add the **Average handover time** KPI to the Efficiency perspective.
- Click **Next**.
- (Optional) Deselect one of the columns. We left all enabled.
- (Optional) Change the icons by clicking the palette icon or the layout using the selection options. You can also remove the legend and the help text if you wish. When you are finished, click **Next**.
- Click **Finish**. The view should resemble Figure 14-52.

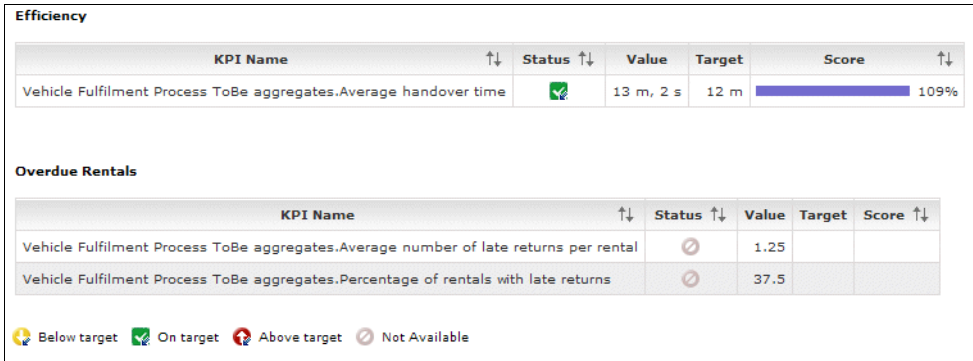



Figure 14-52 Completed scorecards view

Note: The scorecard view is best suited to KPIs that were defined in the business measures editor with the Use Target option. In our example, only the Average handover time was defined in this way. Therefore, only this KPI makes full use of the scorecard's ability to show performance against targets. The other KPIs (that were defined with limits) have no status, target, or score values because these require a target for them to be calculated.

Configuring the Active Instances view for aggregated values

The next portlet to configure is on the next dashboard page, Aggregated Analysis. The portlet on this page is a copy of the Active Instances portlet. Despite its name, it is possible to configure this view to show aggregated measures for completed processes. That is why we made two copies of the view: one for configuring to show active instances data, the other to show aggregated measures for all of the completed instances.

1. Click the **Aggregated Analysis** page link at the top to switch to that page.
2. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Aggregated Values portlet to open the configuration page.
3. Expand **Business Measures Model** → **ITSOSpeedyRentals** and click **ITSO Speedy Rentals Process Catalog**.
4. Click **Vehicle Fulfilment Process ToBe aggregates monitoring context** in the Business Measures Groups section. Choosing this group ensures that the view will be configured to show aggregated data for completed instances instead of individual data for active instances.
5. Select the aggregate business measures that you want to display. As shown in Figure 14-53, we selected all of the measures except key_definition, which is a meaningless built-in measurement. Click **Next**.

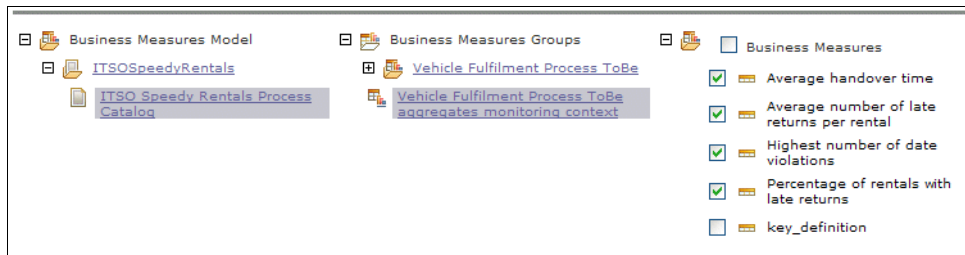


Figure 14-53 Selecting the aggregate measures to display

6. The next page enables you to configure sorting, but this is not relevant when the view is showing aggregate measures, so click **Next**.

7. The next page enables you to configure a filter, but this is also irrelevant when the view is showing aggregate measures, so click **Next**.
8. Click **Finish**. The view should resemble that shown in Figure 14-54.


Average handover time ↑↓	Average number of late returns per rental	Highest number of date violations	Percentage of rentals with late returns
13 m, 2 s	1.25	5	37.5

Figure 14-54 Completed aggregate values view

Note: We configured only one aggregate metric in the business measures editor — the highest number of date violations — so this view is not showing much new information. This view also shows KPIs, although they are typically better represented by the Gauges, KPIs, or scorecard views.

Configuring the alerts view

The next portlet to configure is on the next dashboard page, Current Rentals. The alerts view displays alerts that were generated by the action manager component as a result of configuring it to produce alerts when it detects situation events. Therefore you must have completed the alerts configuration as described in 14.2.7, “Configuring the adaptive action manager” on page 387. The following steps describe how to set up the alerts view:

1. Click the **Current Rentals** page link at the top to switch to that page.
2. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Alerts portlet to open the configuration page.
3. (Optional) Change the colors to be used and click **Next**.
4. Click **Finish**.
5. You will see no alerts initially. To test the view you must cause the situation event to be produced. In our example, this occurs when a rental is late for return three times. When such a rental is encountered, an alert will be produced and the view will resemble Figure 14-55.



<div> <input type="button" value="Mark Read"/> <input type="button" value="Mark Unread"/> <input type="button" value="Remove"/> </div>				
<div> <input type="button" value="14"/> <input type="button" value="4"/> <input type="button" value="3"/> <input type="button" value="1"/> </div> Results 1 to 3 of 3				
<input type="checkbox"/>	Time ↑↓	Subject ↑↓	Alert Source	
<input type="checkbox"/>	Mar 15, 2006 9:04:07 PM	Overdue limit violation		
<input checked="" type="checkbox"/>	Mar 15, 2006 9:04:36 PM	Overdue limit violation		
<input checked="" type="checkbox"/>	Mar 15, 2006 9:08:39 PM	Overdue limit violation		

Figure 14-55 Completed alerts view

Configuring the active instances view

The next portlet to configure is another copy of the Active Instances portlet. This time we configured it for its true purpose: showing active instance data.

1. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Active Instances portlet to open the configuration page.
2. Expand **Business Measures Model** → **ITSOSpeedyRentals** and click **ITSO Speedy Rentals Process Catalog**.
3. Click **Vehicle Fulfilment Process ToBe** in the Business Measures Groups section.
4. Select the business measures that you want to display for each active rental instance. We selected the following measures:
 - City
 - Country
 - Location type
 - Loyalty Club ID
 - Number of times overdue
 - Requested vehicle class
 - Update agreement duration
 - Vehicle collection duration
 - Vehicle return duration
5. Click **Next**.
6. The next page enables you to configure sorting, but we accepted the defaults and clicked **Next**.
7. The next page can be very useful because it enables you to filter the data in the active instances view. We decided not to do this and clicked **Next**.
8. Click **Finish**. The view should resemble Figure 14-56.












Actions	Diagram	City 	Country	Location type	Loyalty Club ID	Number of times overdue	Requested vehicle class	Update agreement duration
		Manchester	England	Airport		1	A	18 s
		Raleigh	USA	Suburbs		0	C	
Vehicle collection duration		Vehicle return duration		Vehicle Fulfilment Process ToBe_Wait on Vehicle Return		Vehicle Fulfilment Process ToBe_Reserve Vehicle		Activities
16 s		5 m, 36 s						
1 m, 6 s		5 m, 55 s						


Figure 14-56 Completed active instances view

Note: We have split the screen capture in Figure 14-56 over two lines to show all of the view.

Configuring the process diagrams view

The next portlet to configure on the page is the process diagrams view. Use the following steps.

Important: This view shows the process diagram using a Scalable Vector Graphic (SVG). Therefore you must install an SVG viewer into your browser before using this view. You can obtain an SVG viewer on the Internet. We used the Adobe SVG viewer available at <http://www.adobe.com/svg/>

1. Click the Configure icon  to the upper right of the ITSOSpeedyRentals Process Diagrams portlet to open the configuration page.
2. Expand **Business Measures Model** → **ITSOSpeedyRentals** and click **ITSOSpeedyRentals Process Catalog**.
3. Select **Vehicle Fulfilment Process ToBe** as shown in Figure 14-57. (Optional) Remove the informational help text by deselecting the option. Click **Next**.

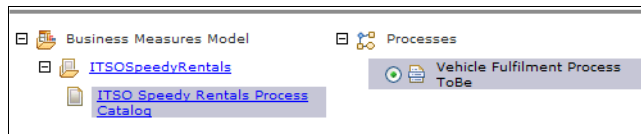


Figure 14-57 Selecting the process for the process diagrams view

4. Click **Finish**. The view should resemble Figure 14-58.

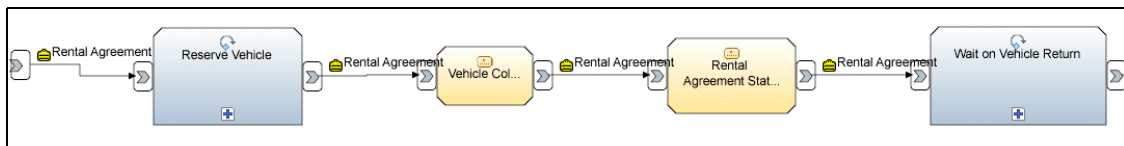


Figure 14-58 Completed process diagrams view


Note: It is possible to interact with the diagram by hovering the mouse over names, or by clicking the expansion sign for subprocesses such as Reserve Vehicle or Wait on Vehicle Return in our example.

Configuring the dimensions view

The next portlet to configure is the dimensions portlet on the next dashboard page, Business Analysis. This is a very powerful view that can be configured in an almost limitless variety of ways to show many different chart types and any data from the model that you chose. The view is initially configurable by an administrator, as we will do now. After the view is configured, a user can also

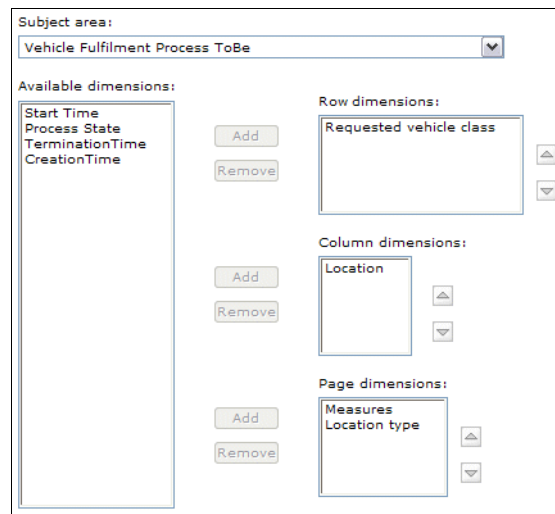
interact with it to perform dynamic analysis. When the user logs out, his or her view is restored to the configuration that was defined by the administrator. Therefore, the administrator's configuration can be thought of as the default or initial view for users. We show many examples of customizing this view further in 15.1.4, "Multi-dimensional business analysis" on page 421.

Use the following steps to configure the view as we did:

1. Click the **Business Analysis** page link at the top to switch to that page.
2. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Dimensions portlet to open the configuration page.
3. Select **Vehicle Fulfilment Process ToBe** from the Subject area list.
4. From the list of available dimensions, select **Measures** then click **Add** to add it to the list of Page dimensions at the bottom. Measures is a special dimension that contains all of the business measures that were defined in the business measures editor as *quantitative data in dimensional analysis*.

Note: Page dimensions are placed at the top of the chart as pull-down lists. Row and Column dimensions can be displayed as either x-axis labels or different-colored bars with a legend about what each color represents.

5. In the same way, add the **Location type** dimension as another Page dimension, the **Location** as a Column dimension, and the **Requested vehicle class** as row dimension so that the completed configuration is as shown in Figure 14-59. Click **Next**.



The screenshot shows a configuration window for the 'Vehicle Fulfilment Process ToBe' subject area. It features four main sections: 'Available dimensions' on the left, and 'Row dimensions', 'Column dimensions', and 'Page dimensions' on the right. The 'Available dimensions' list includes 'Start Time', 'Process State', 'TerminationTime', and 'CreationTime'. The 'Row dimensions' section contains 'Requested vehicle class'. The 'Column dimensions' section contains 'Location'. The 'Page dimensions' section contains 'Measures' and 'Location type'. Each section has 'Add' and 'Remove' buttons, and the rightmost sections have up/down arrow buttons for reordering.

Figure 14-59 Selecting the dimension layout

6. The chart preview appears and all of the dimensions are shown at their top level by default. Right-click the three-dimensional bar and select **Drill Down**.
7. Select **Location: Location** in the dialog and click **OK**.
8. Right-click any orange bar again and select **Drill Down**. This time select **Requested vehicle class: Requested vehicle class** and click **OK**.
9. From the pull-down Measures list at the top of the chart, select the first measurement, **InstancesCount**.
10. Click **Finish**. The chart should resemble Figure 14-60, although it will be highly dependent on the data that has been monitored so far for any process instances that you have run.

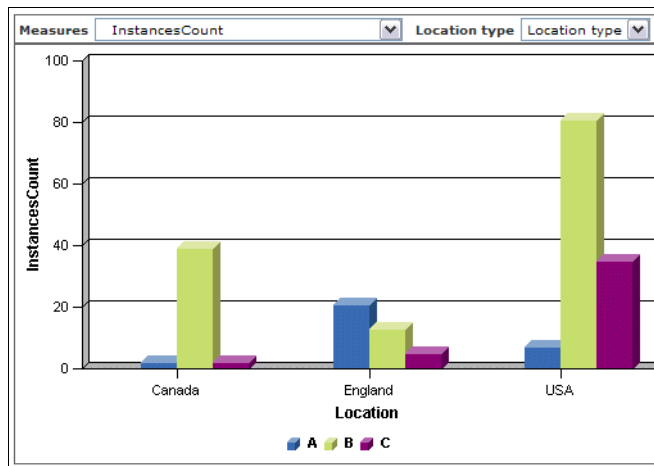



Figure 14-60 Completed dimensions view

Configuring the reports view

The next portlet to configure is the reports portlet, which is on the next dashboard page, Historical Trends. We show how we did the basic configuration of this view. Refer to 15.1.5, “Historical trends” on page 426 for an example of a fully customized look and feel.

Use the following steps to configure the view as we did:

1. Click the **Historical Trends** page link at the top to switch to that page.
2. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Reports portlet to open the configuration page.
3. Select **Vehicle Fulfilment Process ToBe** for Subject area.
4. Select **Trend analysis** from the list of analysis types.

5. Check the **InstancesCount** measure under Modeled measures as shown in Figure 14-61. You can increase the weighting factor to affect the trend analysis because the trend is shown using an exponentially weighted moving average. Click **Next**.

The screenshot shows the 'Analysis Configuration' dialog box. On the left, under 'Subject area', 'Vehicle Fulfilment Process ToBe' is selected. Under 'Dimension', 'None' is selected. Under 'Analysis type', 'Trend analysis' is selected. Under 'Weighing factor', '0.18' is entered. On the right, under 'Measures', the 'Modeled measures' section is expanded, and 'InstancesCount' is checked. Other measures listed include 'Sum of New Items', 'Sum of Resolved Items', 'Group upgrade percentage marker', 'date violation counter', 'Overdue percentage marker', 'Handover watch', 'Elapsed Duration', and 'Working Duration'.

Figure 14-61 Configuring the reports view

6. Click the calendar icons for the From date and the To date to specify a range for the report to cover. Alternatively, click the year to date (YTD) or month to date (MTD) buttons to automatically set the from and to dates. Specify a frequency of daily, monthly, or yearly. We clicked **MTD** and selected **Daily** for the frequency.
7. Click **View Report** to update the report preview with the selected date range and frequency.
8. Click **Finish**. The view should resemble Figure 14-62 on page 413.

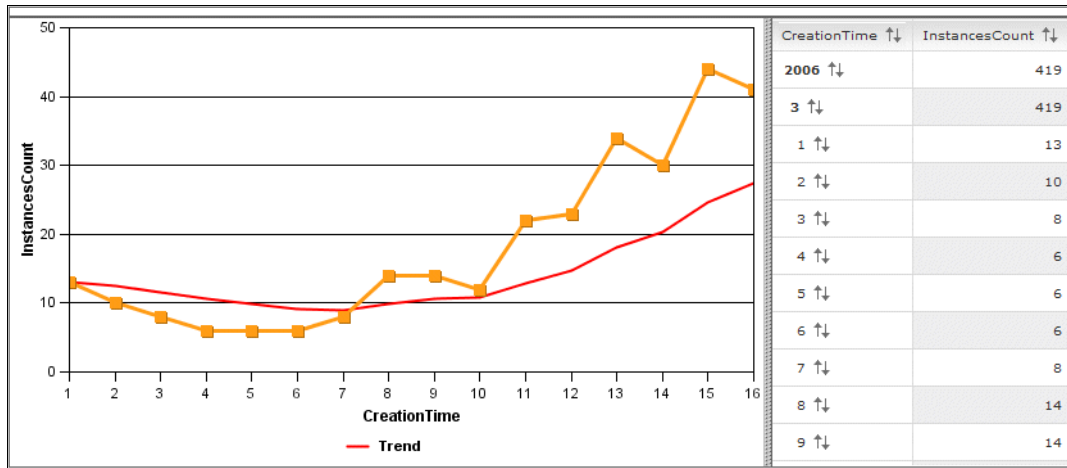



Figure 14-62 Completed reports view

Configuring the organizations view

The next portlet to configure on the page is the organizations view. This is on the next dashboard page, Operations. Use the following steps to configure the view as we did.

Important: To be able to configure this view you must first configure your WebSphere Portal to use an LDAP user registry. We had already done this while configuring the action manager (14.2.7, “Configuring the adaptive action manager” on page 387).

1. Click the Configure icon  to the upper right of the ITSO Speedy Rentals Organizations portlet to open the configuration page.
2. Select the WebSphere Member Manager (WMM) vault slot name that represents the LDAP server that contains your organization information as shown in Figure 14-63 on page 414. Optionally, change the name format and sorting behavior. Click **Finish**.

Select WMM vault slot name:

wmm.system.id.user ▼

Select employee name format:

☐ First name Last name

☐ Last name First name

☒ Last name, First name

Sort employee name by:

☐ First name

☒ Last name

☐ Show informational text

Finish Cancel

Figure 14-63 Selecting the organizational information registry

3. The view should resemble Figure 14-64.

Admins

Big Organization

Bright People

Engineering

Other People

Stupid People

TestUsers

Training

academy

anderson council

barbers

beatles

black sabbath

Search for: Employee Organization:

*Last name: cooper First name: Search

Name ↑↓	E-mail ↑↓	
Cooper, Walter	WZCooper0082@IBM.com	✖
Cooper, Howard	HGCooper0064@IBM.com	⬆
Cooper, Paul	PYCooper0067@IBM.com	⬆
Cooper, Gaylord	GKCooper0094@IBM.com	⬆
Cooper, Freddy	FZCooper0054@IBM.com	⬆
Cooper, Judith	JFCooper0049@IBM.com	✖

Figure 14-64 Completed organizations view

Configuring the export values view

The next portlet is the export values view. This is on the next dashboard page, Control Center.

The ITSO Speedy Rentals Export Values portlet does not need any configuration and can be used immediately. Simply select the date range that you want to export data for, expand the **Business measures model** tree, select the model, and click the **Export Link** as shown in Figure 14-65. Save the file. This can be imported into WebSphere Business Modeler.

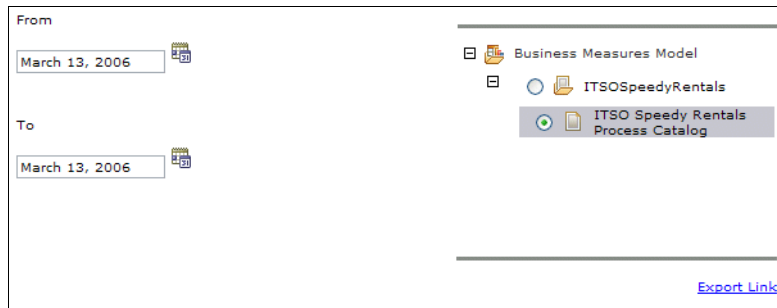


Figure 14-65 Using the export values view



Continuous process improvement

In previous chapters of this book, we showed how to model, assemble, deploy, and manage a business process model. However, the lifecycle does not stop there. Having monitored the process execution, recommendations can be made for process model improvements.

This chapter shows how we analyzed the business process model and added two improvements: additional business measures and an additional task. We modified the existing process to incorporate these improvements by completing the model, assemble, deploy, and manage stages again.

This chapter describes:

- ▶ Monitoring the existing process
- ▶ Improving the process model in WebSphere Business Modeler
- ▶ Assembling the new process model
- ▶ Deploying the new process model to WebSphere Process Server
- ▶ Managing the new business measures model and process model

15.1 Monitoring the existing process

Now that the implementation has been modeled, assembled, and deployed, you can begin to manage that deployment by analyzing its performance using WebSphere Business Monitor. This section shows how we used the dashboards that were created in Chapter 14, “Manage with WebSphere Business Monitor” on page 353. We analyze the results and look for areas of potential improvement.

15.1.1 Executive overview analysis

The Executive Overview page that we created in the dashboard offers a quick and simple way to learn about some key performance indicators for ITSO Speedy Rentals. A glance at the gauges on this page reveals that the number of late rentals is beyond our defined limits. Hover the mouse over the gauge needles to see the exact values, as shown in Figure 15-1.

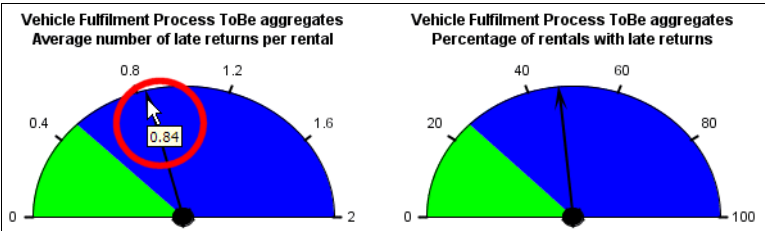


Figure 15-1 At-a-glance evidence of late rentals

The KPIs view on this page provides similar information and displays the actual values in a tabular format for quick viewing. The scorecards view shows that the efficiency of the ITSO Speedy Rentals service team was good, with average vehicle handover time running within the target range, as indicated by the green status in Figure 15-2.

Efficiency						
KPI Name	↑↓	Status ↑↓	Value	Target	Score	↑↓
Vehicle Fulfilment Process ToBe aggregates.Average handover time		🟢	10 m, 27 s	12 m	<div></div>	87%

Figure 15-2 Efficiency scorecard showing on-target performance for vehicle handovers

15.1.2 Aggregated analysis

We defined only one aggregate metric in our business measures model: the highest number of late rentals for any rental contract. The Aggregated Analysis page of the dashboard shows that at least one rental had been late five times, as shown in Figure 15-3 on page 419.

Results 1 to 1 of 1			
Average handover time	Average number of late returns per rental	Highest number of date violations	Percentage of rentals with late returns
10 m, 27 s	0.4	5	47

Figure 15-3 Aggregated analysis shows the highest number of date violations for a single contract

This measure provides a simple way to determine whether there is an issue with any renters repeatedly being late with their return.

15.1.3 Current rentals analysis

The dashboard enables you to analyze active rentals in the system by looking at the Current Rentals page. The alerts, active instances, and process diagram views are all configured to work together through cooperative views. The active instances view shows all active rental contracts, and the alerts view produces alerts for any rental contract that is late for return three or more times.

The active instances view in Figure 15-4 shows two active rental contracts.

Executive Overview
Aggregated Analysis
Current Rentals
Business Analysis
Historical Trends
Operations

ITSO Speedy Rentals Alerts

Mark Read
Mark Unread
Remove

Results 1 to 1 of 1

	Time	Subject	Alert Source
	Mar 16, 2006 6:55:55 PM	<u>Overdue limit violation</u>	

ITSO Speedy Rentals Active Instances

Vehicle Fulfilment Process ToBe

Results 1 to 2 of 2

Actions	Diagram	City	Country	Location type	Loyalty Club ID	Number of times overdue
		Manchester	England	Suburbs		3
		Toronto	Canada	Airport		0

Figure 15-4 Current rentals and alerts

Tip: The process diagrams portlet is very wide. You can minimize it by clicking the minimize icon in the corner of the portlet. To look at it again, click the restore icon. All of the portlets can be minimized, maximized, and restored, just like regular application windows.

Begin by investigating the rental that was not overdue. Click the Diagram link for this instance to update the process diagrams view and show the progress of that instance, as seen in Figure 15-5.

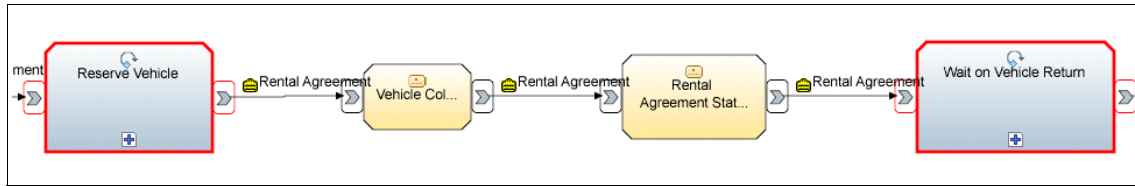


Figure 15-5 Current instance progress highlighted in the process diagrams view

One of the rentals was returned late three times, so an alert was generated. Click the alert source icon for this alert to update the process diagram again and show the specific instance the active instances view for which the alert was generated. This is useful for tracking down which instance was late too many times.

You can also click on the alert message subject to read the alert and to determine the rental contract number in the message body (Figure 15-6).

Figure 15-6 Reading the alert message to find out which rental is late

Click the alert source icon, and the active instances view shows all data for that process. Scroll across and click the activities drill-down icon to investigate the subprocess task Wait on vehicle return. This shows the previous two late returns and that another subprocess is active. Drill down again on this running

subprocess to show that since the latest date violation the rental agreement has not been updated with a new return date (Figure 15-7).

Activity State	Activity Start Time	Activity Assigned User	Activity Assigned User List	Business Name
Completed	Mar 16, 2006 6:50:32 PM			Check If Vehicle Returned
Running	Mar 16, 2006 6:51:07 PM	UNAUTHENTICATED		Update Rental Agreement

Figure 15-7 Rental instance drill down reveals that rental agreement requires updating

15.1.4 Multi-dimensional business analysis

The dimensions view that we configured for business analysis has an almost infinite variety of ways to show different chart types, data choices, drill-down levels, and dimension combinations, as well as look-and-feel factors such as colors, labels, and fonts. We created a basic initial configuration for the view in Chapter 14, “Manage with WebSphere Business Monitor” on page 353, and now we use this view as a starting point for an analysis of the ITSO Speedy Rentals business process.

General analysis

The view initially shows the number of rentals by the first level of location and by the requested car class. The simplest thing that a user of this dashboard can do is pick a measure from the pull-down list of business measures that we had defined, as shown in Figure 15-8.

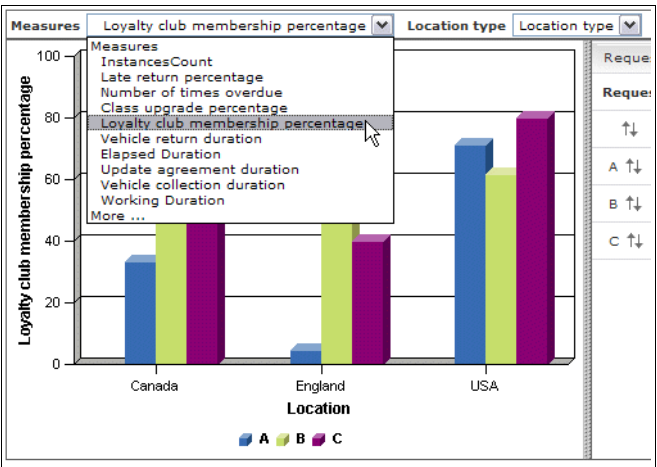


Figure 15-8 Selecting which measure to analyze

You can also filter the data for a location type using the other pull-down list. Figure 15-8 analyzes the percentage of rentals that were made with a loyalty club

ID. The data reveals that USA-based rentals show greater use of the ITSO Speedy Rentals loyalty program, with steady distribution across each class of rental. In contrast, rentals in England show lower use of the program, particularly among the cheaper class-A rental users. This information is particularly useful to the ITSO Speedy Rentals marketing team, which might decide to improve use of the program in England.

Select Airport for the location type. For airport rentals, Canada was actually significantly higher than the United States in the use of loyalty ID (Figure 15-9).

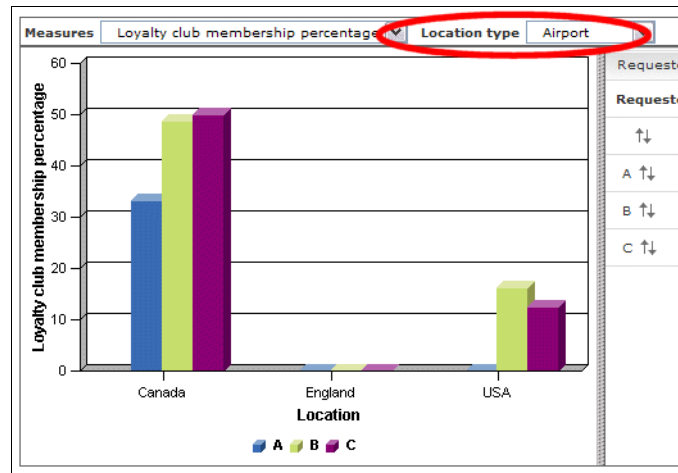


Figure 15-9 Comparing loyalty card use at airports

You can investigate the time taken for the manual steps in the process, such as vehicle collection, which is typically a key contributor to customer satisfaction. By selecting the Vehicle collection duration and drilling down on the USA location, you can analyze collection performance by vehicle class and by city in the USA.

The easiest way to analyze the number of rentals by vehicle class is to switch to a pie chart type as shown in Figure 15-10 on page 423. This also shows the ability to filter the data using a pull-down list for the location.

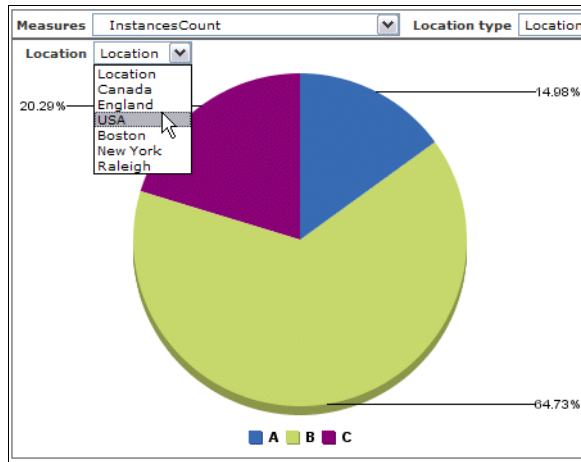


Figure 15-10 Analyzing the percentage of rentals by vehicle class and dimension

The pie chart view enables you to look at each location distribution, one at a time. However, the radar area chart shown in Figure 15-11 is even more effective, combining information for each location on the same chart at the same time.

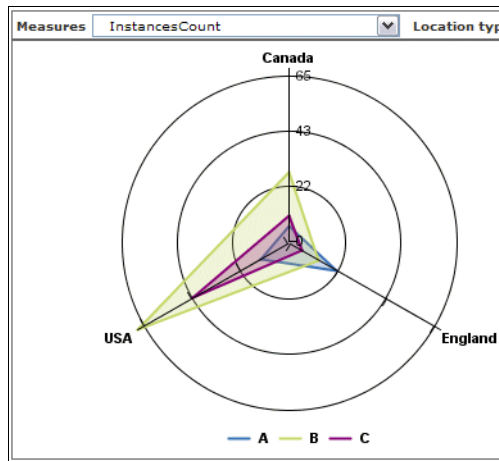


Figure 15-11 Distribution of rentals showing vehicle class and country

Upgrades analysis

The original analysis showed that upgrading vehicle class because of unavailability of the requested class has a significant cost for ITS0 Speedy Rentals. Ideally, this should happen less than 15% of the time. You can use the dashboard to compare reality with this goal across a variety of locations, location types, and rental classes.

First, look at charts showing the upgrades percentages for each vehicle class at different location types. Figure 15-12 shows two of these charts side by side.

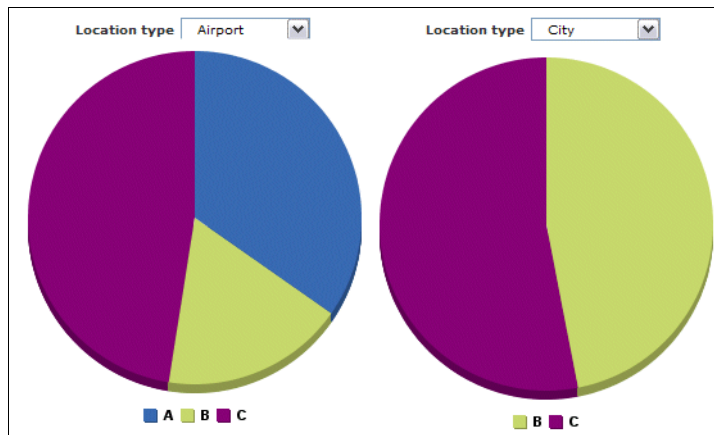


Figure 15-12 Upgrade distribution by vehicle class for City and Airport rentals

The diagram clearly shows that renters at the airport are most likely to need an upgrade due to unavailability of class C cars, with the same being true in the city. The city locations do not suffer from needing A class upgrades.

Another, perhaps more useful analysis, is to understand the percentage of rental contracts that need an upgrade. (The pie chart analysis in Figure 15-13 reveals only the breakdown of which class vehicles are upgraded the most, not how often upgrades occur among rentals.) We configured the dimensions view to show this information as a horizontal bar in Figure 15-13.

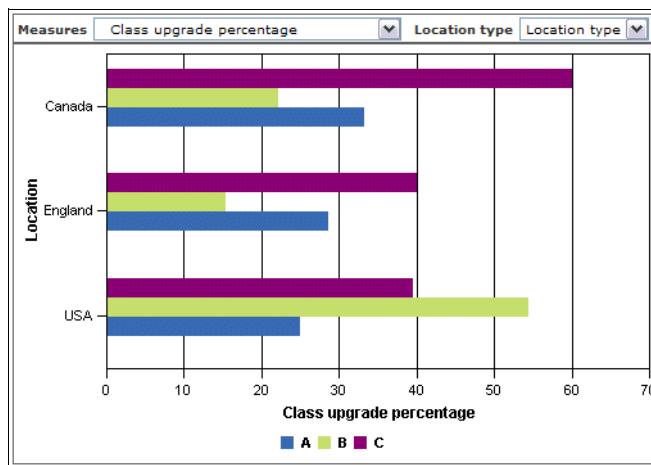


Figure 15-13 Percentage of rentals requiring upgrade by location and country

This analysis shows that the number of rentals that are requiring upgrades is too high — often well above the 15% that was considered a goal when the process was designed. Because this measurement was so critical, and was so far off the target, we decided to focus ITSO Speedy Rentals on improving these figures. Based on analysis of the WebSphere Business Monitor chart, the implementation was changed to adjust the number of vehicles in each class at each location, and new key performance indicators were added to the business measures model to show current performance compared to this new target. These steps are described in the other sections of this chapter, beginning with 15.2, “Improving the process model in WebSphere Business Modeler” on page 427.

Late returns analysis

Next, analyze another crucial cost factor: how many late returns ITSO Speedy Rentals encountered.

Note: We did not directly model costs in our business measures model, but this could be done as a natural extension by adding cost data to the business measures model calculations. We relied on our business analysis in the modeling phase for indications as to where costs may lie.

Using a 3-D bar chart to compare the late returns percentages across all of the cities that ITSO Speedy Rentals operates in, this enquiry shows that although no specific patterns were obvious, as a general point far too many rentals were being returned late, as shown in Figure 15-14.

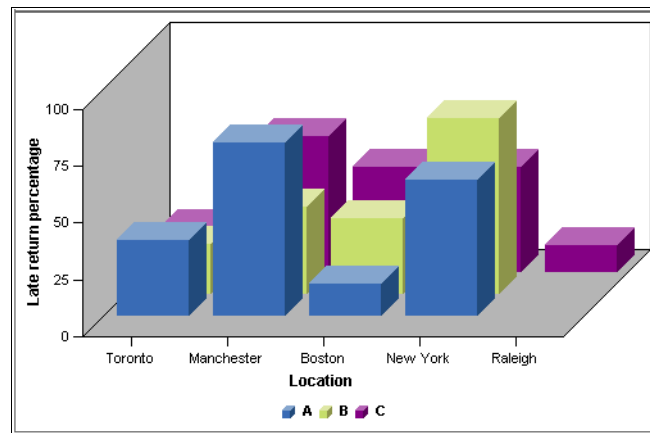


Figure 15-14 Late return percentages by city and vehicle class requested

Therefore we decided to implement a change to the business process to perform an additional step of contacting the customer after a late return to confirm the

new rental agreement, in an effort to try to reduce the number of repeat late returns. Upcoming sections of this chapter, beginning with 15.2, “Improving the process model in WebSphere Business Modeler” on page 427, explain the steps that were taken to change, reimplement, and redeploy the model.

15.1.5 Historical trends

Next, look at the historical trend information to understand how many rentals are occurring over time in an effort to understand the current business performance and to forecast future growth. We further customized the appearance of the trend chart that we created in “Configuring the reports view” on page 411 to produce the chart shown in Figure 15-15.

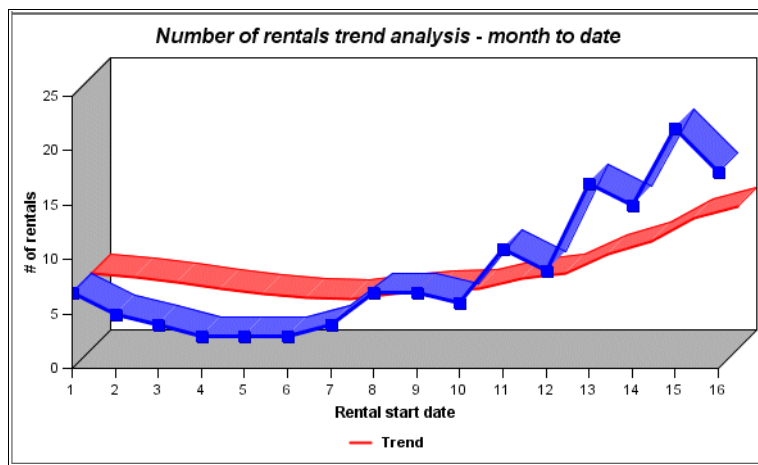


Figure 15-15 Number of rentals trend over the month to date

Note: All of the customizations to the charts, changes to the graph types, color customization, title, label changes, and more can be performed by right-clicking on the various views and looking at the available options.

The trend shows that ITSO Speedy Rentals can hope for a bright future if the recent growth is sustained.

15.1.6 Exporting values

On the dashboard Control Center page, use the export values portlet to save data from the past month of rentals. A simple import brings the data in this XML file into WebSphere Business Modeler to show real branching percentages and



durations. This enabled us to perform more accurate simulations and analysis based on real observations captured by WebSphere Business Monitor.

15.2 Improving the process model in WebSphere Business Modeler

This section shows how we modified the model built in Chapter 11, “Model with WebSphere Business Modeler” on page 167 to include a new activity and to add an additional business measure. We then export the new model.

15.2.1 Adding the Confirm Updated Agreement activity

You can update the model to include a new automated activity. that automatically e-mails the updated rental agreement to a customer.

1. Open the model in WebSphere Business Modeler.
 - a. Double-click **Vehicle Fulfilment Process ToBe** to open the process editor.
 - b. Select the 1-pane layout icon .
 - c. Expand **While Loop: Wait on Vehicle Return**, and add a new local task selected from the process editor palette into this while loop.
 - d. Enter Confirm Updated Agreement as the name of the new task.
2. Connect the new activity.
 - a. Connect the activities using the  button, inserting the task between **Update Rental Agreement** and **Merge**. You can move the connection from Update Rental Agreement to Merge to the new activity, then make a new connection from **Confirm Updated Agreement** to **Merge**. Make sure that both connections have Rental Agreement as associated data.

The new activity should be inserted as shown in Figure 15-16.

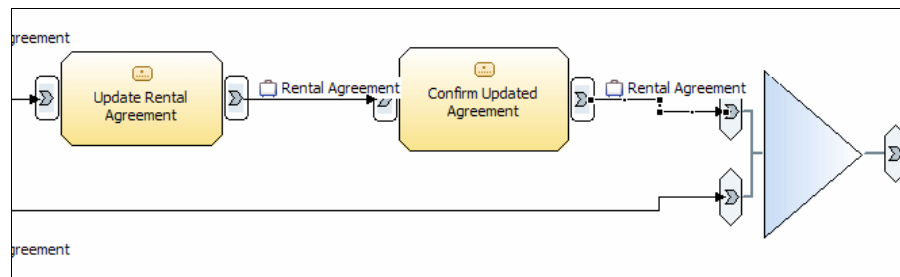



Figure 15-16 Addition of the Confirm Updated Agreement task

- b. Open the 4-pane view by clicking the  icon. This opens the project tree.
 - c. Select the connection between **Update Rental Agreement** and **Confirm Updated Agreement**. In the Attributes view, ensure that the name of the connection is Updated Rental Agreement Connection.
 - d. Select the connection between **Confirm Updated Agreement** and **Merge**. In the Attributes view, set the name of the connection to Confirm Updated Agreement Connection.
3. Name the inputs and outputs of Confirm Updated Agreement.
 - a. Click the **Confirm Updated Agreement** task.
 - b. In the Attributes view, click the **Inputs** tab. Rename the existing input to Confirm Updated Agreement Input.
 - c. In the Attributes view, click the **Outputs** tab. Rename the existing output to Confirm Updated Agreement Output.
4. Add a classifier to mark this automated activity as an IT System task.
 - a. In the Attributes view, click the **Classifier** tab.
 - b. Click **Add** to add a new classifier. In the Select Classifier Value dialog, select **IT System** and click **OK**.
5. Add IT Rental System Bulk Resource as the resource for this automated task.
 - a. In the Attributes view, click the **Resources** tab.
 - b. In the Bulk resource requirements section, click **Add**.
 - c. Select the new bulk resource and click ... in the Bulk resource column to open the Select Bulk Resource or Resource Definition dialog. Click **Rental IT System Resource** and click **OK**.
 - d. Click ... in the Time required column to open the Select duration dialog. Set the duration to **2** milliseconds and click **OK**.
6. Set the implementation type technical attribute to Java to indicate that this task will be implemented by Java code.
 - a. In Process Server mode, open the Technical Attributes view using **Window → Show View → Other → Business modeler views → Technical Attributes view**.
 - a. Select the **Implementation** tab.
 - b. Set the Implementation type to **Java**.
7. Save the process.

15.2.2 Adding new business measures

Now add functionality to measure the percentage of vehicle class upgrades for each vehicle class (A, B, and C). An acceptable vehicle class upgrade percentage for class A is 25%, and 15% for class B and C.

Synchronize the business measures

The additional task that was added to the process in the previous section must be synchronized with the business measures model:

1. In the project tree, right-click **Vehicle Fulfilment Process ToBe** and select **Synchronize**.
2. In the Synchronize business measures dialog, the Confirm Updated Agreement task has been added as an additional process element to measure (Figure 15-17). Click **OK**.

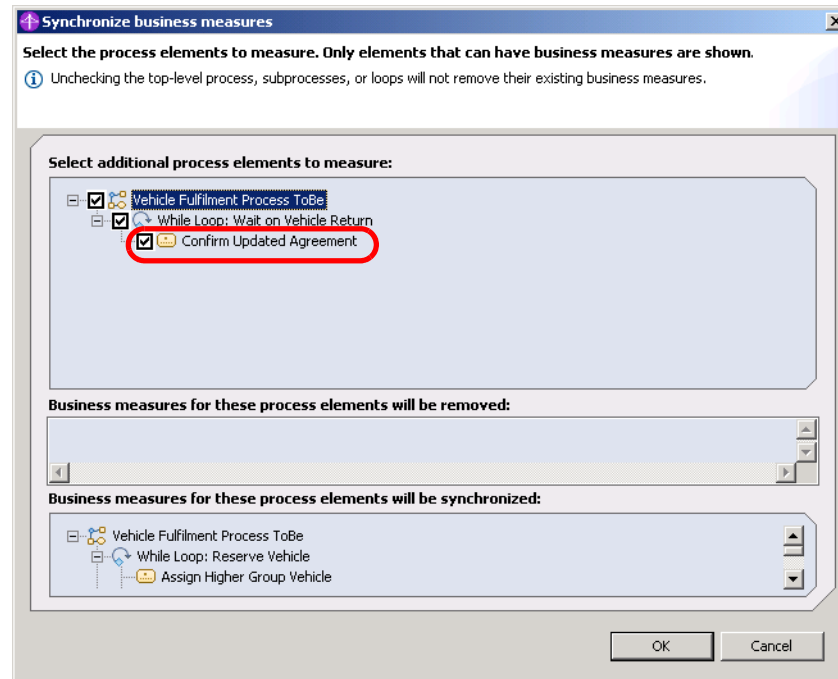


Figure 15-17 Synchronize business measures

3. In the project tree, double-click **Vehicle Fulfilment Process ToBe Business measures**. Click the **Diagram** tab.

4. In the business measures diagram view, expand the While Loop: Wait on Vehicle return task. You should see the Confirm Updated Agreement tasks is now visible in the business measures diagram.
5. Right-click in the white space in the diagram and click **Root** to the return to the top level process view.

Create a metric

Define the Percentage of upgrades for class A rentals metric:

1. In the Attributes view for the process, click the **Metrics** tab.
2. Click **Add** in the Metrics section. Name the metric Percentage of upgrades for class A rentals, and change the Type to **Double**.
3. Select the **Percentage of upgrades for class A rentals** metric.
4. Scroll to the Value section and click **Add**. Click in the Trigger column and click ... to open the Select Trigger dialog.
5. In the Select Trigger dialog:
 - a. Click **Add new trigger**.
 - b. Set Source category to **Metric**.
 - c. Click ... next to Source. In the Select Metric dialog, click **Class upgrade percentage** and click **OK**.
 - d. Click ... next to On condition. In the Expression Builder (Figure 15-18), and follow these steps:
 - i. Set First term to **Modeling artifact**.
 - ii. In First term details, expand **Vehicle Fulfilment Process ToBe** and select **Requested vehicle class**.
 - iii. Set Operator to **is equal to**.
 - iv. Set Second term to **Text**.
 - v. Enter a Text value of A.
 - vi. Click **Apply** then **OK**.

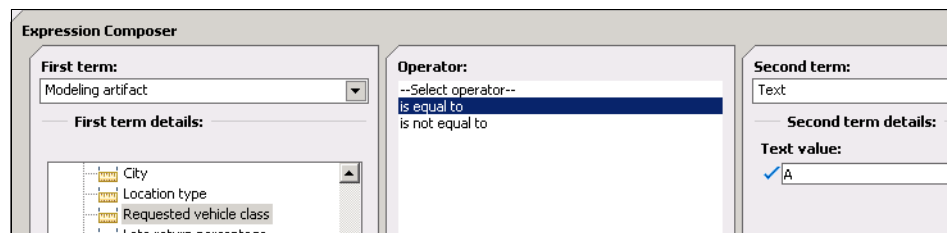


Figure 15-18 Expression Builder

e. The Select Trigger dialog should now match Figure 15-19. Click **OK**.

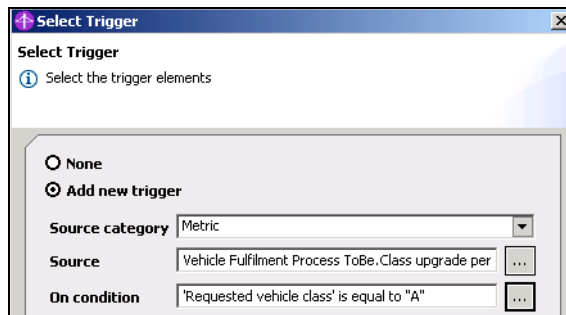


Figure 15-19 Select Trigger dialog

6. Click ... in the Calculation field for the trigger to open the Expression Builder. In the Expression Builder dialog:
 - a. Set First term to **Number**.
 - b. Enter a Number value of 100.
 - c. Click **Apply** then **OK**.
7. Scroll to the Dimensional analysis and database schema settings section and follow these steps:
 - a. Set Usage in WebSphere Business Monitor to **Quantitative data in dimensional analysis**.
 - b. Click **Add** to add a new aggregation measure.
 - c. Set the Aggregation measure name to Average percentage of upgrades for class A rentals.
 - d. Set Aggregation function to **Average**.
8. Save the business measures model.

Creating additional metrics

Create two additional metrics:

- ▶ Percentage of upgrades for class B rentals
- ▶ Percentage of upgrades for class C rentals

Follow the instructions for the previous metric to create these two additional metrics:

1. Change the metric name, the trigger, and the aggregation measure name to reflect the correct vehicle class for each metric.
2. Save the business measures model.

Create KPIs

Create the following KPIs:

1. In the main editor, click the **KPIs and Aggregate Metrics** tab.
2. Click the **KPIs** link on the left side of the editor, then click **Add** in the KPIs section to add a new KPI.
3. Set the following properties for the KPI:
 - a. Set the Name field to Average percentage of upgrades for class A rentals.
 - b. Set the Type to **Double**.
 - c. Set the Aggregation function to **Average**.
 - d. Set the Aggregation source to **Percentage of upgrades for class A rentals**.
 - e. Uncheck **Use Target**.
 - f. Set the Lower limit to 0.
 - g. Set the Upper Limit to 25.
4. Repeat this process to define two additional KPIs, using the values in Table 15-1.

Table 15-1 Additional KPI properties

KPI	Type	Aggregation function	Aggregation source	Use Target	Upper limit	Lower limit
Average percentage of upgrades for class B rentals	Double	Average	Percentage of upgrades for class B rentals	No	15	0
Average percentage of upgrades for class A rentals	Double	Average	Percentage of upgrades for class C rentals	No	15	0

5. Save the changes.

15.2.3 Export the project

Now export the model so it can be imported into WebSphere Integration Developer, and export the business measures model so it can be imported into WebSphere Business Monitor.

Follow these steps:

1. Right-click anywhere in the project tree and select **Export**.

2. In the WebSphere Business Modeler Export dialog, select **WebSphere Business Monitor and development tool** and click **Next**.
3. In the Destination and source window:
 - a. Click **Browse** to specify the directory where you want to put the files.
 - b. Select **ITSO Speedy Rentals To-Be Project** from the Project list.
 - c. Set Monitor project version to **2**.
 - d. Check **Export user defined event types to event catalog**.
 - e. Click **Overwrite files**.
 - f. Click **Next**.
4. In the WebSphere Process Server export details window:
 - a. Specify the WebSphere Integration Developer module by checking **Module project name** and entering VehicleRentV2 as name of the module.
 - b. Specify the WebSphere Integration Developer library by checking **Library project name** and entering VehicleRentLibV2 as name of the library.
 - c. Specify the Project interface name by checking **Project interchange name** and enter VehicleRentV2.
 - d. Click **Finish**.
5. A window opens when the export process is complete. Click **OK** to acknowledge the completion of the export.

15.3 Assembling the new process model

To implement the changes to the vehicle rental process, import the process artifacts that were generated from WebSphere Business Modeler, and customize the new process and implement a service component for the new Confirm Updated Agreement process task. Then reassemble the process module and services module, and redeploy and unit-test the new process.

15.3.1 Importing new process artifacts

This section imports the new process artifacts and adjusts the modules and library:

1. In WebSphere Integration Developer, navigate to the Servers tab on the bottom of the screen, right-click **WebSphere Process Server v6.0**, and choose **Add and Remove Projects**.

2. On the right, select **VehicleRentApp** and **VehRentSvcsApp**, then click **Remove**. Click **Finish**. This undeploys these modules from the unit-test environment; later you will redeploy the new versions that you develop here.
3. Click **File** → **Import** → **Process Interchange** and click **Next**. Choose to import the WebSphere Process Server\VehicleRentV2.zip project interchange file that you exported in 15.2.3, “Export the project” on page 432. Alternatively, you could use the VehicleRentV2.zip file supplied with the additional material accompanying this book, located in the Lifecycle\ModelerExport\WebSphere Process Server directory.
4. Click **Select All** to select both the VehicleRentV2 module and the VehicleRentLibV2 library, and click **Finish**.
5. Click **Project** → **Clean**, select **Clean all projects**, and click **OK**. This rebuilds the artifacts in the workspace. You should have only two errors in the problems view, both related to unimplemented operations in the business rules in the V2 process. Ignore these for now.
6. In the Business Integration view, expand **VehicleRentV2** → **Business Logic** → **Processes**. Right-click the **CheckIfVehicleReturned** subprocess and click **Delete**.

Note: This is service component reuse in action: because the subprocess did not change in the new main process design, you can continue to use the previously created CheckIfVehicleReturned subprocess that is already a part of the VehRentSvcs module. Notice how well SCA helps to isolate change. Changes in one module do not affect changes in another.

7. Delete all components in the VehicleRentV2 module under Human Tasks and Rule Groups. You already created these service components, therefore you only need to import them again into our assembly diagram for VehicleRentV2 and connect them to the new process. We will cover this shortly.
8. In the VehicleRentV2 module, expand **Interfaces**, and select the **ConfirmUpdatedAgreement** interface. This corresponds to the new task that has been added to this version of the process. Right-click the interface and select **Move**. In the Module Selection dialog, click **VehicleRentLib** and click **OK**. This makes the new interface available in your shared library, which is used across multiple modules.
9. Select the remainder of the interfaces under VehicleRentV2 and click **Delete**. This creates a lot of errors in the problems view, but the next steps fix them.
10. Double-click the **VehicleRentV2** module to open the Dependencies Editor. Perform the following steps:
 - a. Select the **VehicleRentLibV2** library and click **Remove**.

- b. Click **Add**, select **VehicleRentLib**, and click **OK**.
 - c. Save and close the Dependencies Editor.
11. Click **Project** → **Clean**, select **Clean all projects**, and click **OK** to rebuild the workspace. After the workspace has been rebuilt, most of the errors will disappear from the problems tab.
 12. Right-click the **VehicleRentLibV2** library and click **Delete**, choosing to also delete the file system contents, and click **Yes**. You do not need this library because you already have the VehicleRentLib library.

15.3.2 Customizing the new process

This section fixes the broken references that have occurred and implements the same process logic customizations that were made to the previous version of the process.

1. Expand **VehicleRentV2** → **Business Logic** → **Processes**, and double-click **VehicleFulfilmentProcessToBe** to open it in the Business Process Editor.
2. Fix the broken reference to the VehicleFulfilmentProcessToBe interface partner:
 - a. Click the interface partner **VehicleFulfilmentProcessToBe**.
 - b. In the Properties tab, click **Details**.
 - c. To set the interface, click **Browse** and in the Interface Selection dialog click **VehicleFulfilmentProcessToBe**. Click **OK**.
 - d. Save the changes.
3. In the process diagram, in While Loop: Reserve Vehicle, highlight **Read From Local Information Repository**. In the properties view, click the **Details** tab. Comment out the last line of the Java condition:

```
//RentalAgreementVariable2=(commonj.sdo.DataObject)ListOf1ToUnboundedRentalAgreementsVariable.getList("rentalagreementItem").get(0);
```

Add this line:

```
RentalAgreementVariable2 = RentalAgreementVariable;
```

4. Perform a similar customization by selecting, from in While Loop: Wait on Vehicle Return, the **Read From Local Information Repository** element. In the properties view, click the **Details** tab. Comment out the last line of the Java condition:

```
//RentalAgreementVariable3=(commonj.sdo.DataObject)ListOf1ToUnboundedRentalAgreementsVariable2.getList("rentalagreementItem").get(0);
```

Add this line:

```
RentalAgreementVariable3 = RentalAgreementVariable;
```

5. Save the process and close the Business Process Editor.

15.3.3 Implementing the new process task

This section implements the new Confirm Updated Agreement task that is being added to the process by creating a new component in the VehRentSvcs module and implementing it in Java.

Note: This is a quick and easy implementation to show in our scenario how to implement new components and integrate them with new versions of a business process. A real-world implementation would actually implement the required functionality of sending a confirmation e-mail to the customer, and could be implemented in Java, a Web service, or an adapter service.

1. Expand the **VehicleRentSvcs** module, and double-click the **VehicleRentSvcs** assembly diagram.
2. Expand the **VehicleRentLib** library, then **Interfaces**, and drag the **ConfirmUpdatedAgreement** interface onto the VehicleRentSvcs assembly diagram. In the Component Creation dialog, select **Component with no Implementation Type** and click **OK**.
3. Rename the new component to ConfirmUpdatedAgreement.
4. Right-click **ConfirmUpdatedAgreement** and click **Generate Implementation** → **Java**. In the Generate Implementation dialog, click **vehrentsvcs** for the package and click **OK**.
5. The Java implementation file opens in a Java editor. Locate the InputCriterion method, and delete the following lines:

```
//TODO Needs to be implemented.  
return null;
```

Replace them with the following lines inside the InputCrtierion method:

```
String updatedReturnDate =  
(confirmUpdatedAgreementInput.getDate("PlannedReturnDateTime")).toSt  
ring();  
System.out.println("Rental Agreement has been updated with new  
planned return date of " + updatedReturnDate);  
return confirmUpdatedAgreementInput;
```

6. Save the changes to the Java file, and close the Java editor.

7. Right-click **ConfirmUpdatedAgreement** and choose **Export** → **SCA Binding**. This creates an SCA export connected to ConfirmUpdatedAgreement on the assembly diagram. The SCA export will allow this new service component to be used by the new process.
8. Save your changes to the VehRentSvcs assembly diagram and close the editor.

15.3.4 Assembling the new modules

This section assembles the new process with the new service component that has been created to perform the Confirm Updated Agreement process task.

1. Expand the **VehicleRentV2** module, then double-click the **VehicleRentV2** assembly diagram to open it.
2. With the mouse, draw a selection box around all of the components that are connected to the right side of the process (everything to the right of Vehicle Fulfilment Process ToBe), and click Delete (Figure 15-20).

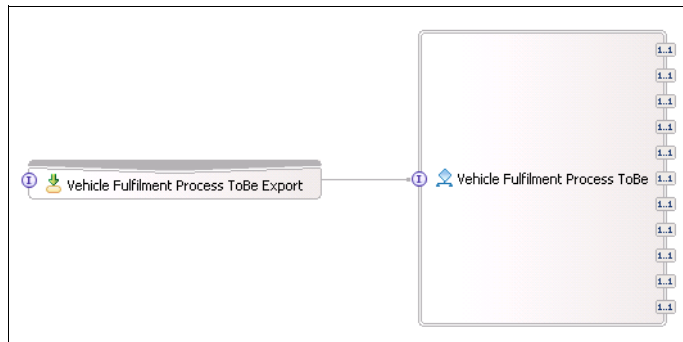


Figure 15-20 Modified assembly diagram

3. In the Business Integration view, right-click the white space and select **Show Files**. This opens the Physical Resources view, which shows the files underlying the visual elements in the Business Integration view.
4. In the Physical Resources view, expand the **VehicleRent** module. Multiple-select all 10 files that have the .import extension, and click **Edit** → **Copy** (Figure 15-21 on page 438).

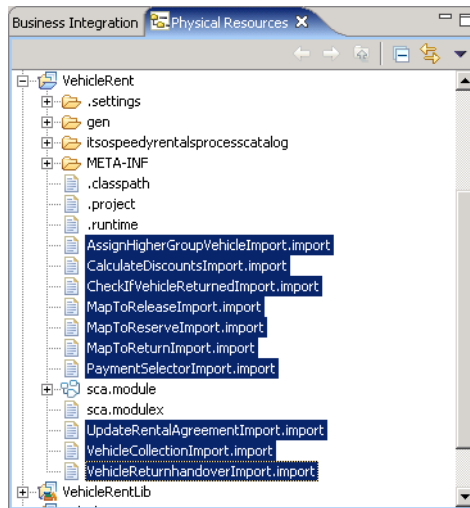


Figure 15-21 Physical resources view

5. Right-click **VehicleRentV2** and click **Paste**. All of the imports appear, jumbled, on your open VehicleRentV2 assembly diagram. To clean this up, right-click in the assembly diagram, deselect **Arrange Contents Automatically**, then select it again. The contents will be rearranged on the diagram.

Note: We have copied all of the imports you used in the previous version of the VehicleRent module. You only need to add the new component import and reconnect the process component to the service components to complete the assembly diagram.

6. In the Business Integration view, expand **VehRentSvc**, then *expand* (not double-click) the **VehRentSvc** assembly diagram. Select the **ConfirmUpdatedAgreementExport** component, and drag it over to the VehicleRentV2 assembly diagram. In the Component Creation dialog, select **Import with SCA Binding**. Rename the Import1 element to **ConfirmUpdatedAgreementImport**.
7. Now you can wire all of the references in the Process component to the Imports you have just imported. Right-click the **Vehicle Fulfilment Process ToBe** component and select **Wire (Advanced)** to display the Advanced Wiring dialog. Use the settings in Table 15-2 on page 439 to wire each source to a target. When all of the sources are wired to targets, click **OK**.

Table 15-2 Advanced wiring

Wire source	Wire target
CalculateDiscountsPartner	CalculateDiscountsImport
Payment Partner	PaymentSelectorImport
RentalAgreementStatusSetToActivePartner	MapToReleaseImport
ReturnVehicleToSystemPartner	MapToReturnImport
VehicleCollectionPartner	VehicleCollectionImport
VehicleReturnhandoverPartner	VehicleReturnhandoverImport
WhileLoopReserveVehicleAssignHigherGroupVehiclePartner	AssignHigherGroupVehicleImport
WhileLoopReserveVehicleReserveVehicleInSystemPartner	MapToReserveImport
WhileLoopWaitonVehicleReturnCheckIfVehicleReturnedPartner	CheckIfVehicleReturnedImport
WhileLoopWaitonVehicleReturnConfirmUpdatedAgreementPartner	ConfirmUpdatedAgreementImport
WhileLoopWaitonVehicleReturnUpdateRentalAgreementPartner	UpdateRentalAgreementImport

8. Save the assembly diagram, and close the editor. You should see no errors in the Problems view, although there will be some warnings, which can be safely ignored.

The new process component has now been connected to the old service components and the new service component.

15.3.5 Deploying and testing the new components

Now you can deploy the new process module and services module to the unit-test environment, where you can verify the function of these new modules.

1. Click the **Servers** tab, right-click **WebSphere Process Server v6**, and click **Add and remove projects**.
2. Select **VehRentSvcApp** and click **Add**. Also, select **VehicleRentV2App** and click **Add**. Click **Finish** and wait a bit while these modules are deployed.

3. In the VehicleRentV2 assembly diagram, right-click the **Vehicle Fulfilment Process To Be** component and choose **Test Component** to open the component test editor.
4. Click the **Configurations** tab, multiple-select all of the emulators, and click **Remove** because you do not need them in this component test.
5. Click back to the **Events** tab. Provide the values shown in Table 15-3 to start an instance of the process and click **Continue**. Click **Finish** to deploy the component test if prompted.

Table 15-3 Sample input values to start a process instance

Filed	Value	Constraint
VehicleGroupRequested	B	A, B or C. The three types of vehicles in the Reserve database.
RiskLevel	low	low, middle or high. Used in the business rule when an upgrade is needed.
InventoryID	null (using pull-down)	Actual InventoryID will be returned by PoolManager Web Service.

6. The component test proceeds through the vehicle reservation step, and stops at the Vehicle Collection human task. To complete this task:
 - a. In the **Servers** view, right-click **WebSphere Process Server v6.0** and click **Launch** → **BPC Explorer**.
 - b. Log in to the console with agent2 as both the user ID and password.
 - c. You should see a Vehicle Collection task awaiting your action. Click the check box beside the task and click **Work on**.
 - d. Provide the values in Table 15-4 for this human task and click **Complete**.

Table 15-4 Input values to complete VehicleCollection task

Field	How to locate the value to use
ReservationID	Locate the ReservationID value in the input message VehicleCollectionInput. Use this same value in the output message.
PlannedReturnDateTime	This value should be a date/time that is <i>earlier</i> than the current date/time.

Field	How to locate the value to use
InventoryID	Locate the InventoryID value in the input message VehicleCollectionInput. Use this same value in the output message.

- e. Log out of the BPC Explorer but keep the window open.
7. The component test proceeds through the vehicle release, and a timeout event occurs from entering the value for PlannedReturnDateTime that is before the current date/time. This simulates a rental timeout business event:
 - a. Log back into the BPC Explorer, this time with agent1 for user ID and password.
 - b. Click the check box beside UpdateRentalAgreement and click **Work On**.
 - c. Provide the values in Table 15-5 for this human task and click **Complete**.

Table 15-5 Input values to complete UpdateRentalAgreement task

Field	How to locate the value to use
ReservationID	Locate the ReservationID value in the input message UpdateRentalAgreementInput. Use this same value in the output message.
PlannedReturnDateTime	This value should be a date/time that is <i>later</i> than the current date/time. This represents the branch agent updating the rental agreement with the new planned return date after contacting the customer.
VehicleStatus	Set to out.
InventoryID	Locate the InventoryID value in the input message UpdateRentalAgreementInput. Use this same value in the output message.

8. The component test continues, and the ConfirmUpdatedAgreement task that you added runs. Check the console view for a message similar to this:

Rental Agreement has been updated with new planned return date of Sat May 06 07:58:00 EDT 2006
9. The component test is successful for the new process and new service component. In the component test window, click the blue Stop icon to end the component test process instance.
10. Stop the server.

You have now successfully implemented the changed process and the new Confirm Updated Agreement process task component. By implementing these changes, you have made important operational changes to the vehicle rental process that will provide enhanced customer service to ITSO Speedy Rentals customers.

15.3.6 Export the enterprise applications

To deploy the new process implementation to WebSphere Process Server, export the VehicleRentAppV2 and VehRentSvcApp modules. Follow these steps:

1. Click **File** → **Export** → **EAR File**, then click **Next**.
2. In the Export dialog, set the EAR project to **VehicleRentV2App** and browse to a directory to export the file to. Click **Finish**.
3. Repeat this process to export VehRentSvcApp.

15.4 Deploying the new process model to WebSphere Process Server

To implement the changes to the vehicle rental process, perform the following steps in WebSphere Process Server:

- ▶ Stop and remove the old process running in WebSphere Process Server.
- ▶ Install the new process, exported from WebSphere Integration Developer.

15.4.1 Stopping the old process

With the WebSphere Process Server server running, follow these steps:

1. Stop the business process template in the VehicleRentApp module by using the administrative console to ensure that no further process instance can be run:
 - a. Open the administrative console.
 - b. Select **Applications** → **Enterprise Applications**, then click the enterprise application **VehicleRentApp**.
 - c. Under Related items, click **EJB Modules**, then select the EJB JAR file **VehicleRentEJB.jar**.
 - d. Under Additional Properties, select **Business processes**.
 - e. Select the vehicle fulfilment process and click **Stop**.

2. Stop the business process template and the human task template in the VehRentSvcApp module:
 - a. Select **Applications** → **Enterprise Applications**, then click the enterprise application **VehRentSvcApp**.
 - b. Under Related items, click **EJB Modules**, then select the EJB JAR file **VehRentSvc.jar**.
 - c. Under Additional Properties, select **Business processes**.
 - d. Select the **Check if vehicle returned** process and click **Stop**.
 - e. Navigate back one level and select **Human Tasks** under Additional Properties.
 - f. Select all task templates and click **Stop**.
3. Make sure that there are no running process instances. Finish all of the running processes by using BPC Explorer as described in 13.3.1, “Using the BPC Explorer as an end user” on page 341.

15.4.2 Installing the new process

1. Uninstall the VehicleRentApp and VehRentSvcApp using the administrative console.
 - a. In the administrative console, select **Applications** → **Enterprise Applications**.
 - b. Check the **VehicleRentApp** enterprise application and click **Stop**.
 - c. Check the **VehicleRentApp** enterprise application and click **Uninstall**.
 - d. Save the changes.
 - e. Repeat this process for VehRentSvcApp.
2. Install the new VehicleRentV2App and VehRentSvcApp enterprise applications that you exported in 15.3.6, “Export the enterprise applications” on page 442. Alternatively, use the EAR files from the additional materials supplied with this book in the Lifecycle\WIDExport directory.

Note: Find instructions for installing EAR files into WebSphere Process Server in 13.2.2, “Installing the application using the administrative console” on page 339.

3. Save your changes.

The new vehicle fulfilment process is now deployed.

15.5 Managing the new business measures model and process model

This section describes how to take the new version of the business measures model that was created in 15.2.2, “Adding new business measures” on page 429 and import it into WebSphere Business Monitor.

15.5.1 Generating the new schema from the new model

First run the schema generator for the new model.

1. Ensure that the Monitor Server is started, and log into the administrative console.
2. Configure the schema generator to use the new Monitor.zip that was exported from WebSphere Business Modeler in 15.2.2, “Adding new business measures” on page 429. Be careful to ensure that the option to ignore older deployments is *unchecked*.

Important: If you do not uncheck this option, the schema generator will create scripts and schema definitions for a completely new set of database tables. You do not want this because you want to deploy the new business measures model as a version upgrade to the previous one, and to retain the data that was previously captured against the old model.

3. Specify a new output directory for the generated schema. Figure 15-22 shows the updated settings.

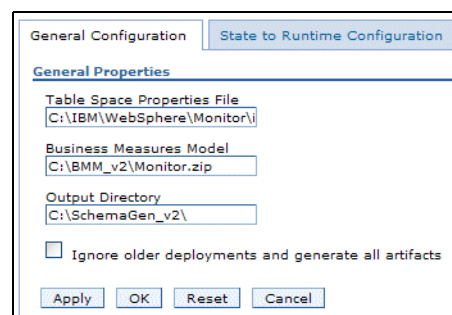


Figure 15-22 Updating schema generation configuration

4. Generate the new schema by clicking **Generate** in the generation section of the administrative console. The schema generator will create scripts that make changes only to the existing database tables.

5. Review the contents of the the newly generated artifacts directory (C:\SchemaGen_v2\ in our example).
6. Copy the same directory to the Dashboard Client as C:\DashboardSchemaGen_v2\.
7. Stop all existing replication daemons at this stage.

Note: We recommend that you also complete all of the existing process instances that were run using the previous model before deploying the new business measures model to reduce the risk of some processes not being monitored. In our scenario this should not happen because both the old and new process models should be recognized by both the old and new business measures model.

8. Shut down the Monitor Server and Dashboard Client to ensure that the databases are not in use.

Important: You should back up the databases at this stage.

9. Now run the generated DDL scripts from C:\SchemaGen_v2\schemagen\ using the same command sequences that you used in “Database configuration” on page 360. You must update all three databases: State, Runtime, and History. Figure 15-23 on page 446 shows example script output.

Important: Be sure to use the Version 2 scripts from C:\SchemaGen_v2\ - do not re-run the Version 1 scripts from C:\SchemaGen\.

```

C:\DB2 CLP
C:\Documents and Settings\Administrator>cd \SchemaGen_v2\schemagen
C:\SchemaGen_v2\schemagen>db2 terminate
DB20000I The TERMINATE command completed successfully.
C:\SchemaGen_v2\schemagen>set DB2CODEPAGE=1208
C:\SchemaGen_v2\schemagen>db2 connect to STATE user db2admin
Enter current password for db2admin:

    Database Connection Information
    Database server        = DB2/NT 8.2.1
    SQL authorization ID   = DB2ADMIN
    Local database alias   = STATE

C:\SchemaGen_v2\schemagen>db2 +c -stvf state.ddl > state.log
C:\SchemaGen_v2\schemagen>notepad state.log
C:\SchemaGen_v2\schemagen>db2 commit
DB20000I The SQL command completed successfully.
C:\SchemaGen_v2\schemagen>db2 disconnect STATE
DB20000I The SQL DISCONNECT command completed successfully.
C:\SchemaGen_v2\schemagen>db2 terminate
DB20000I The TERMINATE command completed successfully.
C:\SchemaGen_v2\schemagen>_

```

Figure 15-23 Updating the state database with the Version 2 schema

10. Extract the zip files from the schema generation output directory as you did with the Version 1 model, but this time extract to Version 2 directories.

Restriction: A known issue with the generation of the Version 2 replication configuration scripts makes it necessary to manually alter two of the scripts between extracting them and running them. This issue occurs when an additional task is added to the process model, as we did in our scenario.

The problem is found in the two target scripts:

- ▶ State_to_Runtime_setup_target.bat
- ▶ Runtime_to_History_setup_target.bat

Each script will fail if it is not edited because the new scripts will try to create indices on the replication CCP tables, but the commands will fail because those indices already exist from the previous definition.

To fix this, you must change the scripts so that this situation is interpreted as a warning instead of an error. To do this, we found the section shown below in both scripts and added the text shown in bold.

```
if "%WBIRMSTATUS%"=="0" goto CMD_37_SUCCESS

@REM
*****
@REM
*****
@REM If Available. The following IF's check for Warning Conditions.
if "%WBIRMSTATUS%"=="2" goto CMD_37_WARNING
if "%WBIRMSTATUS%"=="1" goto CMD_37_WARNING
```

This issue will be resolved in a later release.

11. After editing the scripts for the restriction above, run the new replication daemon configuration scripts the same way you did for the Version 1 model.

Important: Be sure that you run the new scripts, not the Version 1 scripts.

12. When the scripts have all completed, restart all of the servers and the daemons as you did for the Version 1 deployment.

Note: There are no new start scripts for the replication daemons. The new setup scripts change the configuration but you should use the original start scripts to actually start the daemons after they have been updated.

13. Log in to the Monitor Server administrative console and go to the model import screen as you did for Version 1. Browse to the new model (be sure to

use the new one and not the previous version) and click **Import** to import the Version 2 model. When the import succeeds, you can begin to run new processes and have the data captured.

The next stage is to update the configuration of the DB2 Cube Views.

14. On the Dashboard Client, start the OLAP Center and use the same steps as for Version 1 to import the Version 2 models. Be careful that you use the new version cubes.

Note: We found that the old cube view definitions no longer existed when we started the OLAP Center. You might find that they still exist for some models, in which case you must use the OLAP Center to manually remove them before proceeding.

15. Start the DB2 Alphablox administrative console and locate the previously defined cubes from the **Administration** → **Cubes** page. You must delete the existing cube definitions, so select each one and click **Delete** until they are all removed.
16. Redefine the cubes one at a time in the same way as for the Version 1 model. Be sure that the names exactly match the cube names in DB2 Cube Views.
17. Start the cubes in DB2 Alphablox as you did for Version 1.
18. The dashboards should now be updated for the latest model. You might want to customize the portlets to show the new measures and KPIs.

15.5.2 Monitoring the new process model

You have now taken a second pass through the SOA lifecycle to model improvements to the process, and to assemble, deploy, and monitor the improved process.

Over time, as the new process model executes, you can use the WebSphere Business Monitor dashboards to examine your process to see whether the modifications were beneficial. You can also use the new business measures to analyze the process execution.

Figure 15-24 on page 449 shows the late vehicle return percentage by location. Compared to Figure 15-14 on page 425, improvements to the process model have significantly reduced the number of late vehicle returns for all classes of vehicle.

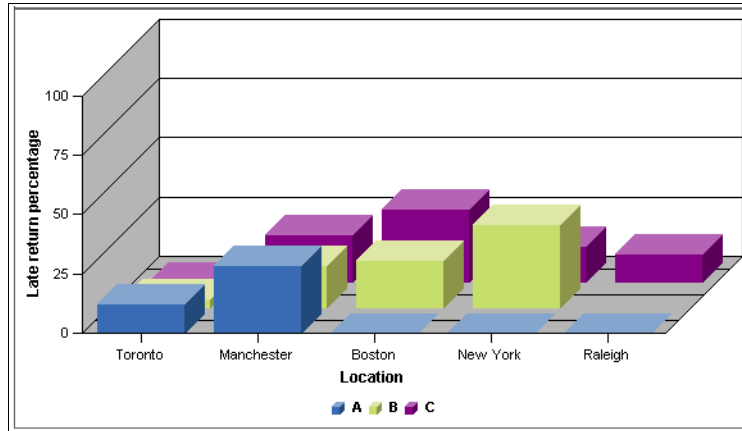


Figure 15-24 Reduced late returns due to improvements in process model

Figure 15-25 also highlights this improvement, showing the late vehicle return percentage by country. This compares favorably to the same measurement taken before the process improvements were implemented, as shown in Figure 15-13 on page 424.

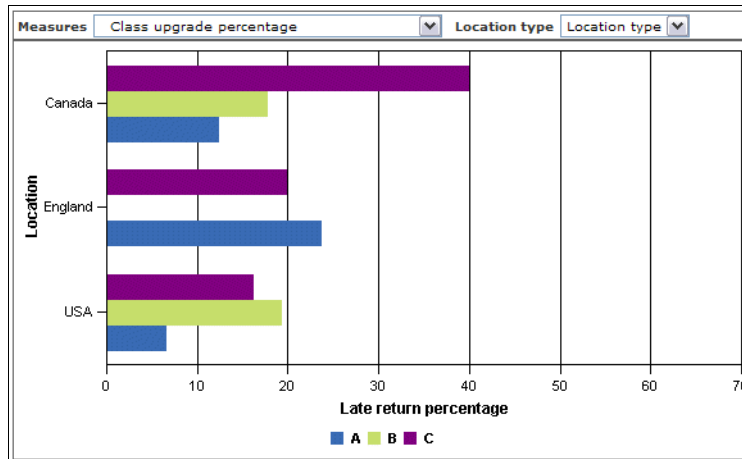
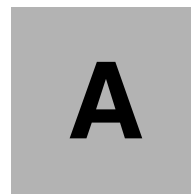


Figure 15-25 Process improvements by country



Part 4

Appendixes



Product installation

This appendix shows how we installed and configured the products that we used throughout the book. We have referenced other sources to avoid duplication of details that are provided elsewhere.

This appendix addresses the following products:

- ▶ WebSphere Business Modeler Version 6.0
- ▶ WebSphere Integration Developer Version 6.0.1
- ▶ WebSphere Process Server Version 6.0.1
- ▶ WebSphere Business Monitor Version 6.0

Runtime topology

Our runtime environment consisted of the following products:

- ▶ WebSphere Process Server Version 6.0.1
- ▶ WebSphere Business Monitor Version 6.0

The supported configuration requires you to execute the business process in a dedicated WebSphere Process Server that is installed on a separate machine from the WebSphere Business Monitor servers. WebSphere Business Monitor then requires two additional servers called the Monitor Server and the Dashboard Client. Figure A-1 shows the overall topology that we created, including the location of the WebSphere Business Monitor databases.

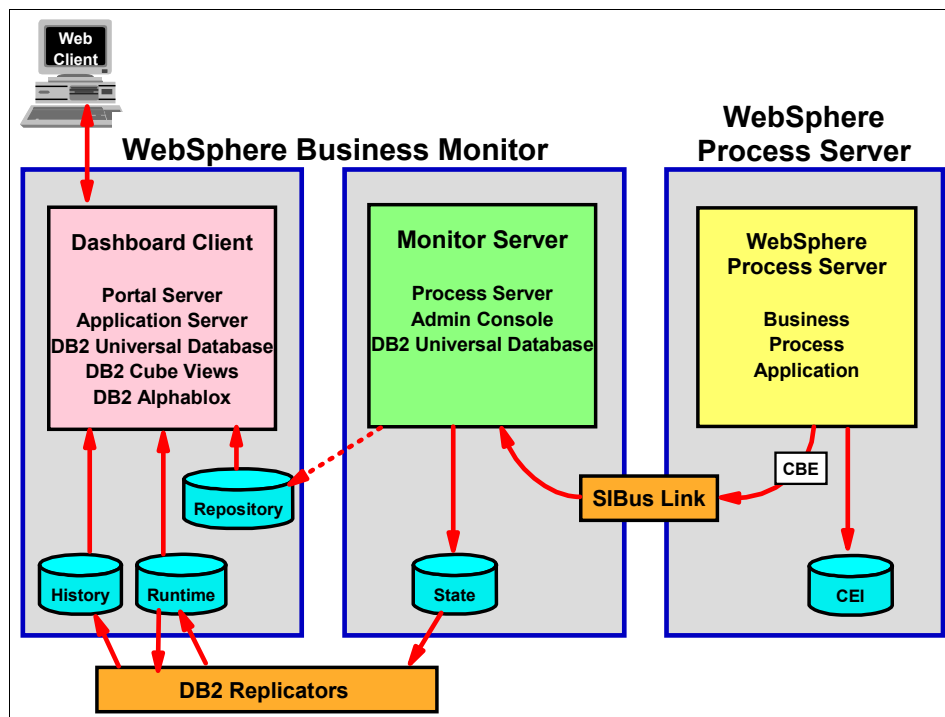


Figure A-1 Runtime topology

WebSphere Business Modeler Version 6.0

We installed WebSphere Business Modeler Advanced Edition Version 6.0 on Microsoft Windows XP Professional by using setup.exe from the root of the installation CD. This started the Launchpad, and we proceeded through the installation, accepting all defaults (Figure A-2).

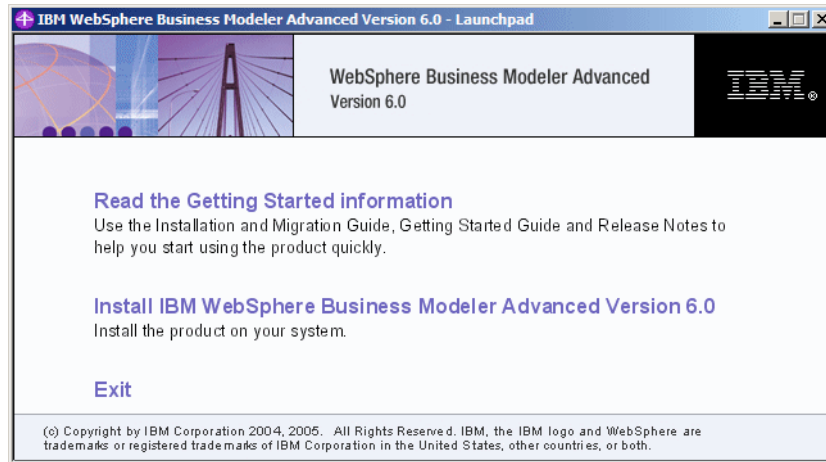


Figure A-2 WebSphere Business Modeler Version 6.0 Launchpad

Installing WebSphere Business Modeler into WebSphere Integration Developer

An alternative is to install the WebSphere Business Modeler as plug-ins to a WebSphere Integration Developer installation. This enables you to use a single Eclipse environment that contains the capabilities of both products. This can be useful for a smaller team in which an individual might perform multiple roles.

To do this, install the WebSphere Integration Developer, then add the WebSphere Business Modeler plug-ins by following the instructions in the Information Center at:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.btools.help.modeler.doc/doc/tasks/installing/installplugin.html>

At the time we wrote this book, we found that we could not successfully apply all of the fixes for both products in this configuration, so we used the products separately throughout the book.

Fixes

After the product installation was completed, we downloaded Interim Fix 3 from the support Web site and installed this fix in accordance with the instructions provided in the accompanying readme file. If further fixes are available, we recommend that you download and install them as well.

Note: Find support for WebSphere Business Modeler, including fixes, at:
<http://www.ibm.com/software/integration/wbimodeler/advanced/support/>

WebSphere Integration Developer Version 6.0.1

We installed WebSphere Integration Developer Version 6.0.1 on Microsoft Windows XP Professional by using launchpad.exe from the first of the installation CD-ROMs. This started the Launchpad and we proceeded through the installation, opting to install the WebSphere Process Server integrated test environment and accepting the default for all other options (Figure A-3).

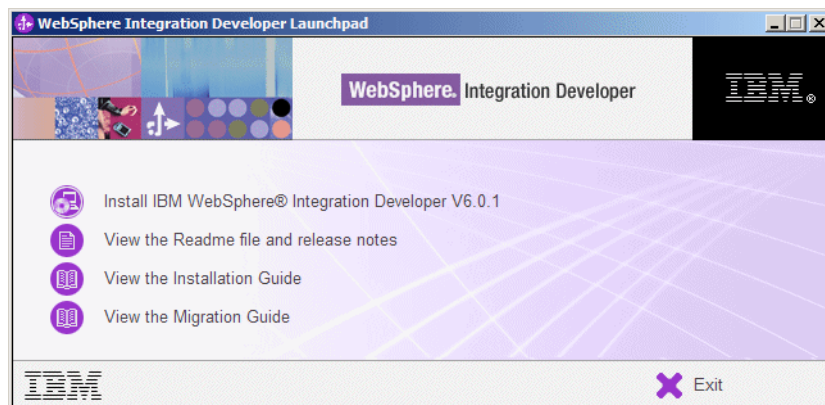


Figure A-3 WebSphere Integration Developer Version 6.0.1 Launchpad

Fixes

After the product installation was completed, we used the Rational Product Updater to download and install all available fixes. To launch it, select **Start → All Programs → IBM WebSphere → Integration Developer V6.0.1 → Rational Product Updater**. We installed Interim Fixes 001, 002, and 003. If further fixes are available, we recommend that you download and install these as well.

Note: Find support for WebSphere Integration Developer, including fixes, at:
<http://www.ibm.com/software/integration/wid/support/>

WebSphere Process Server Version 6.0.1

We installed WebSphere Process Server Version 6.0.1 on a dedicated server running Microsoft Windows 2003 Server. This was a different server from the two other machines that we subsequently used for installing WebSphere Business Monitor Version 6.0.

Restriction: Installing the WebSphere Process Server that will execute your business processes on the same machines as WebSphere Business Monitor is not supported.

Note: The Information Center for WebSphere Process Server can be found at:
<http://www.ibm.com/software/integration/wps/library/infocenter/>

To monitor processes that will run on this WebSphere Process Server, you must enable the Common Event Infrastructure because this is used for passing events to the WebSphere Business Monitor server.

We also decided to use DB2 instead of Cloudscape for both the business process container and the Common Event Infrastructure. We began by installing DB2 Universal Database Enterprise Server Edition Version 8.2 on the machine.

After DB2 was installed, we installed WebSphere Process Server according to the Information Center instructions. You can re-create the same configuration by following the steps we used:

1. Start **launchpad.exe** from the root of the first installation CD-ROM.

2. Select **WebSphere Process Server installation** as shown in Figure A-4.

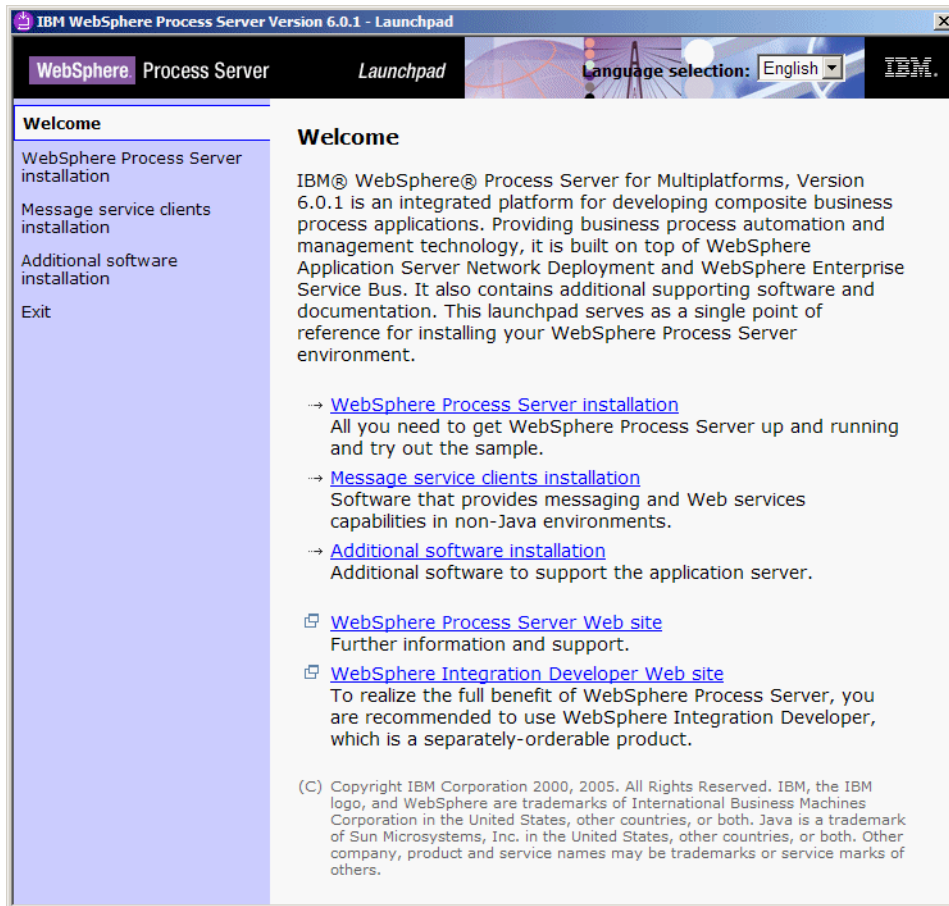


Figure A-4 WebSphere Process Server Launchpad

3. Select **Launch the Installation Wizard for WebSphere Process Server for Multiplatforms**.

4. Step through the wizard until you reach the installation root directory option. Instead of accepting the default value, we used C:\WPS601\ to reduce the likelihood of encountering problems with Windows limitations on directory name lengths (Figure A-5). Click **Next**.

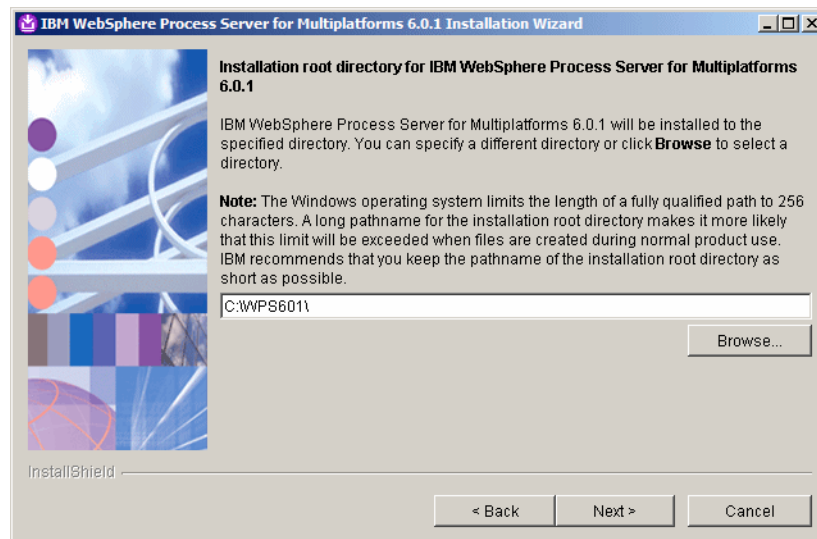


Figure A-5 Changing the default installation directory

5. Select **Custom installation** and click **Next** as shown in Figure A-6.

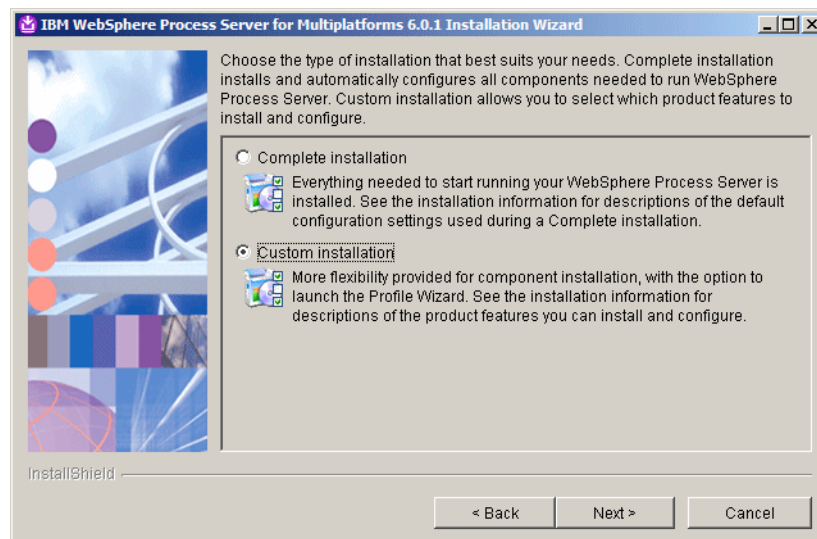


Figure A-6 Custom installation

6. Deselect the **Sample Applications Gallery** and click **Next** as shown in Figure A-7. It is a good idea to remove the samples from a production system.

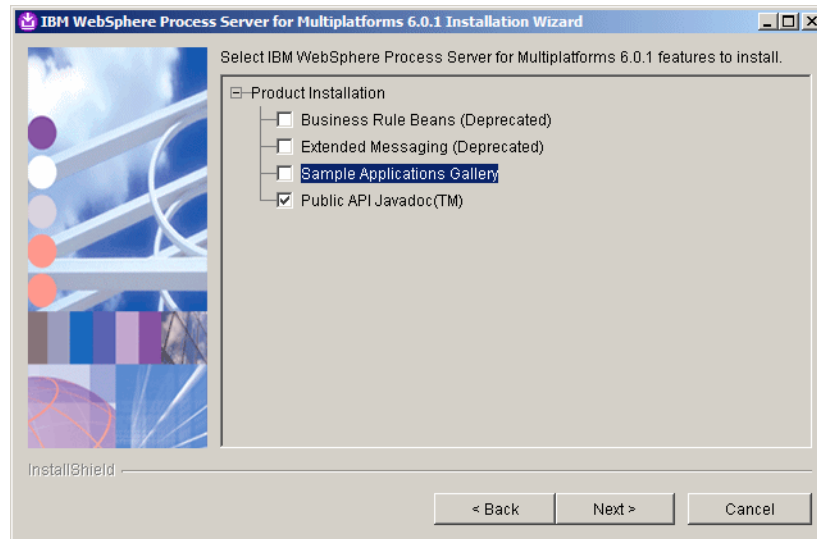


Figure A-7 Deselect the Sample Applications Gallery

7. Proceed with the installation until it completes, then click **Next** to confirm launching the Profile Wizard as shown in Figure A-8.

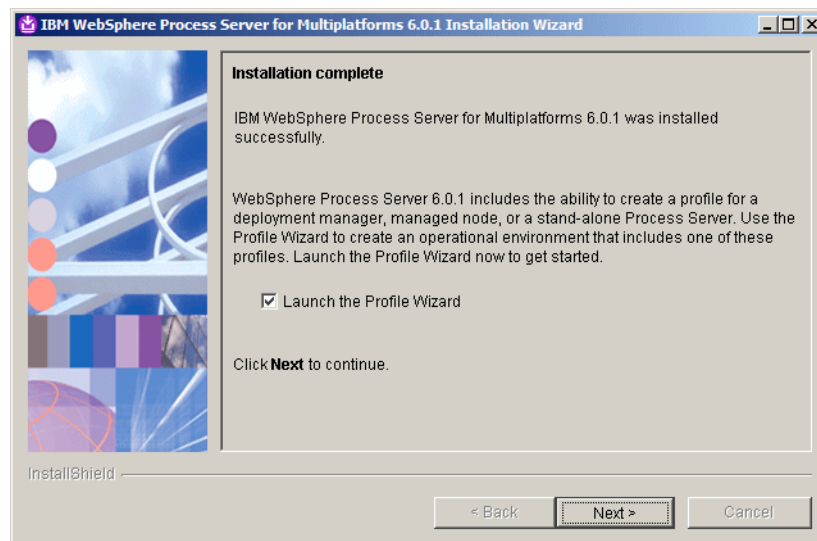


Figure A-8 Launching the Profile Wizard

8. Continue stepping through the wizard and opt to create a **Stand-alone profile** named ProcSrv01. Accept the default directory, node name, host name and port numbers. Click **Next**.
9. De-select the option to run as a Windows service as shown in Figure A-9, and click **Next**.

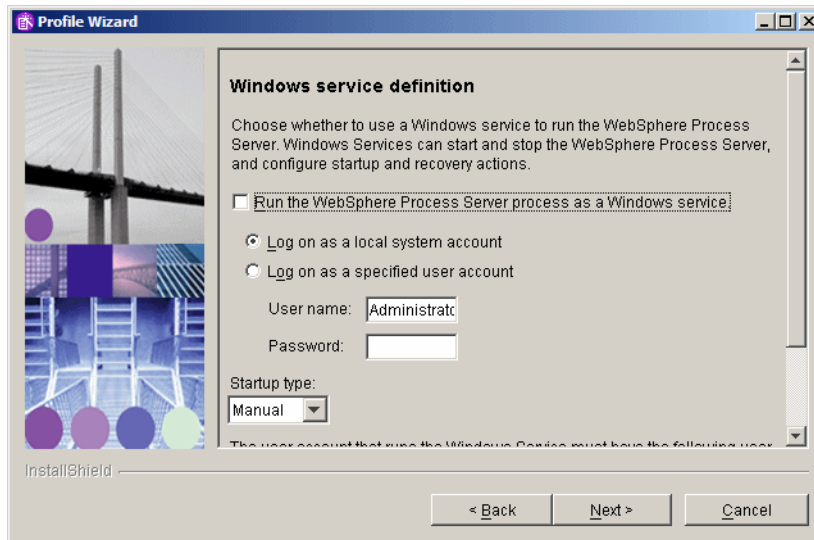
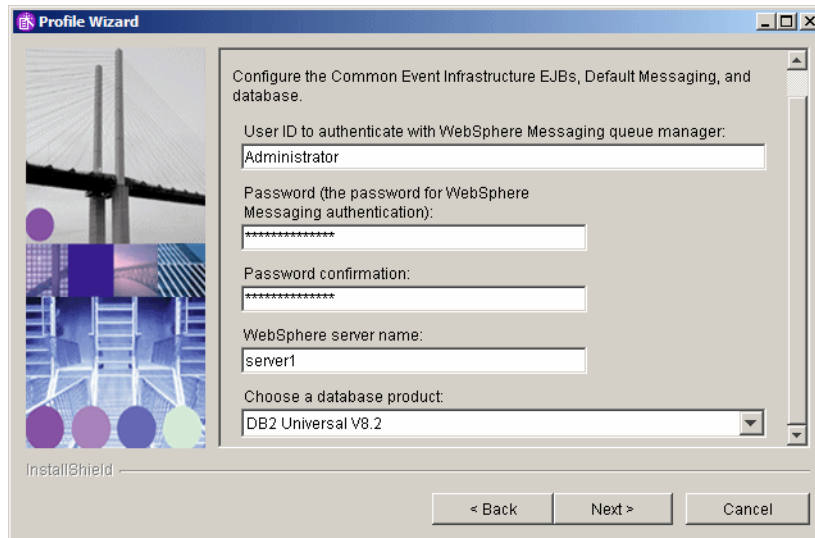


Figure A-9 Disabling the Windows service

10. Do *not* configure the Service Integration Bus in secured mode. Click **Next**.

11. For the Common Event Infrastructure Configuration, specify the administrative user ID on your system for connections to the messaging infrastructure, and select the database type. We opted to use **DB2 Universal V8.2** for the database product because we had already installed DB2 on the same system, specifically to use with the Common Event Infrastructure. (Figure A-10 shows our overall settings.) Click **Next**.



The screenshot shows a 'Profile Wizard' window with a title bar containing a small icon and the text 'Profile Wizard'. The window is divided into two main sections. On the left is a decorative panel with a background image of a bridge and a series of colored circles (purple, blue, green) at the bottom. Below the image, the text 'InstallShield' is visible. The right section contains the configuration fields. At the top, it says 'Configure the Common Event Infrastructure EJBs, Default Messaging, and database.' Below this are five input fields: 'User ID to authenticate with WebSphere Messaging queue manager:' with the value 'Administrator'; 'Password (the password for WebSphere Messaging authentication):' with masked characters '*****'; 'Password confirmation:' with masked characters '*****'; 'WebSphere server name:' with the value 'server1'; and 'Choose a database product:' with a dropdown menu showing 'DB2 Universal V8.2'. At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.

Profile Wizard

Configure the Common Event Infrastructure EJBs, Default Messaging, and database.

User ID to authenticate with WebSphere Messaging queue manager:
Administrator

Password (the password for WebSphere Messaging authentication):

Password confirmation:

WebSphere server name:
server1

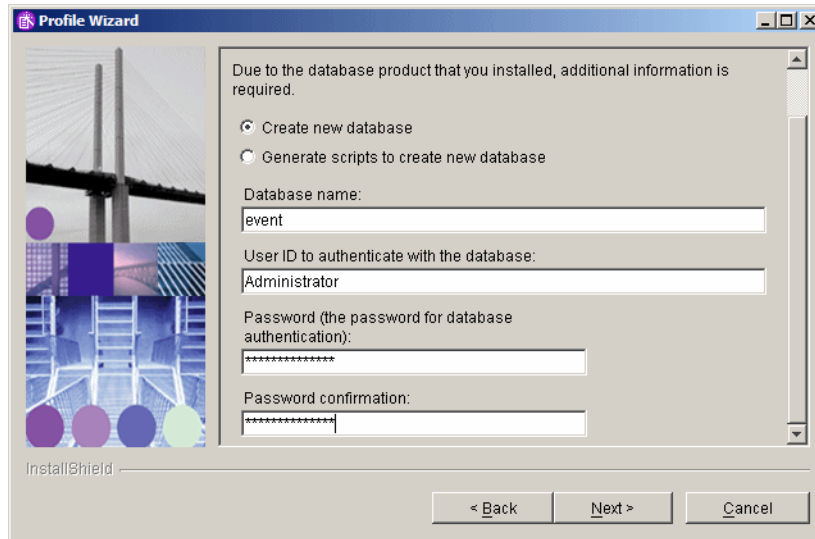
Choose a database product:
DB2 Universal V8.2

InstallShield

< Back Next > Cancel

Figure A-10 Common Event Infrastructure settings

12. Specify that you want to create a new database and again use your administrative user ID for connections to this database, as shown in Figure A-11. Click **Next**.



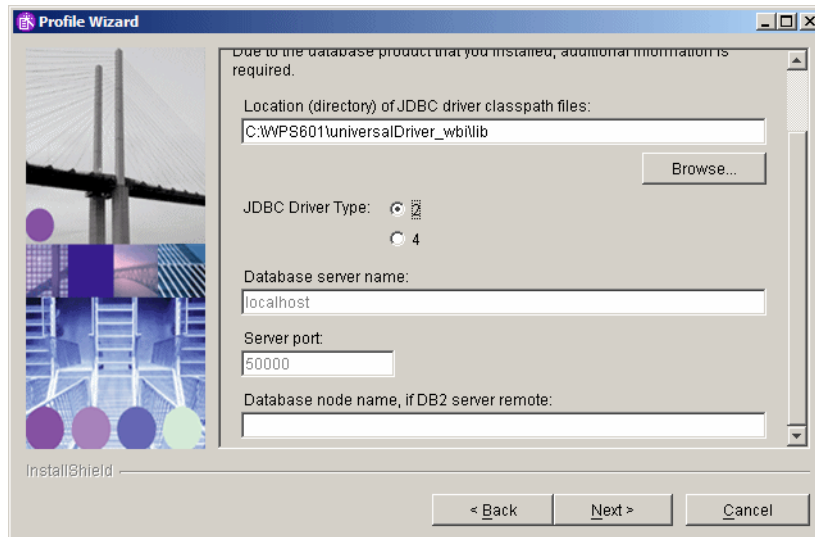
The screenshot shows the 'Profile Wizard' window. On the left is a decorative graphic with a bridge and colorful circles. The main area contains the following text and fields:

- Due to the database product that you installed, additional information is required.
- ☒ Create new database
- ☐ Generate scripts to create new database
- Database name:
- User ID to authenticate with the database:
- Password (the password for database authentication):
- Password confirmation:

At the bottom are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure A-11 Creating the event database

13. Select a **JDBC Driver Type 2** connection to the database (because DB2 is installed on the same machine), as shown in Figure A-12. Click **Next**.



The screenshot shows the 'Profile Wizard' window. On the left is a decorative graphic with a bridge and colorful circles. The main area contains the following text and fields:

- Due to the database product that you installed, additional information is required.
- Location (directory) of JDBC driver classpath files:
- JDBC Driver Type: ☒ 2 ☐ 4
- Database server name:
- Server port:
- Database node name, if DB2 server remote:

At the bottom are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure A-12 Database Configuration Information for CEI

14. Choose *not* to configure the sample Business Process Choreographer at this stage because the sample configuration uses Cloudscape, not DB2. Click **Next**.
15. Choose *not* to create an Application Scheduler. Click **Next**.
16. Select **DB2 Universal** as the database product name as shown in Figure A-13, then click **Next**.

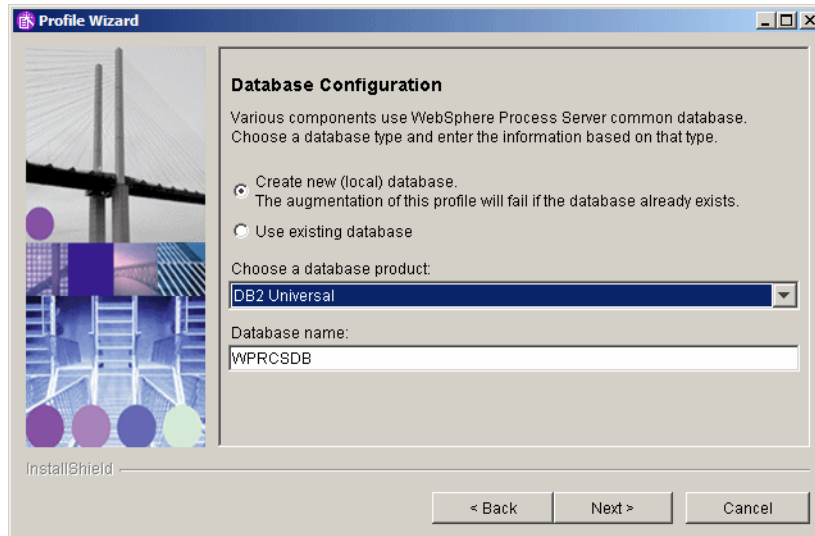
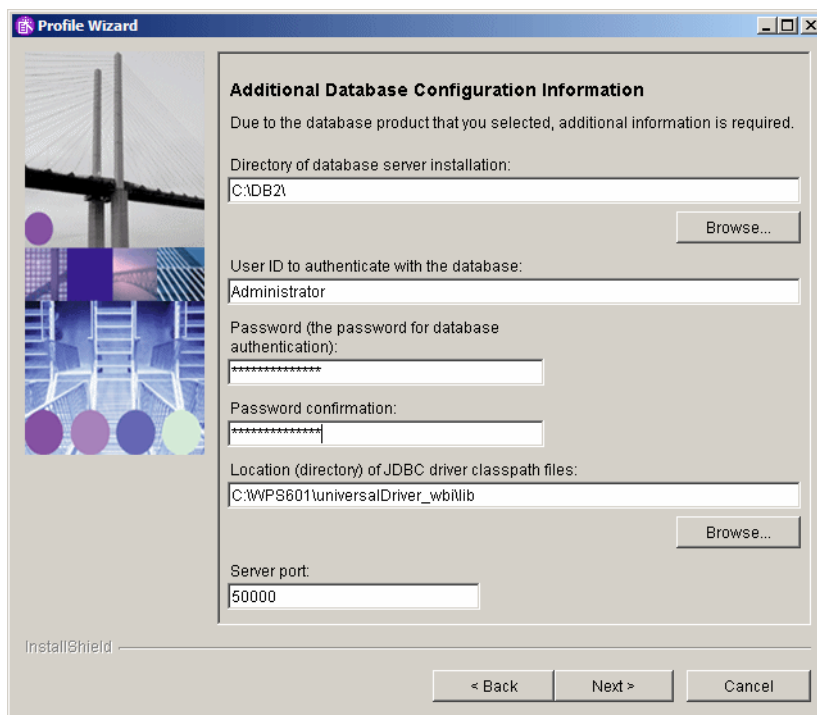


Figure A-13 Specifying DB2 Universal for the common database

17. Name the database server directory C:\DB2\ and provide the administrative user and password for connecting to the database as shown in Figure A-14, and click **Next**.



The screenshot shows a 'Profile Wizard' window with a title bar containing a shield icon and the text 'Profile Wizard'. On the left side, there is a vertical strip with a background image of a bridge and four colored circles (purple, blue, green, yellow). The main area of the window is titled 'Additional Database Configuration Information' and contains the following fields and buttons:

- 'Directory of database server installation:' with a text box containing 'C:\DB2\' and a 'Browse...' button.
- 'User ID to authenticate with the database:' with a text box containing 'Administrator'.
- 'Password (the password for database authentication):' with a text box containing '*****'.
- 'Password confirmation:' with a text box containing '*****'.
- 'Location (directory) of JDBC driver classpath files:' with a text box containing 'C:\WPS601\universalDriver_wbi\lib' and a 'Browse...' button.
- 'Server port:' with a text box containing '50000'.

At the bottom of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'. The 'InstallShield' logo is visible in the bottom left corner.

Figure A-14 Specifying additional information for the common database

18. Click **Next** twice more to start profile creation and wait for this to complete. When it finishes, deselect the option to launch the First Steps console and click **Finish**.

Note: At this point the product is installed and a profile has been created but the Business Process Choreographer configuration is not yet created. The next steps explain how to do this.

19. Before you install the business process container, you must create the BPEDB database in DB2. The BPEDB database is used by the business process container.

Open a DB2 Command Window and run the create database script using the following command:

```
db2 -tf C:\WPS601\ProcessChoreographer\createDatabaseDb2.dd1
```

Note: Use this script is only for non-production stand-alone development, evaluation, or demonstration purposes. Otherwise, you must create the database manually, adjust the DB2 parameters, and consider using dedicated table space containers.

20. Ensure that the CLI packages are bound to the new database by running the following commands:

```
db2 connect to BPEDB
db2 bind C:\PROGRA~1\IBM\SQLLIB\bnd\@db2cli.lst blocking all grant
public
```

Note: You should alter the second statement depending on where you installed DB2. We installed it into the default location, C:\Program Files\IBM\SQLLIB\ . We also found that this statement only succeeds if you use Windows short names, so we had to convert C:\Program Files\ to C:\PROGRA~1\ .

21. When the database creation is complete, start the WebSphere Process Server using **Start → All Programs → IBM WebSphere → Process Server 6.0 → Profiles → ProcSrv01 → Start the server.**
22. Launch the administrative console using **Start → All Programs → IBM WebSphere → Process Server 6.0 → Profiles → ProcSrv01 → Administrative console.**
23. Log in to the console, expand **Servers**, click **Application Servers**, and click **server1**.
24. Expand **Business process container settings** and click **Business process container**.

25. Note the warning messages, shown in Figure A-15, that both the business process container and the human task manager are not installed at this stage. Click **Business process container installation wizard**.

The screenshot shows a web-based configuration interface titled 'Application servers'. At the top, there is a 'Messages' section with two warning icons and text:

- ⚠ The business process container is not installed. If you want to install it, click the business process container installation wizard link. After the container is installed, only the general properties and custom properties can be changed.
- ⚠ The human task manager is not currently installed. If you want to use any human interaction, you must also install and configure the human task manager.

Below the messages, the breadcrumb path is 'Application servers > server1 > Business process container'. A descriptive paragraph states: 'The business process container provides services to run business processes within an application server. After installing the business process container, you can only change the general properties on this page and on the custom properties page.'

The 'Configuration' tab is active, showing two columns of settings:

General Properties	Additional Properties
<input type="checkbox"/> Enable Common Event Infrastructure logging	Business process container installation wizard
<input type="checkbox"/> Enable audit log	Human task container
Retry limit: <input type="text" value="5"/>	Runtime Configuration
Retention queue message limit: <input type="text" value="20"/>	Custom Properties
Retention queue: <input type="text" value="jms/BPERetQueue"/>	
Hold queue: <input type="text" value="jms/BPEHldQueue"/>	
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

Figure A-15 Starting the business process container installation wizard

26. The first page of the business process container installation wizard asks for the database configuration information. Specify the **DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA))** template then enter your administrative user details as shown in Figure A-16. Click **Next**.

→ Step 1 : Select the database configuration that this business process container will use

Select the desired JDBC provider. A new XA datasource will be created for this provider. ?

* JDBC providers

DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA)) ?

Implementation class name

com.ibm.db2jcc.DB2XADataSource

? The Java class name of the JDBC driver implementation.

Classpath

`\${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar`
`\${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar`
`\${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisu`

? A list of paths or JAR file names which together form the location for the resource provider classes. Class path entries are separated by using the ENTER key and must not contain path separator characters (such as ` ` or ` `). Class paths may contain variable (symbolic) names that can be substituted using a variable map. Check your driver installation notes for specific JAR file names that are required.

Data source user name

Administrator

? The user name used to access the data source

Data source password

? The password used to access the data source

Custom Properties

databaseName=BPEDB
driverType=2
serverName=RICHJT42p.hursley.ibm.com
portNumber=50000
description=DataSource for Process Choreographer
traceLevel=
traceFile=
fullyMaterializeLobData=true
resultSetHoldability=2
currentPackageSet=
readOnly=false

? Consult the help documentation for more information about properties that might be required by the JDBC provider.

Next

Reset

Cancel

Figure A-16 Specifying database configuration for the business process container

468 Patterns: SOA Foundation - Business Process Management Scenario

- f. For the JMS provider, use the **Default messaging provider** and enter your administrative user information and the Administrators group that this user belongs to, as shown in Figure A-17. Click **Next**.

→ Step 2 : Select the JMS Provider and Security settings

Select the JMS provider. ?

JMS providers

* JMS providers

Queue manager

Classpath

* JMS user ID

* JMS password

Business process container security configuration

* JMS API User ID

* JMS API password

* Administrator security role mapping

* System monitor security role mapping

Previous Next Reset Cancel

? A JMS provider enables asynchronous messaging based on the Java Message Service (JMS). It provides J2EE connection factories to create connections for specific JMS queue or topic destinations. JMS provider administrative objects are used to manage JMS resources for the associated JMS provider.

? The name of the external JMS provider queue manager that this business process container will use.

? A list of paths or JAR file names which together form the location for the resource provider classes. Class path entries are separated by using the ENTER key and must not contain path separator characters (such as '/' or '\'). Class paths may contain variable (symbolic) names that can be substituted using a variable map. Check your driver installation notes for specific JAR file names that are required.

? The queue connection factory uses this user ID to establish a connection to the queue.

? The password for the user ID entered above

? The user ID that the business process container uses when processing asynchronous API calls.

? The password for the user ID entered above

? The group from the domain's user registry that is associated with the business process administrator role.

? The group from the domain's user registry that is associated with the business process system monitor role.

Figure A-17 Specifying the JMS Provider settings

27. On the final configuration window, accept all of the defaults for creating new JMS resources, then select **Enable the Common Event Infrastructure logging** (and optionally, audit logging) as shown in Figure A-18. Click **Next**.

Important: If you do not enable the Common Event Infrastructure logging then you will not be able to monitor your business processes using WebSphere Business Monitor.

→ Step 3 : Select the JMS resources, BPC Explorer, and logging

Select the JMS resources and whether the Business Process Choreographer Explorer will be installed. ?

Select the JMS resources.

☒ Create new JMS resources using default values ?

☐ Select existing JMS resources ?

Connection Factory: CommonEventInfrastructure_QueueCF ? A connection factory is used to create connections to the associated JMS provider.

Internal queue: CommonEventInfrastructure_Queue ? The internal processing queue for this business process container.

External request processing queue: CommonEventInfrastructure_Queue ? The input queue for the JMS API for this business process container.

Hold queue: CommonEventInfrastructure_Queue ? The hold queue used by this business process container for messages that have been retried and failed the maximum number of times.

Retention queue: CommonEventInfrastructure_Queue ? The retention queue used by this business process container to store messages that have failed, but will be retried.

Business Process Choreographer Explorer.

☒ Select this check box to install the Business Process Choreographer Explorer (Web client). ?

State observer logging

☒ Enable Common Event Infrastructure logging for all processes running in this container. ?

☒ Enable audit logging for all processes running in this container. ?

Previous Next Reset Cancel

Figure A-18 Specifying the logging information

28. Review the summary on the next page and click **Finish** to complete the business process container configuration. A message is displayed to indicate success. Click the **Save to Master Configuration** link and then **Save**.

Note: The business process container is now configured. Next you must configure the human task container.

29. Expand **Servers** and click **Application Servers**, then click **server1**.
30. Expand **Human task container settings** and click **Human task container**.

31. The business process container is now shown as installed, and the warning message says that the human task manager is still not installed. Click **Human task container installation wizard** as shown in Figure A-19.

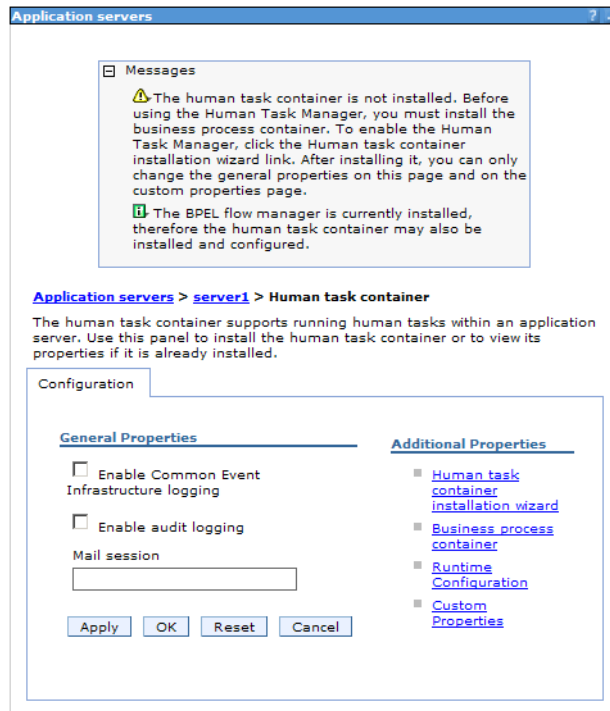


Figure A-19 Starting the human task container installation wizard

32. The first page of the human task manager installation wizard asks for the JMS provider configuration information. Use the **Default messaging provider** and specify your administrative user information and the Administrators group that this user belongs to as shown in Figure A-20. Click **Next**.

→ Step 1 : Select the JMS provider and security settings

Select the same JMS provider and queue manager that is used by the business process container. ?

JMS providers

* JMS providers

Queue manager

Classpath

* JMS user ID

* JMS password

Human Task Manager security configuration

* Escalation user ID

* Escalation password

* Administrator security role mapping

* System monitor security role mapping

Next Reset Cancel

? A JMS provider enables asynchronous messaging based on the Java Message Service (JMS). It provides J2EE connection factories to create connections for specific JMS queue or topic destinations. JMS provider administrative objects are used to manage JMS resources for the associated JMS provider.

? The name of the external JMS provider's queue manager that the Human Task Manager will use. This must be the same one that is used by the business process container.

? A list of paths or JAR file names which together form the location for the resource provider classes. Class path entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Class paths may contain variable (symbolic) names that can be substituted using a variable map. Check your driver installation notes for specific JAR file names that are required.

? The queue connection factory uses this user ID to establish a connection to the queue.

? The password for the user ID entered above

? The user ID that the Human Task Manager will use when processing escalations.

? The password for the user ID entered above

? The group from the domain's user registry that is associated with the human task administrator role.

? The group from the domain's user registry that is associated with the human task system monitor role.

Figure A-20 Specifying JMS provider information

33. On the final configuration window, accept all of the defaults for creating new JMS resources, and select **Enable the Common Event Infrastructure logging** (and optionally, audit logging) as shown in Figure A-21. Click **Next**.

→ Step 2: Select JMS resources, mail session, scheduler, and logging

Select the JMS resources, mail session, scheduler, and logging for the Human Task Manager. ?

Select the JMS resources.

☒ Create a new JMS resources using default values. ?

☐ Select existing JMS resources. ?

Connection Factory: BPECF ? A connection factory is used to create connections to the associated JMS provider.

Hold queue: BPERetQueue_wps601Node01_server1 ? The hold queue for human task messages that could not be processed.

Retention queue: BPERetQueue_wps601Node01_server1 ? The retention queue (used internally by the container) for the Human Task Manager.

Mail session resource

☐ Mail session for Human Task Manager ?

State observer logging

☒ Enable Common Event Infrastructure logging ?

☒ Enable audit logging for all human tasks ?

Previous Next Reset Cancel

Figure A-21 Enabling logging

34. Review the summary on the next page and click **Finish** to complete the human task container configuration. When this completes, a message appears indicating success. Click the **Save to Master Configuration** link and then **Save**.
35. Click **Logout** and restart the server to complete the overall configuration.

Fixes

When we wrote this book, there were no fixes to download. We recommend that you check the support page and obtain any fixes that are made available.

Note: The support page for WebSphere Process Server can be found at:

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

WebSphere Business Monitor Version 6.0

We installed WebSphere Business Monitor Version 6.0 on two machines that were both running Microsoft Windows 2003 Server. The first machine was used for the Monitor Server and the State database. The second machine was used for Dashboard Client and the Runtime, History, and Repository databases. Figure A-1 on page 454 shows a complete picture of our runtime topology.

This topology is likely to be the best performing two-server configuration because it locates the databases on the machines that access them most frequently. For example, the Dashboard Client makes extensive use of the Runtime and History databases, whereas the Monitor Server uses the State database to persist the data it extracts from incoming events.

We used the Launchpad on both machines by starting `launchpad.bat` (Figure A-22) and installed the product by following the steps described in Appendix B of the redbook *WebSphere Process Integration V6: Business Process Management Modeling through Monitoring*, SG24-7148. The only difference in our environment was that, as described above, we created the State database only on the Monitor Server then created the other databases on the Dashboard Client instead.

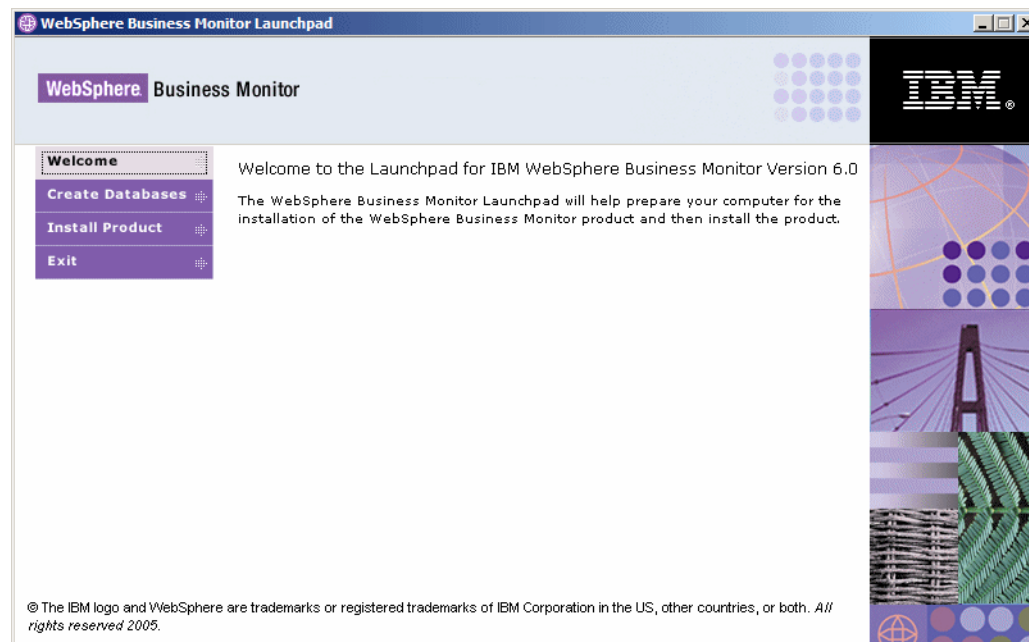


Figure A-22 WebSphere Business Monitor Launchpad

We also strongly recommend that you follow the validation steps to verify your installation. These are described in the Information Center in the section **WebSphere Business Monitor → Introduction → Release Notes → About this release → Known limitations → WebSphere Business Monitor installation validation**.

Note: The WebSphere Business Monitor Information Center can be found at:
<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp>

Fixes

We downloaded and applied a fix for the DB2 Alphablox used by WebSphere Business Monitor. This fix is mentioned in the Information Center release notes under **Introduction → Release Notes → About this release → Known limitations → Other known limitations**. It can be downloaded from:

http://www.ibm.com/support/docview.wss?rs=2026&context=SSCPX3&dc=D600&uid=swg21227687&loc=en_US&cs=UTF-8&lang=en

Note: You should also check the WebSphere Business Monitor support page for any other fixes that might have been made available:

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

To install the fix, we first installed the product then ensured that the Dashboard Client server was stopped. We renamed our existing aasserver.jar and cubing.jar files in the Dashboard Client's C:\IBM\WebSphere\AppServer\lib\ext\ directory.

Then we extracted the contents of the downloaded fix zip file to a temporary directory to obtain the three JAR files, and copied these replacement files to the C:\IBM\WebSphere\AppServer\lib\ext\ directory.

Tip: The fix zip file contains three JAR files and a text document in a directory called alphablox830b157. You should *not* reproduce this directory structure when you copy the files into your Dashboard Client directory C:\IBM\WebSphere\AppServer\lib\ext\.



B

WebSphere Business Monitor system administration

This appendix provides a quick reference for useful system administration information as well as pointers to locations with further information. The information is primarily about WebSphere Business Monitor installations on Windows systems, but also has some relevance to installations based on IBM AIX® platforms. It covers the following topics:

- ▶ Starting and stopping servers
- ▶ Log file locations
- ▶ Administrative consoles
- ▶ User IDs and passwords
- ▶ Model undeployment from WebSphere Business Monitor

Monitor Server

Monitor Server is one of two servers that make up a WebSphere Business Monitor runtime environment. We installed and ran our Monitor Server on a machine with the DNS name monitor.itso.ral.ibm.com.

Monitor Server runs on top of WebSphere Process Server that was installed as part of the prerequisite software; therefore much of the administration is the same as WebSphere Process Server. We have reproduced the most important administration basics for reference here.

Starting the Monitor Server

The following sequence starts the Monitor Server: **Start → Programs → IBM WebSphere → Process Server 6.0 → Profiles → monitor → Start the server.**

Stopping the Monitor Server

The following sequence stops the Monitor Server: **Start → Programs → IBM WebSphere → Process Server 6.0 → Profiles → monitor → Stop the server.**

Monitor Server log files

The server has a standard out log file and a standard error log file in the following locations:

```
C:\IBM\WebSphere\ProcServer\profiles\monitor\logs\server1\SystemOut.log
C:\IBM\WebSphere\ProcServer\profiles\monitor\logs\server1\SystemErr.log
```

Monitor Server administrative console

The administration of the Monitor Server is performed through the administrative console, which is the administrative console found in WebSphere Process Server with additional options specifically related to configuring WebSphere Business Monitor. The administrative console can be started in either of the following ways:

- ▶ **Start → Programs → IBM WebSphere → Process Server 6.0 → Profiles → monitor → Administrative console**

or

- ▶ Open a browser and connect to:

`http://<your_monitor_server>:9060/ibm/console`

In our case, we used `http://monitor.itso.ral.ibm.com:9060/ibm/console/`.

Monitor Server user ID and password

Monitor Server has no user ID or passwords configured by default.

Dashboard Client

Dashboard Client is one of two servers that make up a WebSphere Business Monitor runtime environment. We installed and ran our Dashboard Client on a machine with the DNS name dashboard.itso.ral.ibm.com. We installed it using all of the default options.

Dashboard Client runs on top of WebSphere Portal, which was installed as part of the prerequisite software; therefore, much of the administration is the same as for WebSphere Portal. We have reproduced the most important administration basics for reference here.

Starting the Dashboard Client

The following sequence starts the Dashboard Client: **Start → Programs → IBM WebSphere → Portal Server v5.1 → Start the Server.**

Stopping the Dashboard Client

The following sequence stops the Dashboard Client: **Start → Programs → IBM WebSphere → Portal Server v5.1 → Stop the Server.**

Dashboard Client log files

The server has a standard out log file and a standard error log file in the following locations:

```
c:\ibm\WebSphere\PortalServer\log\SystemOut.log
c:\ibm\WebSphere\PortalServer\log\SystemErr.log
```

Dashboard Client administrative console

Administration of the Dashboard Client is performed through the administrative console, which is the administrative console found in WebSphere Portal with additional options specifically related to configuring WebSphere Business Monitor. Start the administrative console by opening a browser and connecting to:

`http://<your_dashboard_client>:9081/wps/portal/`

In our case, we used `http://dashboard.itso.ral.ibm.com:9081/wps/portal/`.

Dashboard Client user ID and password

The default Dashboard Client user ID and password are both wpsadmin.

DB2 Alphablox

DB2 Alphablox is part of the prerequisite software that is installed and used on the Dashboard Client. Occasionally it might be necessary to work with the administration or log files for DB2 Alphablox.

Starting and stopping DB2 Alphablox

DB2 Alphablox runs as part of the WebSphere Portal environment, so starting and stopping the Dashboard Client also starts and stops the DB2 Alphablox environment.

DB2 Alphablox log files

The log file for DB2 Alphablox can be found at:
C:\IBM\DB2Alphablox\repository\servers\AlphabloxAnalytics\logs\Server.log.

Historical log files are also found in this location, with a date and time added to the file name to indicate when they were created. These are named with the format Server_yymmdd_hhmmss.log.

DB2 Alphablox administrative console

The DB2 Alphablox administrative console can be started in either of the following ways:

- ▶ **Start → Programs → IBM DB2 Alphablox 8.3 → AlphabloxAnalytics → DB2 Alphablox Administrative Pages**

or

- ▶ Open a browser and connect to:

`http://<your_dashboard_client>:9081/AlphabloxAdmin/home/`

In our case, we used:

`http://dashboard.itso.ral.ibm.com:9081/AlphabloxAdmin/home/`

DB2 Alphablox user ID and password

DB2 Alphablox has a default user ID of admin with password password.

DB2 Universal Database

DB2 Universal Database is used to host all of the databases used by WebSphere Business Monitor. It is installed as prerequisite using the Launchpad; therefore it installs into default locations with default user IDs and passwords. DB2 Universal Database has a wide range of system administration, and we have reproduced only the most important administration basics for reference here.

Starting DB2 Universal Database

DB2 Universal Database is installed as a service on Windows and is configured to start automatically. You can use the services control panel to change this behavior if you wish. You can also start the server by right-clicking its icon in the system tray and choosing **Start (DB2)**.

Stopping DB2 Universal Database

DB2 Universal Database is installed as a service on Windows and you can use the services control panel to stop the service. You can also stop the server by right-clicking its icon in the system tray and choosing **Stop (DB2)**.

DB2 Universal Database log files

The log file can be found at C:\IBM\SQLLIB\DB2\db2diag.log.

DB2 Universal Database Control Center

DB2 Universal Database Control Center enables you to perform administrative tasks and view database contents, so it can be a useful tool. You can start it by right-clicking the icon in the system tray and choosing **DB2 Control Center**.

DB2 user IDs and passwords

DB2 Universal Database has a default user ID of db2admin with password monPa55w0rd. The default password is monPa55w on AIX installations.

Undeploying a business measures model

In WebSphere Business Monitor V6.0 it is not possible to automatically remove a model deployment from the databases (including the replication setup). However, you can unload the model (which is the opposite of undeployment) using the

administrative console. This stops the server from capturing any further data for this model.

In some circumstances you might wish to fully remove the data and tables from the databases in addition to unloading the model (for example, when you wish to deploy a newer version of the same model, with the same names, and you want to completely discard all previous information).

In these circumstances, you must manually remove all tables and replication artifacts related to your model, and you can do this only by making reference to the scripts that you originally ran to work out what should be removed.

An alternative technique that we find useful sometimes is to reinstall the databases. The following steps outline this technique.

Warning: This technique removes all data from your entire WebSphere Business Monitor system.

1. On the Monitor Server, open the Windows Control Panel → **Add/Remove Programs**.
2. Select **IBM WebSphere Business Monitor Version 6.0** and click **Change/Remove**.
3. Step through the wizard, opting to uninstall all features as shown in Figure B-1.

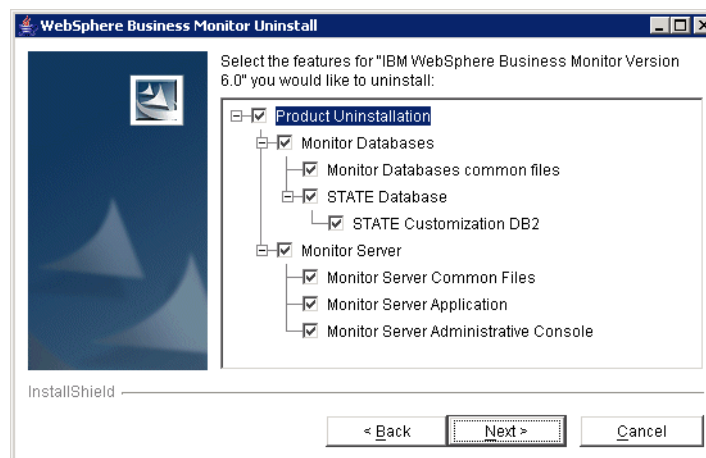


Figure B-1 Uninstalling Monitor Server

Note: It is necessary to remove the databases because you want to re-create them as empty databases without any models deployed. You also have to remove the Monitor Server because this includes the Action Manager database (AAMCAT) that also might contain model deployment data.

4. The uninstallation completes with a message telling you that you must also drop the databases as shown in Figure B-2.

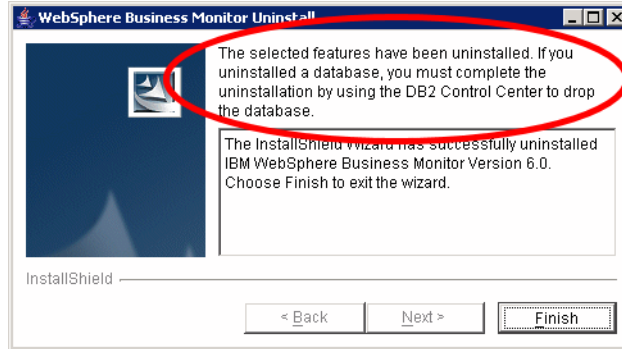


Figure B-2 Uninstallation complete

5. Open DB2 Control Center, select each of the *local* WebSphere Business Monitor databases (STATE, RUNTIME, HISTORY, REPOS, and AAMCAT), and right-click and select **Drop** as shown in Figure B-3.

Important: Be careful to drop only the WebSphere Business Monitor databases and only the ones that are *local* to the Monitor Server (in our example, this meant AAMCAT and STATE). Do not drop the databases that belong to the WebSphere Process Server prerequisite (EVENT and WPRCSDB). Do not drop remote databases — you will drop them locally from the other machine.

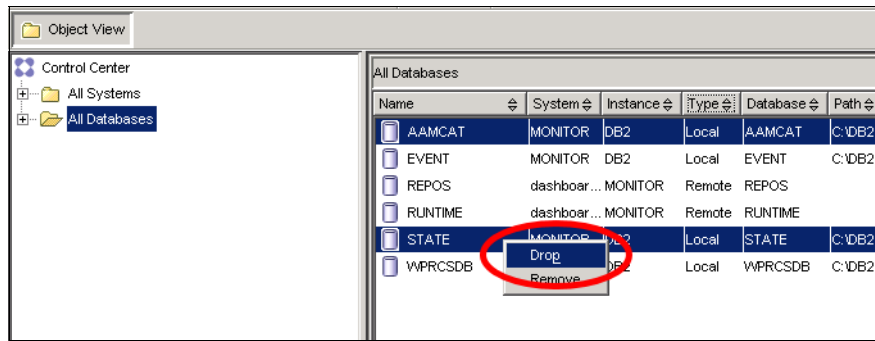


Figure B-3 Dropping the WebSphere Business Monitor databases on Monitor Server

6. The uninstallation is complete on the Monitor Server. Switch to the Dashboard Client machine and run the uninstallation wizard from Add/Remove Programs in the same way as for the Monitor Server.

7. *De-select* the Dashboard Client option when uninstalling as shown in Figure B-4. You only need to drop the databases on this machine.

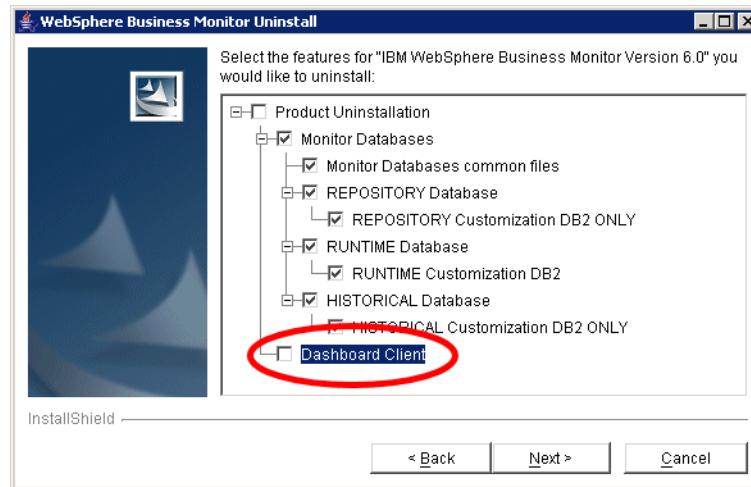


Figure B-4 Uninstalling databases on Dashboard Client

Note: You might have installed your databases in a different configuration than we chose. In our environment the State database was on the Monitor Server machine and the other databases were created on the Dashboard Client machine. You must adapt these steps to your own configuration to ensure that you uninstall all of them and also drop them all.

8. The uninstallation completes with a message that you must also drop the databases, as seen for the Monitor Server in Figure B-2 on page 483.
9. Use DB2 Control Center in the same way as you did on the Monitor Server, dropping only the local WebSphere Business Monitor databases as shown in Figure B-5.

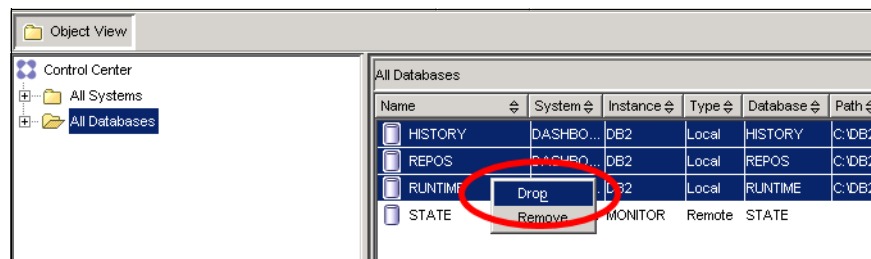


Figure B-5 Dropping WebSphere Business Monitor databases on Dashboard Client

10. The uninstallation is complete on both machines, and all WebSphere Business Monitor databases have been dropped. Next you must manually clear the SIBus repository to remove any messages related to the models that were previously deployed.
11. On the Monitor Server, navigate to the following directory:
C:\IBM\WebSphere\ProcServer\profiles\monitor\databases\com.ibm.ws.sib\
12. There is one subdirectory that has server1-MONITOR in the name as shown in Figure B-6. Delete this directory and all of its contents.

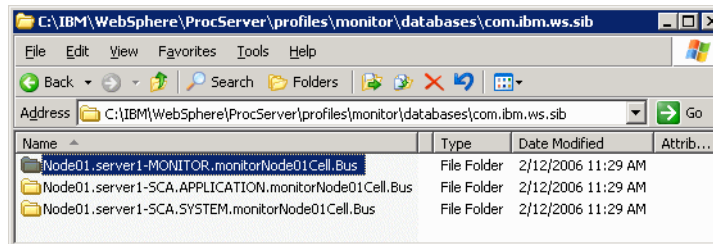
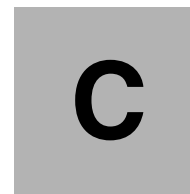


Figure B-6 Deleting the SIBus directory

13. Use the Launchpad on both machines to re-create the databases and install Monitor Server and Dashboard Client, as described in Appendix A, “Product installation” on page 453.
14. Be sure to configure the SIBus link again on the Monitor Server using the configuration script, then restart the servers. You should *not* run the script on the WebSphere Process Server machine because this half of the link will still be configured from the previous installation and configuration.
15. When the installation is complete, the databases will contain no model deployments and no historical data. You can begin to use the server to deploy your new models.



Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG247234>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds to the redbook form number, SG247234.

Using the Web material

The additional Web material that accompanies this redbook includes this file:

<i>File name</i>	<i>Description</i>
SG247234.zip	Zipped Code Samples

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	22 MB for source material
Operating System:	Microsoft Windows 2000 / 2003 / XP
Memory:	1.5 GB or higher recommended

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Abbreviations and acronyms

AAMCAT	Action Manager database	QoS	Quality of Service
BME	Business Measures Editor	ROI	return on investment
BPC	Business Process Choreographer	SACL	State Adaptive Choreography Language
BPM	Business Process Management	SCA	Service Component Architecture
CBE	Common Base Event	SDO	Service Data Objects
CEI	Common Event Infrastructure	SLA	Service Level Agreement
CRM	Customer Relationship Management	SOA	service-oriented architecture
DAS	data access service	SSO	Single sign-on
DBMS	database management system	SVG	Scalable Vector Graphic
EIS	Enterprise Information System	UTE	unit test environment
EJB	Enterprise JavaBeans	WIP	work-in-progress
ERP	Enterprise Resource Planning	WSDL	Web Services Description Language
ESB	Enterprise Service Bus	WS-BPEL	Web Services Business Process Execution Language
HTM	Human Task Manager	WSRP	Web Service Remote Portlets
ISP	Internet service provider	YTD	year to date
J2EE	Java 2 Enterprise Edition		
JCA	J2EE Connector Architecture		
JDBC	Java Database Connectivity		
JMS	Java Messaging Service		
JSF	Java Server Faces		
KPI	Key performance indicator		
LDAP	Lightweight Directory Access Protocol		
LTPA	Lightweight Third Party Authentication		
MDB	Message Driven Beans		
MTD	month to date		
MVC	Model-View-Controller		
OS	operating system		

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 492. Some of the documents referenced here might be available only in softcopy.

- ▶ *Patterns: Building Serial and Parallel Processes for IBM WebSphere Process Server V6*, SG24-7205
- ▶ *Patterns: SOA Foundation - Service Connectivity Scenario*, SG24-7228
- ▶ *Patterns: SOA Foundation - Service Creation Scenario*, SG24-7240
- ▶ *WebSphere Process Integration V6: Business Process Management Modeling through Monitoring*, SG24-7148

Online resources

These Web sites are also relevant as further information sources:

- ▶ *IBM SOA Foundation: An Architectural Introduction and Overview V1.0*
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf>
- ▶ IBM WebSphere Business Modeler
<http://www.ibm.com/software/integration/wbimodeler/>
- ▶ IBM WebSphere Integration Developer
<http://www.ibm.com/software/integration/wid/>
- ▶ IBM WebSphere Process Server
<http://www.ibm.com/software/integration/wps/>
- ▶ IBM WebSphere Business Monitor
<http://www.ibm.com/software/integration/wbimonitor/>

- ▶ IBM Tivoli Composite Application Manager
<http://www.ibm.com/software/tivoli/solutions/application-management/>
- ▶ IBM Patterns for e-business
<http://www.ibm.com/developerworks/patterns>
- ▶ Introduction to the IBM SOA Programming Model
<http://www.ibm.com/developerworks/webservices/library/ws-soa-progmodel/>
- ▶ WebSphere Technical Podcast series
<http://www.ibm.com/developerworks/podcast/websphere>
- ▶ Specifications: Service Component Architecture and Service Data Objects
<http://www.ibm.com/developerworks/webservices/library/specification/ws-scasdosumm>
- ▶ Business Process Execution Language for Web Services version 1.1
<http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel>

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

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Numerics

- 1-pane layout 178
- 3-D bar chart 425
- 4-pane layout 176
- 80/20 situation 50

A

- Access Integration 52
- Access services 15
- Actions (state machines) 110
- Active Instances portlet 399
- Active Instances view 144
- Activity cost analysis 78
- Activity throughput analysis 78
- Adding classifiers 189
- Administrative console 131, 309, 334, 447
- Administrator 347
- Advanced Business Modeling 81
 - WebSphere Business Integration Server Foundation 81
 - WebSphere MQ Workflow 81
 - WebSphere Process Server 81
- Advanced input logic 226
- Advanced output logic 226
- Advanced Wiring 438
- Aggregate metrics 258
- Aggregated analysis 205
- Aggregated Analysis page 399, 418
- Aggregated Values 396
- Aggregated Values portlet 399
- Aggregation 85
- Alerts portlet 399
- Alerts view 148
- Alphablox 381
- analysis functions 77
- Analysis reports 78
- App Server / Services node 56
- Application Integration 52, 64
- Application pattern 51, 60, 65–66
- Application server 58
- Apply daemon 357, 364
- As-Is model 40
- Assemble 11

- WebSphere Integration Developer 37
- Assembly diagram 120
- Assembly editor 102
- Associated data 188
- Attribute panel 80
- Attributes view 176
- Automated actions 143

B

- Basel II 75
- Basic Business Modeling 80
- Best-practice guidelines 51
- Boolean 234
- bottlenecks 34
- Bottom-up approach 95
- BPC Explorer 129, 320, 341
 - Administrator 347
 - End user 341
 - My Process Templates 341
 - My Tasks 343
 - Process Instances 344
- Broker application pattern 54
- Business Analysis page 399
- Business Analyst 37, 209
- business analyst responsibilities 40–41
- Business application services 15
- Business executive 34
- Business innovation services 16
- Business Integration perspective 265
- business item 183
- Business item modeling 76
- Business Logic folder 266, 285, 303
- Business logic services 104
- Business Measures Editor 83, 354
 - Aggregation 85
 - Counters 84
 - KPI 83
 - Metrics 83
 - Situation events 85
 - Stopwatches 84
 - Triggers 83
- Business metrics 138
- Business Modeling

- Advanced Business Modeling 81
- Basic Business Modeling 80
- Intermediate Business Modeling 81
- Business Object Maps 298
- Business patterns 50, 53
- Business Performance (Observation) Model 83
- Business process 56
- Business Process Choreographer (BPC) 316
 - BPC Explorer 129
- Business Process Execution Language (BPEL) 37
- Business process flow 65
- Business process implementation 263
- Business Process Management (BPM) 82
- Business Process Management scenario 30
- Business rules 285
- Business Rules Manager 131, 348
 - Publish and Revert 349
- Business service 54
 - Sequential execution 54
- business user responsibilities 40
- Business value 74, 121, 138

C

- Call Connection variation
 - Product mapping 59, 63–64
- Capture daemon 357, 364
- CCP tables 447
- CICS 20, 126
- CICS Transaction Gateway 21
- CICS Transaction Server 21
- Classifiers 224
 - IT System 189
 - Manual 189
- Cloudscape 58
- Cloudscape database 268
- Cloudscape datasource 309
- Cloudscape JDBC Provider 335
- Collaboration 52
- Command line 337–338
- COMMAREA 21
- Common Base Event (CBE) 123, 141
- Common base event browser 131
- Common Event Infrastructure (CEI) 141
- Compliance 75
- Composite patterns 51
- Composite states 110
- Condition logic 232
- Connector mapping 91

- Connector node 57
- Control Center 426
- Control Center page 399
- Correlation 110, 284
- Correlation set 283
- Cost and Revenue 201
- Cost per time unit 201
- Counters 84, 252
- Create local task 178
- CRM system 100
- CTG
 - See CICS Transaction Gateway
- Cubes 381
- Current Rentals page 399
- Custom registry 134
- Customer requirements 50

D

- daemons
 - Apply 357, 364
 - R2H 364
 - S2R 364
 - Capture 357, 364
 - R2H 364
 - S2R 364
- Dashboard 354
 - 3-D bar chart 425
 - Active Instances view 144
 - Alerts view 148
 - Control Center page 426
 - Dimensions view 147
 - Gauges view 146
 - Key Performance Indicators view 146
 - Organizations view 149
 - Pie chart analysis 424
 - Process Diagram view 148
 - Reports view 144
 - Scorecards view 145
 - Trend chart 426
- Dashboard Client 360, 362, 401, 445
 - Administrative console 401
 - History 360
 - Repository 360
 - Runtime 360
- Data Catalog 177
- Data maps 116, 295
- Data source 335
- Data type 57

- Database management system (DBMS) 57
- Database registry 134
- DataPower 16
- DataPower Toolkit 26
- DataPower XS40 26
- DB2 354
 - Cubes 381
- DB2 Alphablox 381
 - Administrative console 448
- DB2 Command Window 361–362, 366
- DB2 Control Center 390
- DB2 Cube Views 360, 448
 - Metadata object 360
- DB2 Cube Views metadata 355
- DB2 Database Connection 379
- DB2 replication 354
- DB2 Universal Database 63
- DDL files 355, 360
- DDL scripts 361, 445
- Decision mapping 92
- Decision table 112, 285
- deploy phase 12
- Deployable artifacts 331
- Development services 17
- Dimensional analysis 410
- Dimensions portlet 399
- Directory & Security Services node 56
- Directory Services 58
- Document literal 96
- Document literal wrapped 97
- documentation 75
- Documentation reports 78
- Do-While Loop 87
- Dynamic analysis 78
 - Aggregated 205
 - Process Cases 206
 - Process Comparison 206
 - Process Instances 206

E

- EAR file 338
- Eclipse 37
- Edit simulation attribute 203
- EIS
 - See Enterprise Information System
- EIS imports/exports 116
- EJB JAR file 442
- EJB Modules 442

- End user 341
- Enterprise Application 336
- Enterprise application integration (EAI) 100
- Enterprise Architect 13
- Enterprise architect role 209
- Enterprise Demilitarized Zone 67
- Enterprise Information System (EIS) 20
- Enterprise Information Systems (EIS) 114, 126
- Enterprise Secure Zone 67
- Enterprise Service Bus 16
- enterprise transformation 20
- ERP system 100
- ESB
 - See Enterprise Service Bus
- Execution environment 354
- Execution rules 54
- Executive Overview 398, 402, 418
- Export 120, 210
- Export Values portlet 399
- Exposed Serial Process 65
 - Application pattern
 - Workflow variation 66
 - product mapping, WebSphere Application Server Network Deployment 69
- Exposed Serial Workflow variation 65
- Expression Language 281
- Expression type 281
- Extended Enterprise 52, 64–65

F

- Failed event manager 131–132
- File registry 134
- FileRegistrySample.jar 308
- FlatFile 350
- For Loop 87
- Free Form Layout 174

G

- Gateway 25
- Gauges portlet 398
- Gauges view 146, 402
- Generated artifacts 359
- Global Services element 95
- Governance
 - See SOA Governance
 - WebSphere Service Registry and Repository 6
- Graphical modeling panel 80

H

- Historical Configuration 357
- Historical Trends page 399
- History 360
- History database 359
- History database replication (R2H) 364
- HTTP Server 69
- Human Interaction Resolution 55
- Human resources 292
- Human Task 108, 292, 443
- Human task generation 92
- Human Task Manager (HTM) 92

I

- IBM Tivoli Composite Application Manager for SOA 26–28
- If-Then rule set 111
- If-Then Template 289
- Import 120
- Import with SCA Binding 305
- IMS 20, 126
- Information Aggregation 52
- Information as a Service scenario 31
- Information Center 361
- Information services 15
- Information technology 292
- Infrastructure services 17
- Input branches 223
- Input logic 225
- Inputs 220
- Installation capabilities 127
- Integration patterns 50, 53
- Integration services 115
- Integrity and confidentiality 69
- Interaction and Collaboration Services scenario 29
- Interaction services 15
- Inter-enterprise Zone 68
- Interface 120
- Interface maps 115, 295
- Intermediate Business Modeling 81
- Intermediate Results 54, 57
- Internet service provider (ISP) 68
- IT architect 50
- IT architect responsibilities 40
- IT development responsibilities 40
- IT executive 34
- IT production responsibilities 40
- IT requirements 153

- IT service management 17
- IT System Classifier 189
- ITSO Speedy Rentals 153
 - 'As-Is' process 169
 - Business model 154
 - Implementation details 163
 - IT requirements 157
 - Overview of the process 168
 - Simulating the 'As-Is' 198
 - Solution architecture 159

J

- J2C Authentication data 310
- J2EE 20
- J2EE Connector Architecture (JCA) 126
- J2EE DeploymentManager (JSR-88) 129
- J2EE standards 126
- JAAS Configuration 310
- JACL 313
- Java 2 Enterprise Edition(J2EE) 123
- Java Database Connectivity (JDBC) 63, 126
- Java editor 436
- Java implementation file 436
- Java Messaging Service (JMS) 126
- Java Server Faces (JSF) 129
- JAX-RPC wrapper style format 93
- JCA filesystem adapter 294
- JDBC Provider 315, 334

K

- Key performance indicators (KPIs) 82, 256
- Key Performance Indicators (KPIs) portlet 398
- Key Performance Indicators view 146
- KPIs and Aggregate Metrics 256

L

- LDAP configuration 387
- Library 120
- Library project name 433
- Lightweight Directory Access Protocol (LDAP) 57, 64, 133, 307, 387
- Lightweight Third Party Authentication (LTPA) 312
- Literal Value 201
- Local Information Repository 216, 276, 435
- Local integration 26
- Local Operating System (OS) 133
- Local process mapping 91

Local repository mapping 92
Local task mapping 91
Loop conditions 226

M

Manage
 Tivoli Composite Application Manager for SOA 38
 WebSphere Business Monitor 38
Manual Classifier 189
Mapping
 Connector mapping 91
 Decision mapping 92
 Local process mapping 91
 Local repository mapping 92
 Local task mapping 91
 Process mapping 90
 While loop mapping 92
Master Configuration 340
Message Brokers Toolkit 29
Metadata object 360
Metrics 83
Microsoft .NET 58
Model 10
 WebSphere Business Modeler 37
Modeling artifact 234
Modeling Mode 87
modifying a business model 427
Module 120
Module project name 433
Monitor Server 360–361, 445
 Administrative console 447
Monitoring infrastructure 354
Month to date (MTD) 412
My Portal 401
My Process Templates 341
My Tasks 343

N

Notepad 375
Notification actions 143
Notification Broadcaster 87
Notification Receiver 87

O

Observer 87
OLAP Center 379

One-Way Operation 277
Operation faults 97
Operational technology mode 94
Operations page 399
Operations/events 110
Optimization services 16
Organization catalog 177
Organization modeling 77
Organizations portlet 399
Organizations view 149
Originating Task 108
Outline view 80, 176
Output branches 223
Output Directory 356
Output logic 225
Outputs 220

P

Parallel Process application pattern 59–60
 Parallel Workflow variation 59, 61
Parallel Workflow 64
Parallel Workflow variation 59
Participating Task 108
Partner services 15
Partner Zone 68–69
Patterns for e-business 50
 Web site 52
Patterns Web site 53
Performance 122
Performance analysis and reporting 143
Physical Resources view 437
Pie chart analysis 424
Plain Old Java Object (POJO) 114
PoolManager 268
Portal Composite 52
Portlets 393
portType interface 91
Potential Owner 331
Probability (%) 198
Process Cases analysis 206
Process Catalog 177
Process Comparison analysis 206
Process cost analysis 78
Process cost comparison analysis 78
Process Diagrams portlet 399
Process Editor view 176
Process Execution Rules 57
Process flow 56, 64

- Logic 54
- Process Improvement 75
- Process instance resource allocation analysis 78
- Process Instances 344, 348
- Process Instances analysis 206
- Process Interchange File 434
- Process level technical attributes 227
- Process Manager node 56
- Process modeling 76
- Process monitoring 138
- Process Portal 64
- Process services 15
- Process simulation 78
- process specialist responsibilities 40
- Product mapping 51, 69
 - Call Connection variation 59, 63–64
- Project Interchange 264
- Project interchange name 433
- project tree 80
- Project Tree view 176
- Protection of investment 123
- Publish and Revert 349

Q

- Quality of Service (QoS) 124
- Quantitative data 410
- Quantitative data in dimensional analysis 431
- Quantitative metrics 245

R

- R2H Apply daemons 364
- R2H Capture daemons 364
- Rational Application Developer 104, 268
- Rational Software Architect 37, 104, 268
- Recurring time interval for bundle creation 203
- Redbooks Web site 492
 - Contact us xvi
- Reference 120
- Reference Partner 267
- Reliability (service qualifier) 125
- Reports 78
 - Analysis reports 78
 - Documentation reports 78
 - Procedure reports 78
 - Simulation reports 78
 - Weighted average reports 78
- Reports portlet 399
- Reports view 411

- Repository 360
- Repository node 57, 63
- Request/response operation 90
- Resource adapters 126
- Resource catalog 177
- Resource modeling 77
- Resource roles analysis 77
- Resources 224
- Resulting action 250
- Return on Investment (ROI) 75
- Role Members 293
- Role requirement 197
- Roles 224
- RPC 96
- Rule Groups 285
- Rule set 112
- Run
 - WebSphere Process Server 38
- Runtime 360
- Runtime database 359
- Runtime Database Population Interval 357
- Runtime database replication (S2R) 364
- Runtime pattern 51, 56–57, 62, 67

S

- S2R Apply daemons 364
- S2R Capture daemons 364
- SAP 69, 126
- Sarbanes-Oxley 75
- SCA (Service Component Architecture)
 - Process Manager 58
- SCA Binding 437
- SCA component generation 93
- SCA Destination 294
- SCA export 437
- Scalability 122
- Schema artifacts 360
- Scorecards portlet 398
- Scorecards view 145, 404
- Security 122, 125
- Security Services 58
- Select Trigger dialog 243
- Selector component administration 132, 350
- Selectors 293
- Self-Service 52
- Sequential execution 54
- Serial Process 53, 59, 62
 - Generic profile runtime pattern 58

- Orchestration 66
- Pattern 69
- Product mapping 69
- Rules 54
 - rules tier 54
- Runtime pattern 57
- Serial Process application pattern 54
- Serial Workflow 53
 - workflow variation 55
- Serial Process application pattern 54, 60
 - Business and IT drivers 60
- Serial Process Rules 65
- Serial Workflow 53
 - Serial Workflow runtime variation 57
 - Serial Workflow variation 58
- Service Component Architecture (SCA) 37–38, 102
 - Process Manager 58
- Service Connectivity
 - scenario realizations 25
- Service Connectivity scenario 23
 - realizations
 - Gateway 25
 - Local integration 26
- Service Creation scenario 20
 - realizations 20
 - Consume services 22
 - Create Web service from WSDL 22
 - Direct exposure 20
 - Indirect exposure 21
- Service Data Objects (SDOs) 102
- Service Oriented Architecture (SOA) 100
- Service qualifiers 124
 - Reliability 125
 - Security 125
 - Session management 125
 - Transaction support 124
- Services, Ports, and Bindings 97
- Session management 125
- Siebel 126
- Simple decision conditions 233
- Simulation 82
- Simulation Control Panel 203
- Simulation reports 78
- Simulation snapshot 203
- single sign-on (SSO) 312
- Situation events 85
- Six Sigma 75
- SMTP configuration 387
- Snippet 279
- SOA Design 18
- SOA Foundation
 - Reference Architecture 13
 - Access services 15
 - Business application services 15
 - Business innovation services 16
 - Development services 17
 - Enterprise Service Bus 16
 - Information Services 15
 - Interaction services 15
 - Middleware Services view 14
 - Optimization services 16
 - Partner services 15
 - Process services 15
 - Solution view 13
 - scenarios 17
 - Business Process Management 30
 - Information as a Service 31
 - Interaction and Collaboration Services 29
 - Service Connectivity 23
 - Service Creation 20
- SOA Foundation lifecycle 10, 448
 - Assemble 11
 - Deploy 12
 - Manage 12
 - Model 10
- SOA Governance 13, 19
- SOA Management 19
- SOA Security 19
- SOAP 69, 314
- SOAP/HTTPS 69
- Solicit-Response and Notification type operations 97
- Solution architecture 153
- source application tier 54
- Source category 251
- Special case contracts 235
- Staff Worklist Adapter 58–59, 62, 64
- State change 251
- State database 359
- State machines 110
 - Actions 110
 - Composite states 110
 - Conditions 110
 - Correlation 110
 - Operations/events 110
 - States 110
 - Transitions 110
- Stop batch file 379

Stopwatches 84, 249
Structure modeling 77
Synchronize business measures 429

T

Table Space Properties File 356
Target application 54, 58
 Execution rules 54
Target application tier 54
Technical attributes 94
 Process level 227
 Task level
 Task level technical attributes 230
Technical Specification 227
Template rule 291
Templates 113
Test environment 307
Time (Modeling Mode element not supported) 87
Timeout 279
Tivoli Access Manager 26
Tivoli Composite Application Manager 38
Tivoli Composite Application Manager for SOA 35, 38
Tivoli Directory Server 64, 387
Tivoli Enterprise Monitoring Services 39
To-Be model 40
Top level global process 89
Top-down approach 94
Total number of tokens 203
Transaction support 124
Transitions 110
Trend analysis 411
Trend chart 426
Triggers 83
Type hierarchy analysis 77
Type instance matching analysis 78

U

Unit-test environment 309
Upgrades analysis 423
Usage setting 431
user ID 56
User registries 133
UserCustomRegistry 308
UserRegistryTransformation.xsl 309
UserRegistryTransformationwithRoleMembers.xsl 309

V

VAN provider 68
Version control capabilities 134
visual snippet editor 119
visual snippets 119

W

Web service 58
Web Service Interface (.wsdl) 97
Web Service Ports 272
Web services application 339
Web Services Description Language (WSDL) 58
Web Services portTypes 95
Web services standards 123
Web services styles
 Document literal 96
 Document literal wrapped 97
 Operation faults 97
 RPC 96
 Services, Ports, and Bindings 97
 Solicit-Response and Notification type operations 97
Web-based Business Rule Manager 111
WebSphere Adapter 104, 350
WebSphere Adapter for FlatFile 351
WebSphere administrator responsibilities 40
WebSphere Business Integration Adapter 69
WebSphere Business Integration Modeler 86
 Mapping
 Files generated 87
 BPEL 88
 WSDL 88
 XSD 88
 Top level global process 89
 Operational technology mode 94
WebSphere Business Integration Server Foundation 81
WebSphere Business Modeler 37, 73, 263
 Analysis
 Activity cost analysis 78
 Activity throughput analysis 78
 Comparative analysis 78
 Dynamic analysis 78
 Process cost analysis 78
 Process cost comparison analysis 78
 Process instance resource allocation analysis 78
 Resource roles analysis 77

- Type hierarchy analysis 77
- Type instance matching analysis 78
- Attribute panel 80
- Configure 171
- Graphical modeling panel 80
- Outline view 80
- project tree 80
- Publishing Server 79
- WebSphere Business Monitor 35, 38, 73, 137, 353
 - Usage 431
- WebSphere Business Monitor and development tool 260
- WebSphere Enterprise Service Bus 16, 27
- WebSphere Integration Developer 27–28, 37, 73
- WebSphere Member Manager (WMM) 413
- WebSphere Message Broker 16
- WebSphere MQ Workflow 81
- WebSphere Portal 387
- WebSphere Process Server 38, 81, 333, 439
 - Administrator tools
 - Administrative console 131
 - Business Rules Manager 131
 - Common base event browser 131
 - Failed event manager 131
 - Classifiers, resources, and roles 224
 - Input branches and output branches 223
 - Input logic and output logic 225
 - Inputs and outputs 220
 - Loop conditions 226
 - Modeling Mode 87
 - Elements not supported
 - Do-While Loop 87
 - For Loop 87
 - Notification Broadcaster 87
 - Notification Receiver 87
 - Observer 87
 - Time 87
 - Performance 122
 - product 58
 - product mapping 59
 - Scalability 122
 - Security 122
 - User registries
 - Custom registry 134
 - Database registry 134
 - File registry 134
 - Lightweight Directory Access Protocol (LDAP) 133
 - Local Operating System (OS) 133
- WebSphere Process Server Modeling Mode 210
- WebSphere rapid deployment 128
- WebSphere Service Registry and Repository 6
- Weighted average reports 78
- While loop mapping 92
- WID-HOME 308
- Windows User Registry 64
- Wire 120
- Workflow Manager 58, 64
- Workflow variation 66
- Work-in-progress (WIP) 54
- Worklist Adapter 69
- WS-BPEL 123
- WSDL file 58
- WSDL partner link 91
- WSDL portType generation 93
- WS-Security 69
 - Integrity and confidentiality 69

X

- XSD element 93
- XSL 309
- XSLT files 293

Y

- Year to date (YTD) 412



Redbooks

Patterns: SOA Foundation - Business Process Management Scenario

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



Redbooks

Patterns: SOA Foundation - Business Process Management Scenario

**Model, assemble,
deploy, and manage
business process
management
solutions**

**Follow
implementation
guides using IBM
WebSphere software**

**Learn by example
with practical
scenarios**

This IBM Redbook focuses on the Business Process Management service-oriented architecture (SOA) scenario of the IBM SOA Foundation.

The SOA Foundation is an integrated set of software, best practices, and patterns that are based on open standards and designed to provide you what you need to get started with service-oriented architecture.

This book describes modeling, assembling, deploying, and monitoring business processes using IBM software. It highlights the core products for Business Process Management:

- IBM WebSphere Business Modeler V6
- IBM WebSphere Integration Developer V6
- IBM WebSphere Process Server V6
- IBM WebSphere Business Monitor V6

The book also provides a complete end-to-end Business Process Management scenario implementation. Step-by-step instructions are provided for building a scenario for a fictional vehicle rental company called ITS0 Speedy Rentals.

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